

# Computation Over Gaussian Networks With Orthogonal Components

Sang-Woon Jeon, *Member, IEEE*, Chien-Yi Wang, *Student Member, IEEE*, and Michael Gastpar, *Member, IEEE*

## Abstract

Function computation of arbitrarily correlated discrete sources over Gaussian networks with orthogonal components is studied. Two classes of functions are considered: the arithmetic sum function and the type function. The arithmetic sum function in this paper is defined as a set of multiple weighted arithmetic sums, which includes averaging of the sources and estimating each of the sources as special cases. The type or frequency histogram function counts the number of occurrences of each argument, which yields many important statistics such as mean, variance, maximum, minimum, median, and so on. The proposed computation coding first abstracts Gaussian networks into the corresponding modulo sum multiple-access channels via nested lattice codes and linear network coding and then computes the desired function by using linear Slepian–Wolf source coding. For orthogonal Gaussian networks (with no broadcast and multiple-access components), the computation capacity is characterized for a class of networks. For Gaussian networks with multiple-access components (but no broadcast), an approximate computation capacity is characterized for a class of networks.

## Index Terms

Distributed averaging, function computation, joint source–channel coding, lattice codes, linear source coding, network coding, sensor networks.

## I. INTRODUCTION

In wireless sensor networks, the goal of communication is typically for a fusion center to learn a *function* of the sensor observations, rather than the raw observations themselves. Examples include distributed averaging, alarm detection, environmental monitoring, and so on. The fundamental paradigm of digital communication suggests that each sensor should independently compress its observations (using sophisticated compression techniques, taking into account possible correlations in the observations as well as the fact that the fusion center is only interested in a function of the observations), whereupon these compressed versions are communicated reliably (at negligible error probability) to the fusion center. For point-to-point communication, this architecture has been shown to be optimal by Shannon [1], a result that is sometimes referred to as the *source–channel separation theorem*. For general networked communication, however, it is well known that this digital communication paradigm leads to suboptimal performance, see e.g. [2]. Furthermore, in terms of the number of nodes in the network, the suboptimality can be dramatic [3]. Hence, for the communication problem where a fusion center needs to learn a function of the sensor observations, it is beneficial to consider *joint source–channel coding*.

Communication strategies for the problem of function computation over networks have been actively studied in the literature, see e.g. [4]–[8] and the reference therein. For one class of strategies of function computation over wireless networks, the essence is to exploit the superposition property of wireless channels to more efficiently

This work has been supported in part by the European ERC Starting Grant 259530-ComCom. The first author was also funded in part by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R & D Program 2013.

The material in this paper was presented in part at the Information Theory and Applications Workshop (ITA), San Diego, CA, February 2013 and the IEEE International Symposium on Information Theory (ISIT), Turkey, Istanbul, July 2013.

S.-W. Jeon is with the Department of Information and Communication Engineering, Andong National University, South Korea (e-mail: swjeon@anu.ac.kr).

C.-Y. Wang is with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland (e-mail: chien-yi.wang@epfl.ch).

M. Gastpar is with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland and the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA (e-mail: michael.gastpar@epfl.ch).

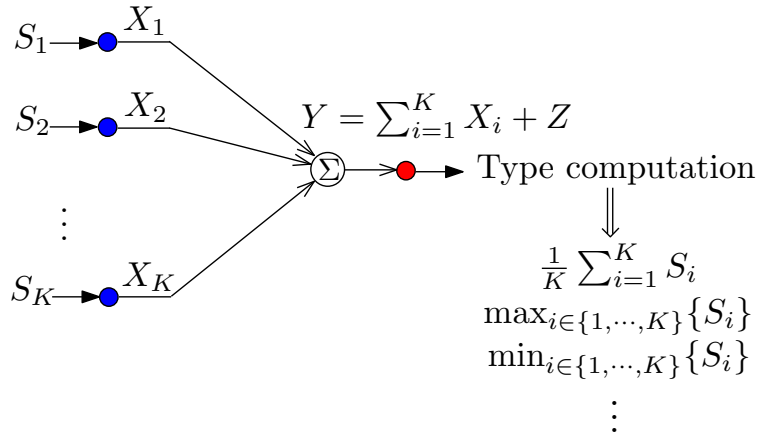


Fig. 1. Type computation over the Gaussian MAC.

compute the desired function. Roughly speaking, previous work in this area can be categorized into three classes: the modulo- $p$  sum computation over (noisy) modulo- $p$  sum networks [4]–[6], [8], [9], the modulo- $p$  sum computation over Gaussian networks assuming an arbitrarily large  $p$  [7], [8], and the sum of Gaussian sources over Gaussian networks under the mean squared error distortion [8], [10]. All these works rely on joint source–channel coding in order to exploit the *similarity between sum-type functions and the superposition property of wireless channels*. We can easily find examples that this joint source–channel coding approach significantly improves an achievable computation rate or decrease an achievable distortion compared to the source–channel separation approach.

In spite of the previous work, however, it is still unclear how to efficiently compute fundamental sample statistics such as sample mean, variance, maximum, minimum, and so on over Gaussian networks. As mentioned before, many sensor applications are interested in the sample mean, for instance, average temperature from several temperature readings. For alarm detection, a relevant function will be the maximum or minimum value among the measurements. One naive approach is to estimate each of the measurements separately, which is universal in the sense that any function of the measurements can be deduced accordingly. Unfortunately, it turns out that this naive approach is quite suboptimal in terms of computation rate for most functions of interest. Another extreme approach is to tackle each function case by case, but we may want to avoid this approach too since there exist numerous important functions to be considered. Therefore, it would be nice to come up with a general coding scheme that is able to compute a broad class of functions including the above fundamental functions but at the same time provide a better computation rate than the separation-based computation.

To achieve this goal, we focus on computing *the type or frequency histogram function* in this paper. For a better understanding, consider the type computation over the Gaussian multiple-access channel (MAC) depicted in Fig. 1. The  $K$  sensors observe their discrete sources  $S_1, S_2, \dots, S_K \in \{0, 1, \dots, p-1\}$ , which can be arbitrarily correlated to each other, and the fusion center wishes to reliably compute its type, represented as

$$\left( \sum_{i=1}^K \mathbf{1}_{S_i=0}, \sum_{i=1}^K \mathbf{1}_{S_i=1}, \dots, \sum_{i=1}^K \mathbf{1}_{S_i=p-1} \right), \quad (1)$$

where  $\mathbf{1}_{(\cdot)}$  denotes the indicator function of an event. As pointed out in [11], computing the type function is very powerful since it yields many important statistics such as sample mean, maximum, minimum, variance, median, mode, and so on. Basically, *any symmetric function whose function value is invariant with respect to permutations of its arguments is computable from the type function*. As seen in (1), the type function consists of multiple arithmetic sums of indicator functions, which can be regarded as binary sources. Therefore, a fundamental question for the type computation is how to exploit the similarity between the arithmetic sum of discrete sources and the superposition property of real-valued transmit signals corrupted by additive noise.

In this paper, we consider the computation of a more general class of functions over a general Gaussian network assuming some orthogonal components, which includes the problem in Fig. 1 as a special case. Two classes of desired functions are considered: the arithmetic sum function and the type function. The former in this paper is defined as a set of multiple weighted arithmetic sums, which includes averaging of the sources and estimating each of the sources as special cases. The latter is counting the number of occurrences of each argument among

the sources, see (1). Regarding the channel model, we consider two types of Gaussian channels. The first model is orthogonal Gaussian networks in which there is no broadcast and multiple-access component, which is equivalent to bit-pipe wired networks [12]–[14]. The second model is Gaussian networks with multiple-access components (and no broadcast component), which includes Gaussian MACs, more generally Gaussian tree networks as special cases [8], [15].

### A. Contribution

The main contributions of the paper is as follows.

- For orthogonal Gaussian single-hop networks, we propose a general computation code which includes both Slepian–Wolf source coding and Körner–Marton linear source coding for computing. An example is presented to demonstrate the benefit of introducing Körner–Marton linear source coding for computing when the sources are correlated.
- We extend Körner–Marton linear source coding for computing to general orthogonal Gaussian networks incorporated with linear network coding at each relay node. We characterize the computation capacity for a class of networks. The result demonstrates that, even without multiple-access component, Körner–Marton linear source coding for computing is still beneficial for a broad class of relay networks.
- For Gaussian MACs, we propose a computation code that first abstracts the original Gaussian MAC into the corresponding modulo sum channel via lattice codes and then applies Körner–Marton linear source coding for computing on top of the transformed channel. We show that the proposed computation code provides a much better computation rate than the separation-based computation, especially when the number of sources becomes large.
- We extend the proposed computation code for Gaussian MACs to general Gaussian networks with multiple-access components. For this, we establish a general transformation method from Gaussian networks with multiple-access components into the corresponding modulo sum channels. On top of this transformed network, we apply the computation code proposed for Gaussian MACs. For a class of networks, we characterize an approximate computation capacity that provides a bounded gap from computation capacity, independent of power  $P$ .

### B. Related Work

In his seminal work [1], Shannon showed that separation of source and channel coding is optimal for discrete memoryless point-to-point channels. However, source–channel separation is not optimal for general networks, for instance, the problems of sending correlated sources over MACs [16], [17] or broadcast channels (BCs) [18], [19]. That is, joint source–channel coding is essentially required for sending correlated sources over networks. Furthermore, it has been proved that an uncoded transmission scheme, a simple way of joint source–channel coding, is optimal or near-optimal for estimating a source from several correlated observations over Gaussian networks [3], [20].

Function computation has been actively studied in the source coding perspective [4], [5], [21]–[26]. In particular, computing the modulo-two sum has been considered in [4] under the distributed source coding framework, which captures the potential of linear source coding [27] for function computation. A more general achievability has been proposed for the modulo-two sum computation in [5] and for a general discrete function in [25], [26]. In [22], computing a general function with the help of side information has been studied. Function computation has been also considered in the context of cascade source coding [23] and interactive source coding [24].

The modulo sum or more generally linear function computation has been recently extended to relay networks under various channel models such as bit-pipe wired networks [28]–[30], linear finite field networks [8], Gaussian networks assuming no broadcast component [8] by incorporating linear network coding [12]–[14], [31], [32] at each relay node. Function multicasting has been studied for linear finite field interference channels [9] and for undirected graphs [33], [34]. A more general classes of function computation over bit-pipe wired networks has been considered in [28], [35].

Scaling laws on function computation has been studied based on the collision model [11], [36], in which concurrent transmission from multiple senders is assumed to cause a collision and, therefore, is not allowable. In particular, it has been shown that the order of  $\frac{1}{\log K}$  scaling law is achievable as the number of sources  $K$

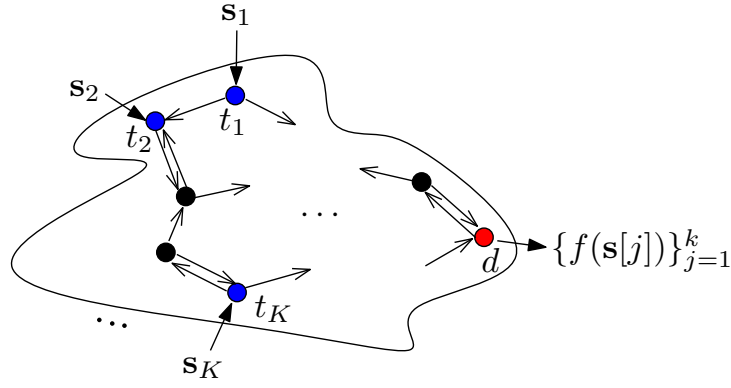


Fig. 2. Computation over a network in which node  $t_i$  observes the length- $k$  source  $\mathbf{s}_i = [s_i[1], \dots, s_i[k]]^T$  and node  $d$  wishes to compute the desired function  $\{f(\mathbf{s}[j])\}_{j=1}^k$ , where  $\mathbf{s}[j] = [s_1[j], \dots, s_K[j]]^T$ .

increases for the type or frequency histogram computation over collocated collision networks [11], [36]. Recently, it has been shown that non-vanishing scaling law is achievable for the type-threshold function computation over collocated Gaussian network even as  $K$  tends to infinity [37].

The potential of linear source coding has been also captured by Nazer and Gastpar in [6], applying the linear source coding in [4] for the function computation over MACs. An efficient way of computing the modulo sum or the sum of Gaussian sources over Gaussian MACs is to apply lattice codes [6], [10], see also [38]–[40] for lattice code construction. Lattice-based network computation has been recently extended to multiple receivers called compute-and-forward [7] in which each relay computes or decodes linear combination of the sources. In [8], a similar lattice code construction has been used for computing a linear function over linear finite field networks and the sum of Gaussian sources over Gaussian networks.

## II. PROBLEM FORMULATION

Throughout the paper, we denote  $[1:n] := \{1, 2, \dots, n\}$ ,  $C(x) := \frac{1}{2} \log(1+x)$ , and  $C^+(x) := \max\{\frac{1}{2} \log(x), 0\}$ . For  $x_i \in \mathbb{F}_p$ ,  $\bigoplus_{i=1}^n x_i$  denotes the modulo- $p$  sum of  $\{x_i\}_{i \in [1:n]}$ , where  $p$  is assumed to be a prime number. Let  $\mathbf{1}_{(\cdot)}$  denote the indicator function of an event. For random variables  $A$  and  $B$ ,  $H(A)$  denotes the entropy of  $A$  and  $I(A; B)$  denotes the mutual information between  $A$ .

### A. Network Model

Consider a network represented by a directed graph  $G = (V, E)$  depicted in Fig. 2. Denote the set of incoming and outgoing nodes at node  $v \in V$  by  $\Gamma_{in}(v) = \{u \in V : (u, v) \in E\}$  and  $\Gamma_{out}(v) = \{u \in V : (v, u) \in E\}$ , respectively. Denote the  $i$ th sender,  $i \in [1:K]$ , by  $t_i \in V$  and suppose that it observes a length- $k$  discrete source vector  $\mathbf{s}_i = [s_i[1], \dots, s_i[k]]^T \in [0:p-1]^k$ . Denote the set of  $K$  sources at time  $j$  by  $\mathbf{s}[j] = [s_1[j], \dots, s_K[j]]^T$ . The receiver  $d \in V$  wishes to compute a symbol-by-symbol function of  $K$  sources, i.e.,  $f(\mathbf{s}[j])$  for all  $j \in [1:k]$ . We assume that  $d \notin \{t_i\}_{i \in [1:K]}$  and  $G$  contains a directed path from all nodes in  $V$  to the receiver  $d$ . Without loss of generality, we assume that the nodes with no incoming edge are included in  $\{t_i\}_{i \in [1:K]}$ .

We mainly consider two desired functions: the arithmetic sum function and the type or frequency histogram function, whose formal definitions are given below.

*Definition 1 (Arithmetic Sum Function):* Let  $\mathbf{s} = [s_1, \dots, s_K]^T \in [0:p-1]^K$ . For the arithmetic sum computation, the desired function is given by  $f(\mathbf{s}) = \{\sum_{i=1}^K a_{li} s_i\}_{l=1}^L$ , where  $a_{li} \in [0:p-1]$ . Hence  $f(\mathbf{s}) \in [0:(p-1)^2 K]^L$  for the arithmetic sum function.

*Definition 2 (Type Function):* Let  $\mathbf{s} = [s_1, \dots, s_K]^T \in [0:p-1]^K$  and  $b_l(\mathbf{s}) = \sum_{i=1}^K \mathbf{1}_{s_i=l}$  for  $l \in [0:p-1]$ . For the type computation, the desired function is given by  $f(\mathbf{s}) = \{b_0(\mathbf{s}), \dots, b_{p-1}(\mathbf{s})\}$ . Hence  $f(\mathbf{s}) \in [0:K]^p$  for the type function.

*Remark 1 (Arithmetic Sum and Type):* The arithmetic sum function in this paper is defined as multiple weighted arithmetic sums, which includes averaging of the sources and estimating each of the sources as special cases. As shown in Definition 2, the type function can be also represented as multiple arithmetic sums. Therefore, the essence

of the type computation is how to efficiently compute arbitrarily correlated multiple arithmetic sums over Gaussian networks.

*Remark 2 (Symmetric Function Computation):* As pointed out by [11], computing the type function is very powerful since it yields many important statistics such as sample mean, maximum, minimum, variance, median, mode, and so on. Basically, any symmetric function whose function value is invariant with respect to permutations of its arguments is computable from the type function. That is, symmetric functions satisfy  $f(s_1, s_2, \dots, s_K) = f(s_{\sigma_1}, s_{\sigma_2}, \dots, s_{\sigma_K})$  for any permutation set  $\{\sigma_i\}_{i \in [1:K]}$  and, therefore, they are deterministic functions of the type function.

We assume arbitrarily correlated stationary and ergodic sources. The following definition formally states the underlying probability distribution and the corresponding random variables regarding the set of  $K$  sources.

*Definition 3 (Sources):* Let  $\mathbf{S} = [S_1, \dots, S_K]^T \in [0 : p - 1]^K$  be a random vector associated with a joint probability mass function  $p_{\mathbf{S}}(\cdot)$ . At each time  $j \in [1 : k]$ ,  $\mathbf{s}[j]$  is assumed to be independently drawn from  $p_{\mathbf{S}}(\cdot)$ .

As a special case in Definition 3, we will consider the following doubly symmetric binary sources throughout the paper.

*Definition 4 (Doubly Symmetric Binary Sources):* Assume  $K = 2$ . Denote the doubly symmetric binary sources with the associated probability  $\alpha$  by DSBS( $\alpha$ ). Let  $\text{Bern}(a)$  be the Bernoulli distribution with the probability  $a$ . For DSBS( $\alpha$ ),  $S_1$  follows  $\text{Bern}(1/2)$  and  $S_2 = S_1 \oplus Z$ , where  $Z$  follows  $\text{Bern}(\alpha)$  and is independent of  $S_1$ .

Let  $f(\mathbf{S})$  denote the desired function induced by the random source vector  $\mathbf{S}$ . The following two definitions define random variables associated with the desired function, which will be used throughout the paper.

*Definition 5 (Arithmetic Sum Function Induced by  $\mathbf{S}$ ):* Define  $U_l = \sum_{i=1}^K a_{li} S_i$  for  $l \in [1 : L]$ , which are the random variables associated with the arithmetic sum function. Then  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function.

*Definition 6 (Type Function Induced by  $\mathbf{S}$ ):* Define  $B_l = \sum_{i=1}^K \mathbf{1}_{S_i=l}$  for  $l \in [0 : p - 1]$ , which are the random variables associated with the type function. Then  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Remark 3 (Worst Case Sources):* Note that  $H(f(\mathbf{S}))$  is upper bounded by  $\min\{K \log p, L \log(p^2 K)\}$  for the arithmetic sum function and  $\min\{K \log p, p \log(K + 1)\}$  for the type function. For both cases,  $H(f(\mathbf{S}))$  scales as the order of  $\log K$  as the number of sources  $K$  increases.

Associated with  $G = (V, E)$ , we consider two classes of Gaussian channels, which are formally stated in the following two definitions.

*Definition 7 (Orthogonal Gaussian Networks):* For this case, we assume Gaussian point-to-point channels with no broadcast and no multiple-access for each  $(u, v) \in E$ . That is, the length- $n$  time-extended input-output is given by

$$\mathbf{y}_{u,v} = h_{u,v} \mathbf{x}_{u,v} + \mathbf{z}_{u,v}, \quad (2)$$

where the elements of  $\mathbf{z}_{u,v}$  are independently drawn from  $\mathcal{N}(0, 1)$ . Each transmit signal should satisfy  $\frac{1}{n} \|\mathbf{x}_{u,v}\|^2 \leq P$  for all  $(u, v) \in E$ . For notational simplicity, we will use the subscript  $(\cdot)_{\text{ptp}}$  to denote orthogonal Gaussian networks.

*Definition 8 (Gaussian Networks With Multiple-Access):* For this case, we assume Gaussian multiple-access channels with no broadcast from  $u \in \Gamma_{in}(v)$  to each  $v \in V$ . That is, the length- $n$  time-extended input-output is given by

$$\mathbf{y}_v = \sum_{u \in \Gamma_{in}(v)} h_{u,v} \mathbf{x}_{u,v} + \mathbf{z}_v, \quad (3)$$

where the elements of  $\mathbf{z}_v$  are independently drawn from  $\mathcal{N}(0, 1)$ . Each transmit signal should satisfy  $\frac{1}{n} \|\mathbf{x}_{u,v}\|^2 \leq P$  for all  $(u, v) \in E$ . For notational simplicity, we will use the subscript  $(\cdot)_{\text{mac}}$  to denote Gaussian networks with multiple-access.

*Remark 4 (Bit-Pipe Wired Networks):* The considered orthogonal Gaussian network is almost equivalent to a bit-pipe wired network in the sense that it can be easily converted into a bit-pipe wired network by using capacity-achieving point-to-point channel codes. Nevertheless, we will state this paper based on Gaussian networks assuming orthogonal components defined in Definitions 7 and 8.

*Remark 5 (Single-Hop Networks):* For notational simplicity, we will use the following simplified notation for the single-hop case. For orthogonal Gaussian single-hop networks, we rewrite the length- $n$  time-extended input-output as

$$\mathbf{y}_i = h_i \mathbf{x}_i + \mathbf{z}_i, \quad (4)$$

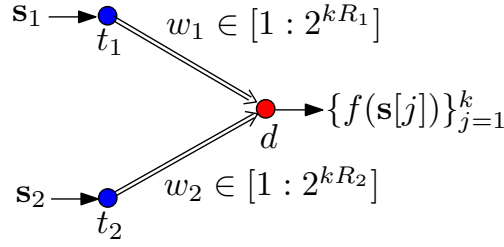


Fig. 3. Two-user distributed source coding for function computation.

where  $i \in [1 : K]$ . For Gaussian single-hop networks with multiple-access or Gaussian MACs, we rewrite the length- $n$  time-extended input–output as

$$\mathbf{y} = \sum_{i=1}^K h_i \mathbf{x}_i + \mathbf{z}. \quad (5)$$

### B. Computation Capacity

Based on the above network model, the length- $n$  block code for orthogonal Gaussian networks is defined as follows, where  $\mathbf{y}^a$  denotes  $y[1], \dots, y[a]$

- (Sender Encoding) The  $i$ th sender  $t_i$  transmits  $x_{t_i,w}^{(t)} = \psi_{t_i,w}^{(t)}(\mathbf{s}_i, \{\mathbf{y}_{u,t_i}^{t-1}\}_{u \in \Gamma_{in}(t_i)})$  for  $t \in [1 : n]$  to node  $w \in \Gamma_{out}(t_i)$ .
- (Relay Encoding) Node  $v \notin \{t_i\}_{i \in [1:K]} \cup \{d\}$  transmits  $x_{v,w}^{(t)} = \psi_{v,w}^{(t)}(\{\mathbf{y}_{u,v}^{t-1}\}_{u \in \Gamma_{in}(v)})$  for  $t \in [1 : n]$  to node  $w \in \Gamma_{out}(v)$ .
- (Decoding) The receiver  $d$  estimates  $\hat{f}(\mathbf{s}[j]) = \varphi^{(j)}(\{\mathbf{y}_{u,d}\}_{u \in \Gamma_{in}(d)})$  for  $j \in [1 : k]$ .

Similarly, we can define the length- $n$  block code for Gaussian networks with multiple-access. Specifically,  $x_{t_i,w}^{(t)} = \phi_{t_i,w}^{(t)}(\mathbf{s}_i, \mathbf{y}_{t_i}^{t-1})$ ,  $x_{v,w}^{(t)} = \phi_{v,w}^{(t)}(\mathbf{y}_v^{t-1})$  for  $v \notin \{t_i\}_{i \in [1:K]} \cup \{d\}$ , and  $\hat{f}(\mathbf{s}[j]) = \varphi^{(j)}(\mathbf{y}_d)$ .

The probability of error is defined by  $P_e^{(n)} = \Pr \left[ \bigcup_{j=1}^k \hat{f}(\mathbf{s}[j]) \neq f(\mathbf{s}[j]) \right]$ . We then define the computation capacity as the follow.

*Definition 9 (Computation Capacity):* The computation rate  $R := \frac{k}{n}$  is said to be achievable if there exists a sequence of length- $n$  block codes such that  $P_e^{(n)}$  converges to zero as  $n$  increases. The computation capacity is the maximum over all achievable computation rates.

From Definition 9, the computation rate is the number of reliably computable functions per channel use.

## III. PRELIMINARIES

Before stating our main results, we first introduce previous work that is closely related to our work in Sections III-A to III-C. For comparison, we introduce a cut-set upper bound in Section III-D and a separation-based lower bound in Section III-E.

### A. Distributed Source Coding

Figure 3 illustrates the two-user distributed source coding for function computation. Two senders respectively observe the length- $k$  sources  $\mathbf{s}_1$  and  $\mathbf{s}_2$  and deliver some information for function computing via messages  $w_1 \in [1 : 2^{kR_1}]$  and  $w_2 \in [1 : 2^{kR_2}]$ . The receiver wishes to compute the desired function  $\{f(\mathbf{s}[j])\}_{j=1}^k$  based on  $(w_1, w_2)$ .

The optimal distributed lossless source coding for  $f(\mathbf{s}[j]) = \mathbf{s}[j]$  in Fig. 3 has been solved by Slepian and Wolf [41]. For the two-user case, the Slepian–Wolf rate region is the set of all rate pairs  $(R_1, R_2)$  satisfying

$$\begin{aligned} R_1 &\geq H(S_1|S_2), \\ R_2 &\geq H(S_2|S_1), \\ R_1 + R_2 &\geq H(S_1, S_2). \end{aligned} \quad (6)$$

Obviously, the above rate region is also an achievable rate region for any desired function. It was proved by Csiszár in [27] that the same Slepian–Wolf rate region is achievable by linear source coding.

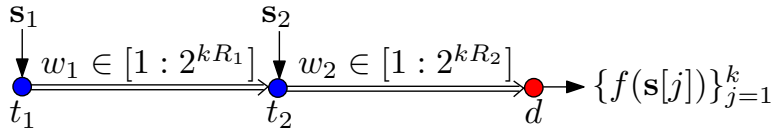


Fig. 4. Two-user cascade source coding for function computation.

The potential of the linear source coding has been first captured by Körner and Marton in [4] in the context of the modulo-two sum computation. Consider binary field sources  $\mathbf{s}_1 \in \mathbb{F}_2^k$  and  $\mathbf{s}_2 \in \mathbb{F}_2^k$  and  $\{f(\mathbf{s}[j]) = s_1[j] \oplus s_2[j]\}_{j=1}^k$ . It was proved in [4] that the set of all rate pairs  $(R_1, R_2)$  satisfying

$$\begin{aligned} R_1 &\geq H(S_1 \oplus S_2), \\ R_2 &\geq H(S_1 \oplus S_2) \end{aligned} \quad (7)$$

is achievable by linear source coding. A simple outer bound shows a necessary condition on an achievable  $(R_1, R_2)$  as

$$\begin{aligned} R_1 &\geq H(S_1|S_2), \\ R_2 &\geq H(S_1|S_2), \\ R_1 + R_2 &\geq H(S_1 \oplus S_2). \end{aligned} \quad (8)$$

*Example 1 (Modulo-Two Sum of DSBS( $\alpha$ )):* For DSBS( $\alpha$ ), the Körner–Marton rate region (7) shows that any rate pair satisfying  $R_1 \geq H_2(\alpha)$  and  $R_2 \geq H_2(\alpha)$  is achievable, which coincides with the outer bound in (8).

Unfortunately, the optimal rate region for the modulo-two sum computation of arbitrarily correlated binary sources is unknown. The outer bound in (8) does not coincide with the convex hull of the union of the Slepian–Wolf rate region (6) and the Körner–Marton rate region (7). A more general achievability containing both the Slepian–Wolf rate region (6) and the Körner–Marton rate region (7) has been proposed by Ahlswede and Han in [5, Section VI]. Let  $W_1$  and  $W_2$  be auxiliary random variables that form a Markov chain  $W_1 - S_1 - S_2 - W_2$ . Ahlswede and Han showed that the set of all rate pairs  $(R_1, R_2)$  satisfying

$$\begin{aligned} R_1 &\geq I(W_1; S_1|W_2) + H(S_1 \oplus S_2|W_1, W_2), \\ R_2 &\geq I(W_2; S_2|W_1) + H(S_1 \oplus S_2|W_1, W_2), \\ R_1 + R_2 &\geq I(W_1, W_2; S_1, S_2) + 2H(S_1 \oplus S_2|W_1, W_2) \end{aligned} \quad (9)$$

is achievable. An example that this new rate region strictly enlarges the convex hull of the union of the Slepian–Wolf rate region (6) and the Körner–Marton rate region (7) was also provided in [5, Example 4].

## B. Cascade Source Coding

Figure 4 illustrates the two-user cascade source coding for function computation. The first sender observes the length- $k$  source  $\mathbf{s}_1$  and delivers some information for function computing via message  $w_1 \in [1 : 2^{kR_1}]$ . The second sender observes  $(\mathbf{s}_2, w_1)$  and again delivers some information of  $(\mathbf{s}_2, w_1)$  for function computing via message  $w_2 \in [1 : 2^{kR_2}]$ . The receiver wishes to compute the desired function  $\{f(\mathbf{s}[j])\}_{j=1}^k$  based on  $w_2$ .

The cascade source coding for function computation has been studied in [23] in the context of lossy computation. For the lossless case depicted in Fig. 4, the computation capacity for a general function has been shown to be represented by the conditional graph entropy [22].

Again, consider arbitrarily correlated binary field sources  $\mathbf{s}_1 \in \mathbb{F}_2^k$  and  $\mathbf{s}_2 \in \mathbb{F}_2^k$  and the modulo-two sum computation of these two sources. In this case, the optimal rate region is given by the set of all  $(R_1, R_2)$  satisfying

$$\begin{aligned} R_1 &\geq H(S_1|S_2), \\ R_2 &\geq H(S_1 \oplus S_2), \end{aligned} \quad (10)$$

which can be attained from a simple application of Slepian–Wolf source coding.

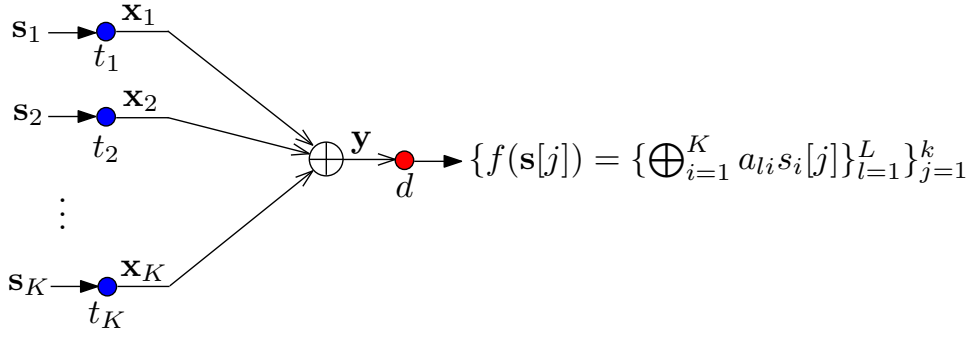


Fig. 5. Modulo- $p$  sum computation over the modulo- $p$  sum channel.

### C. Modulo- $p$ Sum Computation Over MAC

The potential of linear source coding has been also captured by Nazer and Gastpar in the context of the computation over MAC [6]. Figure 5 illustrates the modulo- $p$  sum computation over the deterministic modulo- $p$  sum channel and a more general case can be found in [6, Theorem 1]. The  $i$ th sender observes  $s_i \in \mathbb{F}_p^k$  and the receiver wishes to compute the modulo- $p$  sum function, i.e.,  $f(\mathbf{s}[j]) = \{\bigoplus_{i=1}^K a_{li} s_i[j]\}_{l=1}^L$ ,  $a_{li} \in \mathbb{F}_p$ . The length- $n$  time-extended input-output of the modulo- $p$  sum channel is given by  $\mathbf{y} = \bigoplus_{i=1}^K \mathbf{x}_i$ , where  $\mathbf{x}_i \in \mathbb{F}_p^n$  and  $p$  is assumed to be a prime number.

To compress multiple modulo- $p$  sum functions, which are in general correlated to each other, Nazer and Gastpar applied linear Slepian-Wolf source coding as introduced in the following lemma.

*Lemma 1 (Csiszár [27]):* Let  $(\mathbf{v}_1, \dots, \mathbf{v}_L)$  be the set of length- $k$  sources, independently drawn from some joint probability mass function  $p_{V_1, \dots, V_L}(\cdot)$ . For any point in the Slepian-Wolf rate region, there exist matrices  $\mathbf{H}_1, \dots, \mathbf{H}_L$  of size  $n_l \times k$ , respectively, taking values over a finite field with associated decoding function that can be used to compress the sources in a distributed fashion with  $\Pr[(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_L) \neq (\mathbf{v}_1, \dots, \mathbf{v}_L)] \rightarrow 0$  as  $k$  increases.

*Theorem 1 (Nazer-Gastpar [6]):* Consider the modulo- $p$  sum computation over the modulo- $p$  sum channel depicted in Fig. 5. Let  $V_l = \bigoplus_{i=1}^K a_{li} S_i$ . Then the computation capacity is given by

$$R = \frac{\log p}{H(V_1, \dots, V_L)}. \quad (11)$$

For a better understanding, we briefly explain the achievability here. From lemma 1, we set  $\mathbf{H}_1, \dots, \mathbf{H}_L$  of size  $n_l \times k$ , respectively, which corresponds to some point in the Slepian-Wolf rate region with sum rate  $H(V_1, \dots, V_L)$ .

The  $i$ th sender transmits

$$\mathbf{x}_i = [a_{1i} \mathbf{H}_1 \mathbf{s}_i, \dots, a_{Li} \mathbf{H}_L \mathbf{s}_i]^T \quad (12)$$

for  $i \in [1 : K]$ , where we set  $n = \sum_{l=1}^L n_l$ . Then (12) yields

$$\mathbf{y} = \bigoplus_{i=1}^K \mathbf{x}_i = \left[ \mathbf{H}_1 \left( \bigoplus_{i=1}^K a_{1i} \mathbf{s}_i \right), \dots, \mathbf{H}_L \left( \bigoplus_{i=1}^K a_{Li} \mathbf{s}_i \right) \right]^T. \quad (13)$$

Hence, from Lemma 1, the receiver can recover  $\{\bigoplus_{i=1}^K a_{li} \mathbf{s}_i\}_{l \in [1:L]}$  with an arbitrarily small probability of error as  $n$  increases if

$$\left( \sum_{l=1}^L n_l \right) \log p = n \log p \geq k H(V_1, \dots, V_L). \quad (14)$$

Therefore, setting  $k = \frac{n \log p}{H(V_1, \dots, V_L)}$ , which satisfies (14), provides that

$$R = \frac{\log p}{H(V_1, \dots, V_L)} \quad (15)$$

is achievable.

Nazer and Gastpar recently proposed compute-and-forward [7], which provides a general framework for computing modulo- $p$  sum functions at multiple receivers over Gaussian channels. We briefly describe compute-and-forward here with respect to a single receiver depicted in Fig. 6. The  $i$ th sender observes  $s_i \in \mathbb{F}_p^k$  and the receiver wishes to



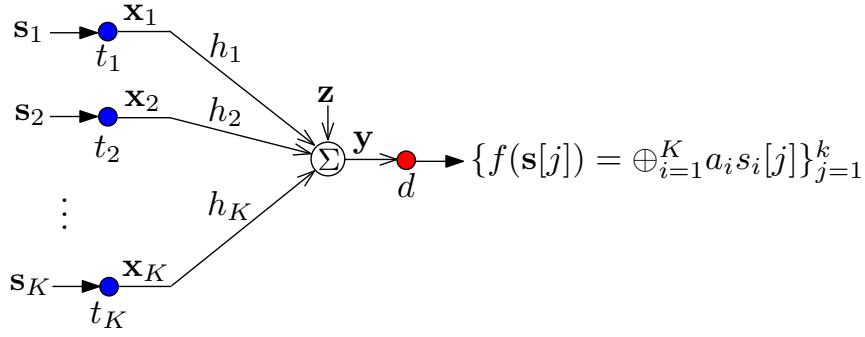


Fig. 6. Modulo- $p$  sum computation over the Gaussian MAC.

compute the modulo- $p$  sum function  $f(\mathbf{s}[j]) = \bigoplus_{i=1}^K a_i s_i[j]$ ,  $a_i \in \mathbb{F}_p$ . Here  $p$  is assumed to be a prime number. The length- $n$  time-extended input–output is given by  $\mathbf{y} = \sum_{i=1}^K h_i \mathbf{x}_i + \mathbf{z}$ , where the elements of  $\mathbf{z}$  are independently drawn from  $\mathcal{N}(0, 1)$  and  $\frac{1}{n} \|\mathbf{x}_i\|^2 \leq P$  for all  $i \in [1 : K]$ .

*Theorem 2 (Nazer–Gastpar [7]):* Consider the modulo- $p$  sum computation over the Gaussian MAC depicted in Fig. 6. Let  $\mathbf{a} = [a_1, \dots, a_K]^T$  and  $\mathbf{h} = [h_1, \dots, h_K]^T$ . Then the receiver can decode  $\{f(\mathbf{s}[j])\}_{j=1}^k = \bigoplus_{i=1}^K a_i \mathbf{s}_i$  reliably for  $n$  sufficiently large if

$$R = \frac{k}{n} \leq C^+ \left( \left( \|\mathbf{a}\|^2 - \frac{P(\mathbf{h}^T \mathbf{a})^2}{1 + P\|\mathbf{h}\|^2} \right)^{-1} \right) (\log p)^{-1} \quad (16)$$

and  $p$  is an increasing function of  $n$  such that  $\frac{n}{p} \rightarrow 0$  as  $n \rightarrow \infty$ .

*Example 2 (Gaussian MAC When  $h_i = a_i = 1$ ):* Suppose that  $h_i = a_i = 1$  for all  $i \in [1 : K]$ . For this case, Theorem 2 yields that the receiver can decode  $\bigoplus_{i=1}^K \mathbf{s}_i$  reliably for  $n$  sufficiently large if  $R \leq C^+ \left( \frac{1}{K} + P \right) (\log p)^{-1}$  and  $p$  is an increasing function of  $n$  satisfying that  $p \rightarrow \infty$  as  $n \rightarrow \infty$ .

#### D. Cut-Set Upper Bound

To describe a cut-set upper bound on the computation capacity, we first introduce the notation. For a subset  $\Sigma \subseteq [1 : K]$ , define  $G(\Sigma) = (V(\Sigma), E(\Sigma))$  as the subgraph of  $G$  consisting of the nodes having a direct path from at least one of the senders in  $\{t_i\}_{i \in \Sigma}$ . Let  $\Lambda(\Sigma)$  denote the set of all cuts dividing all of the senders in  $\{t_i\}_{i \in \Sigma}$  from the receiver  $d$  on  $G(\Sigma)$ . Then the minimum-cut value for general discrete memoryless networks over the cuts in  $\Lambda(\Sigma)$  on  $G(\Sigma)$  is given by

$$\max_{p(\{x_v\}_{v \in V(\Sigma)})} \min_{\Omega \in \Lambda(\Sigma)} I(X_\Omega; Y_{\Omega^c} | X_{\Omega^c}). \quad (17)$$

If there exists an input cost constraint,  $p(\{x_v\}_{v \in V(\Sigma)})$  should be set to satisfy the corresponding input cost constraint.

For orthogonal Gaussian networks, the minimum-cut value is given by

$$\bar{C}_{\text{ptp}}(\Sigma) := \min_{\Omega \in \Lambda(\Sigma)} \sum_{(u,v) \in E(\Sigma), u \in \Omega, v \in \Omega^c} C(h_{u,v}^2 P). \quad (18)$$

Similarly, for Gaussian networks with multiple-access, the minimum-cut value is upper bounded by

$$\bar{C}_{\text{mac}}(\Sigma) := \min_{\Omega \in \Lambda(\Sigma)} \sum_{v \in \Omega^c} C \left( \left( \sum_{u \in \Gamma_{\text{in}}(v), u \in \Omega} |h_{u,v}| \right)^2 P \right). \quad (19)$$

Since the desired function is locally computable [11], for any  $\Sigma \subseteq [1 : K]$ ,  $f(\mathbf{s}[j])$  can be represented as  $f'(\{s_i[j]\}_{i \in \Sigma}) + f''(\{s_i[j]\}_{i \in [1:K] \setminus \Sigma})$ . Hence, by assuming that a genie provides  $\{s_i\}_{i \in [1:K] \setminus \Sigma}$  to the receiver, computing  $\{f'(\{s_i[j]\}_{i \in \Sigma})\}_{j=1}^k$  at the receiver is enough to recover the desired function  $\{f(\mathbf{s}[j])\}_{j=1}^k$ . Therefore assuming full cooperation between the nodes in  $\Omega$  and between the nodes in  $\Omega^c$ , from the source–channel separation theorem [1],

$$R_{\text{ptp}} \leq \min_{\Sigma \subseteq [1:K]} \frac{\bar{C}_{\text{ptp}}(\Sigma)}{H(f(\mathbf{S}) | \{S_i\}_{i \in [1:K] \setminus \Sigma})} \quad (20)$$

for orthogonal Gaussian networks. In the same manner, we have

$$R_{\text{mac}} \leq \min_{\Sigma \subseteq [1:K]} \frac{\bar{C}_{\text{mac}}(\Sigma)}{H(f(\mathbf{S})|\{S_i\}_{i \in [1:K] \setminus \Sigma})} \quad (21)$$

for Gaussian networks with multiple-access.

### E. Separation-Based Computation

We generalize the notion of the separation-based computation introduced in [6]. We refer to [6, Section III] for the formal definition of the separation-based computation. Roughly speaking, the separation-based computation means that a communication network is first transformed into an end-to-end bit-pipe channel by channel coding and then separately applied source coding for computing the desired function over the transformed end-to-end bit-pipe channel.

Let  $\mathbf{R}_f$  denote the distributed compression rate region for computing  $\{f(\mathbf{s}[j])\}_{j=1}^k$  (see [6, Definition 8]) and  $\mathbf{C}_{\text{ptp}}$  denote the capacity region for orthogonal Gaussian networks, which can be represented as the set of all rate tuples such that

$$\sum_{i \in \Sigma} R_i \leq \bar{C}_{\text{ptp}}(\Sigma) \text{ for all } \Sigma \subseteq [1:K]. \quad (22)$$

Then a computation rate  $R_{\text{ptp}}^{(\text{sep})}$  is achievable by separation if

$$\mathbf{R}_f \cap \mathbf{C}'_{\text{ptp}} \neq \emptyset, \quad (23)$$

where  $\mathbf{C}'_{\text{ptp}} = \left\{ \left( \frac{R_1}{R_{\text{ptp}}^{(\text{sep})}}, \dots, \frac{R_K}{R_{\text{ptp}}^{(\text{sep})}} \right) : (R_1, \dots, R_K) \in \mathbf{C}_{\text{ptp}} \right\}$ .

Similarly, we define the achievable computation rate  $R_{\text{mac}}$  for Gaussian networks with multiple-access by separation. Specifically,  $R_{\text{mac}}^{(\text{sep})}$  is achievable by separation if

$$\mathbf{R}_f \cap \mathbf{C}'_{\text{mac}} \neq \emptyset, \quad (24)$$

where  $\mathbf{C}'_{\text{mac}} = \left\{ \left( \frac{R_1}{R_{\text{mac}}^{(\text{sep})}}, \dots, \frac{R_K}{R_{\text{mac}}^{(\text{sep})}} \right) : (R_1, \dots, R_K) \in \mathbf{C}_{\text{mac}} \right\}$  and  $\mathbf{C}_{\text{mac}}$  denotes the capacity region for Gaussian networks with multiple-access, which is upper bounded by

$$\sum_{i \in \Sigma} R_i \leq \bar{C}_{\text{mac}}(\Sigma) \text{ for all } \Sigma \subseteq [1:K]. \quad (25)$$

*Example 3 (I.I.D. Sources and Symmetric MAC):* Suppose that the sources are i.i.d., i.e.,  $p_{\mathbf{S}}(\cdot) = \prod_{i=1}^K p_{S_i}(\cdot)$  and  $p_{S_i}(\cdot) = p_S(\cdot)$ . Then from [6, Lemma 1] (also see [6, Example 1]),  $\mathbf{R}_f$  is given by all rate tuples such that  $R_i \geq H(S)$  for both the arithmetic sum function and the type function computation. Therefore, for symmetric MAC, see the definition in [6, Definition 11], an achievable computation rate by separation is upper bounded by

$$R_{\text{ptp}}^{(\text{sep})} \leq \frac{\bar{C}_{\text{ptp}}([1:K])}{KH(S)} \quad (26)$$

for orthogonal Gaussian single-hop networks and

$$R_{\text{mac}}^{(\text{sep})} \leq \frac{\bar{C}_{\text{mac}}([1:K])}{KH(S)} \quad (27)$$

for Gaussian MACs.

## IV. MAIN RESULTS

In this section, we state our main results. For a better understanding, we first provide a high level description of the proposed approach based on the single arithmetic sum computation over the Gaussian MAC in Section IV-A. We then state our results for orthogonal Gaussian networks and Gaussian networks with multiple-access in Sections IV-B and IV-C, respectively.

### A. Main Idea

We begin this section by explaining the essence of how to compute a single arithmetic sum, i.e.,  $\{f(\mathbf{s}[j]) = \sum_{i=1}^K s_i[j]\}_{j=1}^k$ , over the Gaussian MAC with equal channel gains. For notational simplicity, we rewrite the length- $n$  time-extended input-output as  $\mathbf{y} = \sum_{i=1}^K \mathbf{x}_i + \mathbf{z}$ . We first apply compute-and-forward in [7] to transform the length- $n$  Gaussian MAC into the following length- $m$  modulo- $q$  sum channel:

$$\mathbf{y}' = \bigoplus_{i=1}^K \mathbf{x}'_i, \quad (28)$$

where  $\mathbf{x}'_i \in \mathbb{F}_q^m$ . Here  $q$  is set to be the largest prime number among  $[1 : n \log n]$  and

$$m = nC^+ \left( \frac{1}{K} + P \right) (\log q)^{-1}. \quad (29)$$

Specifically, from Theorem 2 (also see Example 2), by treating  $\mathbf{y}'$  in (28) as the desired function, we can construct the above modulo- $q$  sum channel.

Now consider the computation over the transformed modulo- $q$  sum channel. The key observation is that utilizing a small portion of input finite field elements and then computing the corresponding modulo- $q$  sum can attain the desired arithmetic sum. Furthermore, linear source coding in Lemma 1 (for this case,  $L = 1$ ) can compensate the inefficiency of utilizing only a small portion of input finite field elements by compressing the corresponding modulo- $q$  sum in a distributed manner.

Let  $g(\cdot)$  denote the mapping from a subset of integers  $[0 : q - 1]$  to the corresponding finite field  $\mathbb{F}_q$ . Define  $s'_i[j] = g(s_i[j])$  and  $U' = \bigoplus_{i=1}^K g(S_i)$ . Suppose that the  $i$ th sender observes  $\mathbf{s}'_i = [s'_i[1], \dots, s'_i[k]]^T$ , which can be obtained from  $\mathbf{s}_i$ , and the receiver wishes to compute  $\mathbf{u}' = \bigoplus_{i=1}^K \mathbf{s}'_i$ . Then, from Theorem 1, the receiver can compute  $\mathbf{u}'$  reliably for  $m$  sufficiently large (equivalently, for  $n$  sufficiently large) if

$$\frac{k}{m} \leq \frac{\log q}{H(U')}. \quad (30)$$

Hence by setting  $k = \frac{m \log q}{H(U')}$ , the computation rate

$$\begin{aligned} R &= \frac{k}{n} \\ &= \frac{m \log q}{nH(U')} \\ &= \frac{C^+ \left( \frac{1}{K} + P \right)}{H(U')} \end{aligned} \quad (31)$$

is achievable for the desired function  $\mathbf{u}'$ , where the last equality follows from (29). Since there exists  $n_0 \geq 0$  such that  $q > (p - 1)^2 K$  for all  $n \geq n_0$  ( $q$  is the largest prime number among  $[1 : n \log n]$ ), we have

$$\begin{aligned} \mathbf{u}' &= \left[ g \left( \sum_{i=1}^K s_i[1] \right), \dots, g \left( \sum_{i=1}^K s_i[k] \right) \right]^T, \\ U' &= g \left( \sum_{i=1}^K S_i \right) \end{aligned} \quad (32)$$

for  $n$  sufficiently large. Since  $g(\cdot)$  has one-to-one correspondence, the receiver can compute the arithmetic sum  $\sum_{i=1}^K \mathbf{s}_i$  from  $\mathbf{u}'$ . Finally, from the fact that  $H(g(\sum_{i=1}^K S_i)) = H(\sum_{i=1}^K S_i)$ , the achievable computation rate for the desired function  $\sum_{i=1}^K \mathbf{s}_i$  is given by

$$R = \frac{C^+ \left( \frac{1}{K} + P \right)}{H(\sum_{i=1}^K S_i)}. \quad (33)$$

*Example 4 (Arithmetic Sum of I.I.D. Binary Sources):* Suppose that  $S_i$ 's are independently and uniformly drawn from  $\{0, 1\}$  and the receiver wishes to compute  $\{f(\mathbf{s}[j]) = \sum_{i=1}^K s_i[j]\}_{j=1}^k$ . Let  $U = \sum_{i=1}^K S_i$ . Then, from (33),

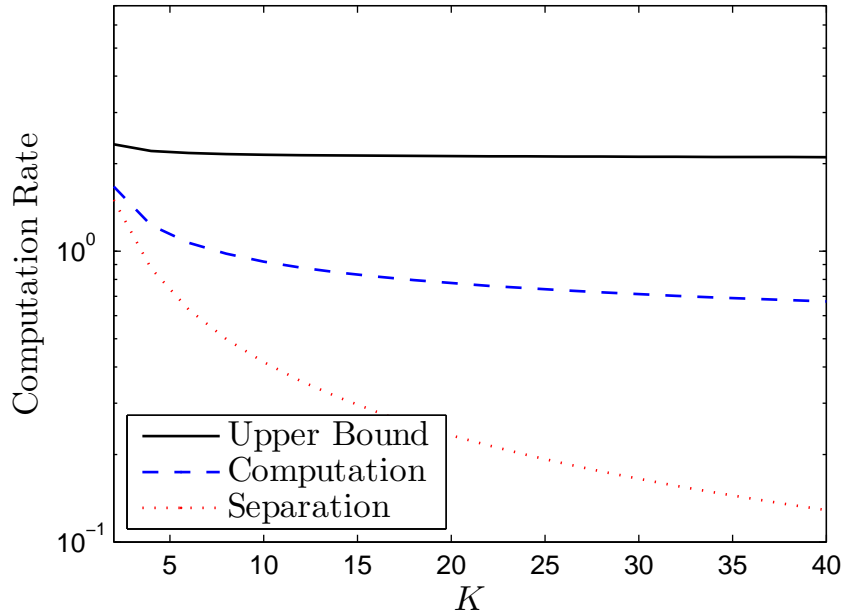


Fig. 7. Computation of  $\sum_{i=1}^K S_i$  for the  $K$ -user Gaussian MAC with equal channel gains when  $P = 15$  dB.

$R = \frac{C^+(\frac{1}{K}+P)}{H(U)}$  is achievable, where  $p_U(x) = \binom{K}{x}2^{-K}$ . On the other hand, an achievable computation rate by separation is upper bounded by  $\frac{C(K^2P)}{K}$  from (27). Lastly, the cut-set upper bound in (21) shows that an achievable computation rate is upper bounded by  $\frac{C(K^2P)}{H(U)}$ . Figure 7 plots these three rates with respect to  $K$ . Since  $H(U)$  scales as the order of  $\log K$  as  $K$  increases [42, Lemma 2.1], our computation-based rate decreases as the order of  $\log K$ , while the separation-based rate decreases almost linearly with an increasing  $K$ . Therefore, the rate gap between the computation-based and separation-based approaches becomes significant as  $K$  increases.

### B. Orthogonal Gaussian Networks

We first state our main result for orthogonal Gaussian single-hop networks and then extend it to general orthogonal Gaussian networks. We also demonstrate a class of networks that achieves the computation capacity.

1) *Single-hop networks*: Consider orthogonal Gaussian single-hop networks in which the length- $n$  time-extended input-output is given by (4). For orthogonal Gaussian networks, we abstract each Gaussian channel into the corresponding error-free bit-pipe channel using point-to-point capacity-achieving codes. Then the problem is equivalent to the distributed source coding problem. In [5, Section VI], a general achievability containing the Slepian–Wolf rate region and the Körner–Marton rate region has been proposed for the modulo-two sum computation of binary sources, which has been introduced in Section III-A. The coding scheme in [5, Section VI] can be straightforwardly generalized to more than two users and general finite field sources, and more importantly, to the arithmetic sum and type computation.

*Theorem 3 (Orthogonal Gaussian Single-Hop Networks)*: Consider the orthogonal Gaussian single-hop network. Let  $W_i$ ,  $i \in [1 : K]$ , be an auxiliary random variable that forms a Markov chain  $W_i - S_i - \{S_j, W_j\}_{j \in [1:K] \setminus \{i\}}$ . Then any computation rate satisfying

$$R \leq \frac{\sum_{i \in \Sigma} C(h_i^2 P)}{I(\{W_i\}_{i \in \Sigma}; \mathbf{S} | \{W_i\}_{i \in [1:K] \setminus \Sigma}) + |\Sigma| H(f(\mathbf{S}) | W_1, \dots, W_K)} \quad (34)$$

for all  $\Sigma \subseteq [1 : K]$  is achievable, where  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function and  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Proof*: We refer to Appendix I for the proof. ■

Notice that if we set  $W_i = \emptyset$  for all  $i \in [1 : K]$ , then Theorem 3 provides

$$R \leq \frac{\sum_{i \in \Sigma} C(h_i^2 P)}{|\Sigma| H(f(\mathbf{S}))} \quad (35)$$

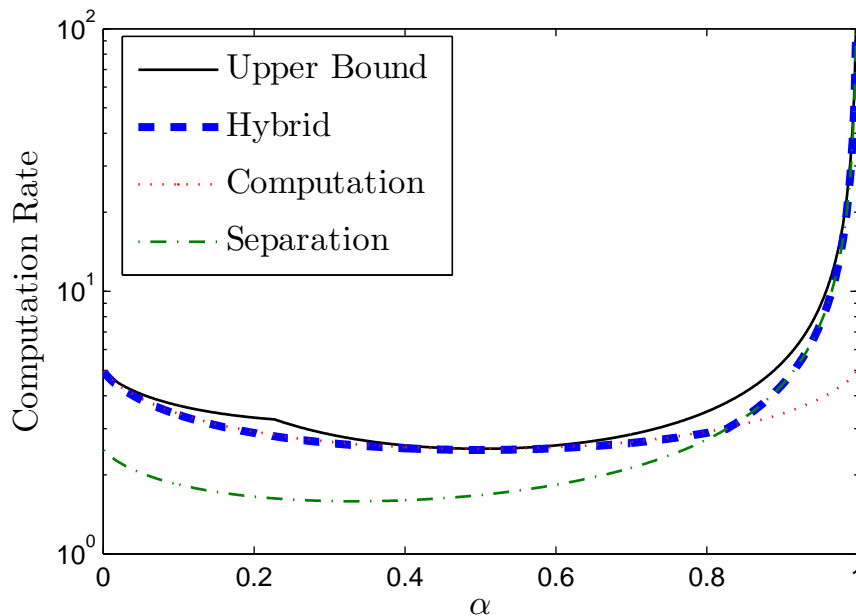


Fig. 8. Computation of  $S_1 + S_2$  for the two-user orthogonal Gaussian single-hop network with equal channel gains when  $P = 15$  dB.

for all  $\Sigma \subseteq [1 : K]$ , which corresponds to the achievable computation rate from linear source coding for computing. On the other hand, if we set  $W_i = S_i$  for all  $i \in [1 : K]$ , then Theorem 3 provides

$$R \leq \frac{\sum_{i \in \Sigma} C(h_i^2 P)}{H(\{S_i\}_{i \in \Sigma} | \{S_i\}_{i \in [1:K] \setminus \Sigma})} \quad (36)$$

for all  $\Sigma \subseteq [1 : K]$ , which corresponds to the achievable computation rate by separation.

Unlike the computation over MAC, for example, see Section IV-A, the following example shows that without multiple-access component linear source coding cannot improve the computation rate achievable by separation if the sources are independent.

*Example 5 (Independent Sources):* Suppose that the sources are independent to each other, i.e.,  $p_{\mathbf{S}}(\cdot) = \prod_{i=1}^K p_{S_i}(\cdot)$ . Then (36) yields  $R \leq \frac{\sum_{i \in \Sigma} C(h_i^2 P)}{\sum_{i \in \Sigma} H(S_i)}$ . For the arithmetic sum and type functions,  $H(f(\mathbf{S})) \geq H(S_i)$  for all  $i \in [1 : K]$  when the sources are independent. Hence, the separation-based computation always outperforms the computation based on linear source coding for independent sources.

When the sources are correlated, however, linear source coding is useful even without multiple-access component. For DSBS( $\alpha$ ), for example, the computation based on linear source coding outperforms the separation-based computation when  $\alpha$  is small, which can be verified in the following example.

*Example 6 (Arithmetic Sum of DSBS( $\alpha$ )):* Suppose that  $K = 2$ ,  $h_1 = h_2 = 1$ , and the sources follows DSBS( $\alpha$ ). The receiver wishes to compute  $\{f(\mathbf{s}[j]) = s_1[j] + s_2[j]\}_{j=1}^k$ . From Theorem 3, let  $W_1 = S_1 \oplus Z_1$  and  $W_2 = S_2 \oplus Z_2$ , where  $Z_1$  and  $Z_2$  are independent and follow  $\text{Bern}(\beta)$ . Then, due to symmetry, any computation rate satisfying

$$\begin{aligned} R &\leq \frac{C(P)}{I(W_1; S_1, S_2 | W_2) + H(S_1 + S_2 | W_1, W_2)}, \\ R &\leq \frac{2C(P)}{I(W_1, W_2; S_1, S_2) + 2H(S_1 + S_2 | W_1, W_2)} \end{aligned} \quad (37)$$

is achievable. Figure 8 plots the computation rate in (37). For comparison, we also plot the computation rates achievable by computation (setting  $W_1 = S_1$  and  $W_2 = S_2$  in (37)) and separation (setting  $W_1 = W_2 = \emptyset$ ), respectively. The cut-set upper bound is given by  $R \leq \min\{C(P)/H(Z), 2C(P)/(H(S_1 + S_2))\}$  [4]. As shown in the figure, the computation based on linear source coding is indeed helpful even if there is no multiple-access component. For this symmetric source, the hybrid approach in Theorem 3 provides the maximum of the two computation rates achievable by the computation and separation schemes.

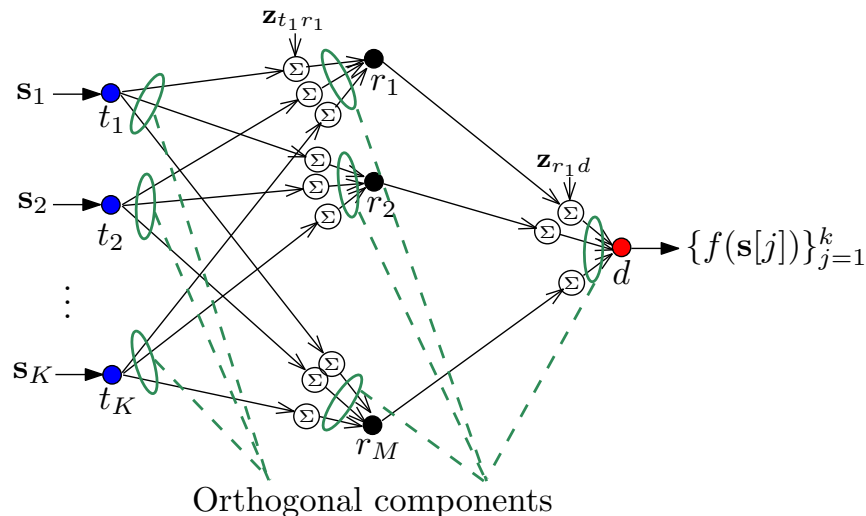


Fig. 9. An example of orthogonal Gaussian networks that achieves the computation capacity, where the channel coefficients are equal to one on all links.

2) *General networks*: Now consider general orthogonal Gaussian networks. For function computation, the previous work in [6], [7] has exploited the similarity between the channel and the desired function. Specifically, modulo- $p$  sum computation over (noisy) modulo- $p$  sum channel or Gaussian MAC has been considered. It seems that channel's multiple-access or superposition property is essentially required to compute sum-type functions more efficiently than the separation-based computation. We demonstrate that, even for orthogonal channels with no multiple-access component, relaying based on linear network coding provides an efficient end-to-end interface for function computation, which yields the following theorem.

*Theorem 4 (Orthogonal Gaussian Networks)*: Consider the orthogonal Gaussian network. Then the computation rate

$$R_{\text{ptp}} = \frac{\min_{i \in [1:K]} \bar{C}_{\text{ptp}}(\{i\})}{H(f(\mathbf{S}))} \quad (38)$$

is achievable, where  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function and  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Proof*: We refer to Section VI-A for the proof. ■

For convenience, denote the cut-set upper bound in (20) for  $\Sigma = [1 : K]$  as

$$R_{\text{ptp}}^{(u)} := \frac{\bar{C}_{\text{ptp}}([1 : K])}{H(f(\mathbf{S}))} \quad (39)$$

and the achievable computation rate in Theorem 4 as

$$R_{\text{ptp}}^{(l)} := \frac{\min_{i \in [1:K]} \bar{C}_{\text{ptp}}(\{i\})}{H(f(\mathbf{S}))}. \quad (40)$$

*Remark 6 (Computation Capacity)*: From (39) and (40), the gap between  $R_{\text{ptp}}^{(u)}$  and  $R_{\text{ptp}}^{(l)}$  is zero if the condition  $\bar{C}_{\text{ptp}}([1 : K]) = \min_{i \in [1:K]} \bar{C}_{\text{ptp}}(\{i\})$  is satisfied, which characterizes the computation capacity. Figure 9 is an example of this class of networks. Basically, any layered network with equal channel gains can be an example.

*Remark 7 (Bit-Pipe Wired Networks)*: As mentioned in Remark 4, we can easily interpret the results in Theorems 3 and 4 for bit-pipe wired networks. Hence, Remark 6 also provides the computation capacity for a certain class of bit-pipe wired networks, which closes the gap between the lower and upper bounds in [28], [35] in the case of the arithmetic sum function computation.

### C. Gaussian Networks With Multiple-Access

We extend the idea presented in Section IV-A to the computation of multiple weighted arithmetic sums, which contains the type function, and also to a general network topology in the following theorem. For the achievability, we first abstract each multiple-access component by compute-and-forward and then apply linear network coding at

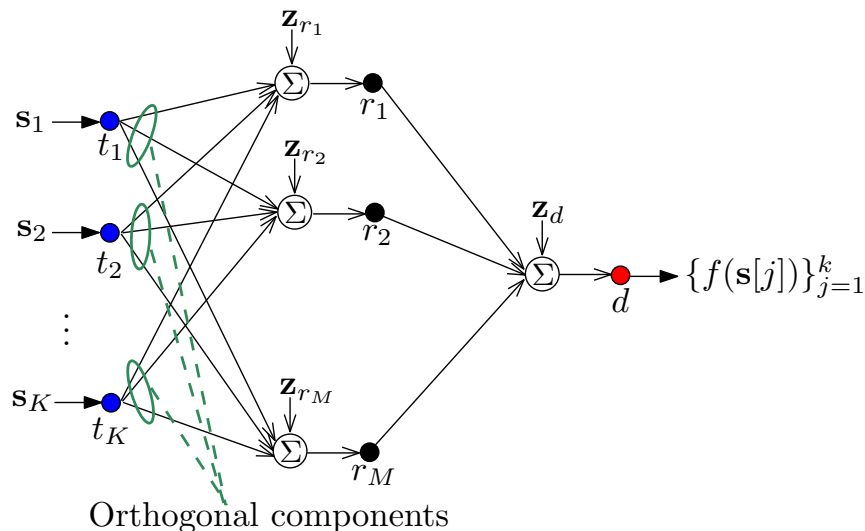


Fig. 10. An example of Gaussian networks with multiple-access that satisfies the condition in Corollary 1, where the channel coefficients are equal to one on all links.

each relay node to convert the original Gaussian network with multiple-access into the end-to-end modulo- $q$  sum channel.

*Theorem 5 (Gaussian Networks With Multiple-Access):* Consider the Gaussian network with multiple-access. Then the computation rate

$$R_{\text{mac}} = \frac{\min_{i \in [1:K]} C_{\text{mac}}^+(\{i\})}{H(f(\mathbf{S}))} \quad (41)$$

is achievable, where

$$C_{\text{mac}}^+(\{i\}) = \min_{\Omega \in \Lambda(\{i\})} \sum_{v \in \Omega^c} \mathbf{1}_{\Gamma_{\text{in}}(v) \cap \Omega \neq \emptyset} C^+ \left( \frac{1}{|\Gamma_{\text{in}}(v)|} + \min_{u \in \Gamma_{\text{in}}(v)} h_{u,v}^2 P \right). \quad (42)$$

Here  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function and  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Proof:* We refer to Section VI-B for the proof.  $\blacksquare$

Similarly, denote the cut-set upper bound in (21) for  $\Sigma = [1 : K]$  as

$$R_{\text{mac}}^{(u)} := \frac{\bar{C}_{\text{mac}}([1 : K])}{H(f(\mathbf{S}))} \quad (43)$$

and the achievable computation rate in Theorem 5 as

$$R_{\text{mac}}^{(l)} := \frac{\min_{i \in [1:K]} C_{\text{mac}}^+(\{i\})}{H(f(\mathbf{S}))}. \quad (44)$$

Then the following corollary holds.

*Corollary 1 (Approximate Computation Capacity):* If  $h_{u,v}$  are the same for all  $u \in \Gamma_{\text{in}}(v)$  and  $\bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} \bar{C}_{\text{mac}}(\{i\}) \leq c_1 |V| \log |V|$  for some constant  $c_1 > 0$ , then

$$R_{\text{mac}}^{(u)} - R_{\text{mac}}^{(l)} \leq \frac{c_2 |V| \log |V|}{H(f(\mathbf{S}))} \quad (45)$$

for any power  $P$ , where  $c_2 > 0$  is some constant.

*Proof:* We refer to VI-B for the proof.  $\blacksquare$

*Remark 8 (Bounds on the Capacity Gap):* First of all, the gap in Corollary 1 does not depend on  $P$  and therefore provides a universal performance guarantee for any  $P$ . Also, since  $H(f(\mathbf{S}))$  is an increasing function of  $K \leq |V|$ , the gap in Corollary 1 increases at most as  $|V| \log |V|$ . Figure 10 is an example of the class of networks satisfying the condition in Corollary 1. Basically, any layered network with equal channel gains can be an example.

*Remark 9 (Tighter Bound for the Gaussian MAC):* For the single-hop case, i.e., the Gaussian MAC, we can easily tighten the gap in Corollary 1. Specifically, if  $C\left(\left(\sum_{i=1}^K h_i\right)^2 P\right) - C\left(\frac{1}{K} + \min_{i \in [1:K]} h_i^2 P\right) \leq c_3 \log K$  for some constant  $c_3 > 0$ , then

$$R_{\text{mac}}^{(u)} - R_{\text{mac}}^{(l)} \leq \frac{c_4 \log K}{H(f(\mathbf{S}))} \quad (46)$$

for any power  $P$ , where  $c_4 > 0$  is some constant.

## V. COMPUTATION OVER LINEAR FINITE FIELD NETWORKS

In this section, we introduce the following linear finite field network and show how to compute the desired function over the considered network. Specifically, we first explain the computation over the modulo- $q$  sum channel in Section V-A. Then we introduce the network transformation method that converts a general linear finite field network into the modulo- $q$  sum channel in Section V-B. The computation coding and transformation method presented in this section will be used for proving the results for general Gaussian networks with orthogonal components in Section VI.

Again, we assume a network represented by a directed graph  $G = (V, E)$  and the same source and desired function in Definitions 1 to 6. For the considered linear finite field network model, the input-output at time  $t$  is given by

$$y_v^{(t)} = \bigoplus_{u \in \Gamma_{in}(v)} h_{u,v} x_u^{(t)}, \quad (47)$$

where  $x_u^{(t)} \in \mathbb{F}_q^{\alpha_u}$ ,  $y_v^{(t)} \in \mathbb{F}_q^{\beta_v}$ , and  $h_{u,v} \in \mathbb{F}_q^{\beta_v \times \alpha_u}$ . Here  $q$  is assumed to be a prime number. Then the length- $n$  time-extended input-output is represented as

$$\mathbf{y}_v = \bigoplus_{u \in \Gamma_{in}(v)} \mathbf{H}_{u,v} \mathbf{x}_u, \quad (48)$$

where  $\mathbf{x}_u = [x_u^{(1)T}, \dots, x_u^{(n)T}]^T \in \mathbb{F}_q^{n\alpha_u}$ ,  $\mathbf{y}_v = [y_v^{(1)T}, \dots, y_v^{(n)T}]^T \in \mathbb{F}_q^{n\beta_v}$ , and  $\mathbf{H}_{u,v} \in \mathbb{F}_q^{n\beta_v \times n\alpha_u}$  denotes the block diagonal matrix consisting of  $h_{u,v}$  at each block diagonal element.

*Remark 10:* Without loss of generality, we can assume that  $\alpha_v = \beta_v$  is the same for all  $v \in V$  since the effect of different values of  $\alpha_v$  and  $\beta_v$  can be equivalently reflected by the channel matrix  $h_{u,v}$ . However, we allow different values of  $\alpha_v$  and  $\beta_v$  in this section for easy explanation of the transformation from the Gaussian network model in Section VI.

*Remark 11:* The considered linear finite field network includes the bit-pipe wired channel model and the linear finite field deterministic model proposed in [32].

### A. Computation Over the Modulo- $q$ Sum Channel

As a special case of the considered linear finite field network, first consider the modulo- $q$  sum channel. The following lemma shows how to compute the arithmetic sum or type function over the modulo- $q$  sum channel when the field size  $q$  is large enough. This lemma is of crucial importance to prove the main theorems in the paper. The key observation is that utilizing a small portion of finite field elements and then computing the corresponding modulo- $q$  sum can attain the desired function. Furthermore, linear source coding in Lemma 1 can compensate the inefficiency of utilizing only a small portion of finite field elements and, as a result, achieves the optimal computation rate when the field size  $q$  is large enough.

*Lemma 2 (Computation Over the Modulo- $q$  Sum Channel):* Consider the computation over the modulo- $q$  sum channel in which the length- $n$  modulo- $q$  sum channel is given as

$$\mathbf{y}' = \bigoplus_{i=1}^K \mathbf{x}'_i, \quad (49)$$

where  $\mathbf{x}'_i \in \mathbb{F}_q^n$  and  $q$  is a prime number. If  $q > (p-1)^2 K$ , then the computation capacity is given by

$$R = \frac{\log q}{H(f(\mathbf{S}))}, \quad (50)$$



where  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function and  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Proof:* Let  $g(\cdot)$  denote the mapping from a subset of integers  $[0 : q - 1]$  to the corresponding finite field  $\mathbb{F}_q$ .

First consider the arithmetic sum computation. Define  $s'_i[j] = g(s_i[j])$ ,  $a'_{li} = g(a_{li})$ , and  $U'_l = \bigoplus_{i=1}^K a'_{li} g(S_i)$  for  $i \in [1 : K]$  and  $l \in [1 : L]$ . Suppose that the  $i$ th sender observes  $\mathbf{s}'_i = [s'_i[1], \dots, s'_i[k]]^T$ , which can be obtained from  $\mathbf{s}_i$ , and the receiver wishes to compute  $(\mathbf{u}'_1, \dots, \mathbf{u}'_L) = \left( \bigoplus_{i=1}^K a'_{1i} \mathbf{s}'_i, \dots, \bigoplus_{i=1}^K a'_{Li} \mathbf{s}'_i \right)$ . From Theorem 1, the receiver can compute  $(\mathbf{u}'_1, \dots, \mathbf{u}'_L)$  reliably for  $n$  sufficiently large if

$$\frac{k}{n} \leq \frac{\log q}{H(U'_1, \dots, U'_L)}. \quad (51)$$

Hence by setting  $k = \frac{n \log q}{H(U'_1, \dots, U'_L)}$ , the computation rate

$$\begin{aligned} R &= \frac{k}{n} \\ &= \frac{\log q}{H(U'_1, \dots, U'_L)} \end{aligned} \quad (52)$$

is achievable for the desired function  $(\mathbf{u}'_1, \dots, \mathbf{u}'_L)$ . From the condition  $q > (p-1)^2 K$ , we have

$$\mathbf{u}'_l = \left[ g \left( \sum_{i=1}^K a_{li} s_i[1] \right), \dots, g \left( \sum_{i=1}^K a_{li} s_i[k] \right) \right]^T, \quad (53)$$

$$U'_l = g \left( \sum_{i=1}^K a_{li} S_i \right) = g(U_l). \quad (54)$$

Since  $g(\cdot)$  has one-to-one correspondence, from (53), the receiver can compute  $\left( \sum_{i=1}^K a_{1i} \mathbf{s}_i, \dots, \sum_{i=1}^K a_{Li} \mathbf{s}_i \right)$  from  $(\mathbf{u}'_1, \dots, \mathbf{u}'_L)$ . Finally, from the fact that  $H((U'_1, \dots, U'_L)) = H(U_1, \dots, U_L)$ , which can be verified from (54), the achievable computation rate for  $\left( \sum_{i=1}^K a_{1i} \mathbf{s}_i, \dots, \sum_{i=1}^K a_{Li} \mathbf{s}_i \right)$  is given by (50).

Now consider the type computation. Define  $b_{li}[j] = \mathbf{1}_{s_i[j]=l}$ . Let  $b'_{li}[j] = g(b_{li}[j])$  and  $\mathbf{b}'_{li} = [b'_{li}[1], \dots, b'_{li}[k]]^T$ . Suppose that the  $i$ th sender observes  $\mathbf{b}'_{li}$ , which can be obtained from  $\mathbf{s}_i$ , and the receiver wishes to compute  $\{\mathbf{u}''_l = \bigoplus_{i=1}^K \mathbf{b}'_{li}\}_{l=0}^{p-1}$ . From Theorem 1, the receiver can compute  $\{\mathbf{u}''_l\}_{l=1}^L$  reliably for  $n$  sufficiently large if

$$\frac{k}{n} \leq \frac{\log q}{H(B'_0, \dots, B'_{p-1})}, \quad (55)$$

where  $B'_l = \bigoplus_{i=1}^K B'_{li}$  and  $B'_{li} = g(\mathbf{1}_{S_i=l})$ . Hence by setting  $k = \frac{n \log q}{H(B'_0, \dots, B'_{p-1})}$ , the computation rate

$$\begin{aligned} R &= \frac{k}{n} \\ &= \frac{\log q}{H(B'_0, \dots, B'_{p-1})} \end{aligned} \quad (56)$$

is achievable for the desired function  $\{\mathbf{u}''_l\}_{l=0}^{p-1}$ . From the condition  $q > (p-1)^2 K$ , we have

$$\begin{aligned} \mathbf{u}''_l &= \left[ g \left( \sum_{i=1}^K b_{li}[1] \right), \dots, g \left( \sum_{i=1}^K b_{li}[k] \right) \right]^T, \\ B'_l &= g \left( \sum_{i=1}^K \mathbf{1}_{S_i=l} \right) = g(B_l). \end{aligned} \quad (57)$$

Since  $g(\cdot)$  has one-to-one correspondence, the receiver can compute  $\{b_0(\mathbf{s}[j]), \dots, b_{p-1}(\mathbf{s}[j])\}_{j=1}^k$  from  $\{\mathbf{u}''_l\}_{l=0}^{p-1}$ . Finally, from the fact that  $H(B'_0, \dots, B'_{p-1}) = H(B_0, \dots, B_{p-1})$ , the achievable computation rate for the type function is given by (50).

The converse for both cases can be easily shown from the same cut-set argument in Section III-D, which completes the proof.  $\blacksquare$

*Remark 12 (Computation of Multiple Arithmetic Sums):* Notice that the channel model in (49) is exactly the same as in (28) except that  $q$  is fixed in this section. Hence Lemma 2 extends the computation of a single arithmetic sum in Section IV-A to the computation of multiple weighted arithmetic sums (the type function can be represented as multiple arithmetic sums, see Definition 2 and Remark 1).

### B. Computation Via Network Transformation

We now describe how to convert a general linear finite field network into the modulo- $q$  sum channel in (49) in the following lemma. The basic principle is similar to those proposed in [8] in the sense that linear network coding is applied at each relay node to construct an end-to-end linear finite field channel and then precoder at each sender to convert the end-to-end channel into the linear finite field channel in (49).

To describe the following lemma, for  $\Sigma \subseteq [1 : K]$ , let  $H_\Omega(\Sigma)$  denote the transfer matrix associated with the cut  $\Omega \in \Lambda(\Sigma)$  on  $G(\Sigma)$ .

*Lemma 3 (Transform Into the Modulo- $q$  Sum Channel):* Consider the linear finite field network in which the length- $n$  time extended input–output is given by (48). For  $n$  sufficiently large, the network can be transformed into the following modulo- $q$  sum channel:

$$\mathbf{y}'_d = \bigoplus_{i=1}^K \mathbf{x}'_{t_i}, \quad (58)$$

where  $\mathbf{x}'_{t_i} \in \mathbb{F}_q^{n \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\}))}$  for all  $i \in [1 : K]$ .

*Proof:* We refer to Appendix II for the proof. ■

Based on Lemmas 2 and 3, we characterize the computation capacity when  $q > (p-1)^2 K$  in the following theorem.

*Theorem 6 (Linear Finite Field Networks):* Consider the linear finite field network in which the length- $n$  time-extended input–output is given as in (48). If  $q > (p-1)^2 K$  and

$$\min_{\Omega \in \Lambda([1:K])} \text{rank}(H_\Omega([1:K])) = \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})), \quad (59)$$

then the computation capacity is given by

$$R = \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\}))}{H(f(\mathbf{S}))}, \quad (60)$$

where  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function and  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Proof:* The achievability follows by the network transformation using Lemma 3 and then the computation over the transformed modulo- $q$  sum channel using Lemma 2. The converse follows by the cut-set upper bound showing that  $R \leq \frac{\min_{\Omega \in \Lambda([1:K])} \text{rank}(H_\Omega([1:K]))}{H(f(\mathbf{S}))}$  and the condition in (59), which completes the proof. ■

*Remark 13 (Type Computation):* For the type computation, the condition  $q > K$  is enough for proving Lemma 2 and Theorem 6.

*Remark 14 (Computation Over Gaussian Networks):* We will apply the same computation coding and network transformation in Lemmas 2 and 3 after converting Gaussian networks into linear finite field networks in Section VI. For this case, we set  $q$  arbitrarily large and, as a result, the condition  $q > (p-1)^2 K$  in Lemma 2 and Theorem 6 disappears for Gaussian networks.

## VI. COMPUTATION OVER GAUSSAIN NETWORKS WITH ORTHOGONAL COMPONENTS

We are now ready to prove Theorems 4 and 5 and Corollary 1, which are about the computation over general Gaussian networks with orthogonal components.

### A. Orthogonal Gaussian Networks

Consider orthogonal Gaussian networks in which the length- $n$  time-extended input–output is given as in (2). In the following, we prove the achievable computation rate in Theorem 4. The achievability follows the network abstraction based on capacity-achieving point-to-point channel codes and then transformation into the modulo- $q$  sum channel via linear network coding at each relay (the same transformation used in Lemma 3). Finally, we apply linear source coding for computation over the transformed modulo- $q$  sum channel as the same manner used in Lemma 2.

1) *Proof of Theorem 4:* By applying capacity-achieving point-to-point channel codes, for  $n$  sufficiently large, the length- $n$  time extended orthogonal Gaussian channel in (2) can be transformed into

$$\mathbf{y}'_{u,v} = \mathbf{x}'_{u,v} \quad (61)$$

for all  $(u, v) \in E$ , where  $\mathbf{x}_{u,v} \in \mathbb{F}_q^{m_{u,v}}$ . Here  $m_{u,v} \log q \leq nC(h_{u,v}^2 P)$  should be satisfied. Hence we set  $q = K + 1$  and  $m_{u,v} = nC(h_{u,v}^2 P)(\log q)^{-1}$ .

In order to use Theorem 3, we define  $\mathbf{x}'_u = \{\mathbf{x}_{u,v}\}_{v \in \Gamma_{out}(u)}$  and  $\mathbf{y}'_v = \{\mathbf{y}_{u,v}\}_{u \in \Gamma_{in}(v)}$  and represent the input-output as

$$\mathbf{y}'_v = \bigoplus_{u \in \Gamma_{in}(v)} \mathbf{H}'_{u,v} \mathbf{x}'_u. \quad (62)$$

Here  $\mathbf{H}'_{u,v}$  is determined by the original channel condition in (61). Notice that the input-output in (62) is the same linear finite field model considered in Section V, see (47). Then we can treat  $\mathbf{y}'_v$  and  $\mathbf{x}'_u$  as ‘super-symbol’ and apply the multi-letter coding over these length- $\eta$  super-symbols, that is, length- $\eta n$  symbols. Hence, from Lemma 3, we can transform the linear finite field network with the input-output (62) into the following modulo- $q$  sum channel:

$$\mathbf{y}'_d = \bigoplus_{i=1}^K \mathbf{x}'_{t_i}, \quad (63)$$

where  $\mathbf{x}'_{t_i} \in \mathbb{F}_q^{\eta \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\}))}$  for all  $i \in [1 : K]$ .

Now consider the computation over the above modulo- $q$  sum channel. Since  $q = K + 1 > K$ , from Lemma 2, the receiver can compute the desired function reliably for  $\eta$  sufficiently large if

$$\frac{k}{\eta \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\}))} \leq \frac{\log q}{H(f(\mathbf{S}))}. \quad (64)$$

Hence by setting  $k = \frac{\eta \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})) \log q}{H(f(\mathbf{S}))}$ , the computation rate

$$\begin{aligned} R &= \frac{k}{\eta n} \\ &= \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})) \log q}{nH(f(\mathbf{S}))} \\ &\stackrel{(a)}{=} \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \sum_{(u,v) \in E(\{i\}), u \in \Omega, v \in \Omega^c} m_{u,v} \log q}{nH(f(\mathbf{S}))} \\ &\stackrel{(b)}{=} \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \sum_{(u,v) \in E(\{i\}), u \in \Omega, v \in \Omega^c} C(h_{u,v}^2 P)}{H(f(\mathbf{S}))} \\ &\stackrel{(c)}{=} \frac{\min_{i \in [1:K]} \bar{C}_{\text{ptp}}(\{i\})}{H(f(\mathbf{S}))} \end{aligned} \quad (65)$$

is achievable, where (a) follows since  $\text{rank}(H_\Omega(\{i\}))$  assuming the channel (62) is the same as that assuming the channel (61) and  $\text{rank}(H_\Omega(\{i\})) = \sum_{(u,v) \in E(\{i\}), u \in \Omega, v \in \Omega^c} m_{u,v}$ , (b) follows since  $m_{u,v} = nC(h_{u,v}^2 P)(\log q)^{-1}$ , and (c) follows from the definition (18). In conclusion, Theorem 4 holds.

## B. Gaussian Networks With Multiple-Access

Consider Gaussian networks with multiple-access in which the length- $n$  time-extended input-output is given as in (3). In the following, we prove the achievable computation rate in Theorem 5 and Corollary 1. The achievability follows the network abstraction based on compute-and-forward in Theorem 2 and the rest of the procedure is similar to that in Section VI-A.

1) *Proof of Theorem 5:* For each node  $v \in V$ , suppose that node  $u \in \Gamma_{in}(v)$  observes  $\mathbf{x}'_{u,v} \in \mathbb{F}_q^{m_v}$  and node  $v$  wishes to decode  $\bigoplus_{u \in \Gamma_{in}(v)} \mathbf{x}'_{u,v}$ . Let  $\mathbf{x}'_{u,v}{}^{\text{lattice}}(\mathbf{x}'_{u,v})$  denote a dither-added transmit lattice point from node  $u$  to node  $v$  for the compute-and-forward framework in [7], which satisfies the power constraint  $P$ . Then node  $u$  transmits  $\mathbf{x}_{u,v}(\mathbf{x}'_{u,v}) = \frac{\min_{u \in \Gamma_{in}(v)} \{h_{u,v}\}}{h_{u,v}} \mathbf{x}'_{u,v}{}^{\text{lattice}}(\mathbf{x}'_{u,v})$  to node  $v$ . We can equivalently interpret that  $h_{u,v} = 1$  and the average power constraint from node  $u$  to node  $v$  is given as  $\min_{u \in \Gamma_{in}(v)} \{h_{u,v}^2\} P$ . From Theorem 2 (see also Example 2), node  $v$  can decode  $\bigoplus_{u \in \Gamma_{in}(v)} \mathbf{x}'_{u,v}$  reliably for  $n$  sufficiently large if  $m_v \leq nC^+ \left( \frac{1}{|\Gamma_{in}(v)|} + \min_{u \in \Gamma_{in}(v)} h_{u,v}^2 P \right) (\log q)^{-1}$  and  $q$  is an increasing function of  $n$  satisfying that  $q \rightarrow \infty$  as  $n \rightarrow \infty$ . Therefore, by treating  $\mathbf{x}'_{u,v}$  as the channel input from node  $u$  to node  $v$  and  $\bigoplus_{u \in \Gamma_{in}(v)} \mathbf{x}'_{u,v}$  as the channel output of node  $v$ , for  $n$  sufficiently large, we can transform each length- $n$  Gaussian multiple-access component (3) into the following length- $m_v$  modulo- $q$  sum channel:

$$\mathbf{y}'_v = \bigoplus_{u \in \Gamma_{in}(v)} \mathbf{x}'_{u,v}, \quad (66)$$

where  $\mathbf{x}'_{u,v} \in \mathbb{F}_q^{m_v}$ . Here  $q$  is the largest prime number among  $[1 : n \log n]$  and

$$m_v = nC^+ \left( \frac{1}{|\Gamma_{in}(v)|} + \min_{u \in \Gamma_{in}(v)} h_{u,v}^2 P \right) (\log q)^{-1}. \quad (67)$$

We again define  $\mathbf{x}'_u = \{\mathbf{x}_{u,v}\}_{v \in \Gamma_{out}(u)}$  and represent the input-output as

$$\mathbf{y}'_v = \sum_{u \in \Gamma_{in}(v)} \mathbf{H}'_{u,v} \mathbf{x}'_u. \quad (68)$$

Here  $\mathbf{H}'_{u,v}$  is determined by the original channel condition in (66). Then, in the same manner from (62) to (65), we can apply multi-letter coding and, from Lemmas 2 and 3, the computation rate

$$\begin{aligned} R &= \frac{k}{\eta n} \\ &= \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})) \log q}{nH(f(\mathbf{S}))} \\ &\stackrel{(a)}{=} \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \sum_{v \in \Omega^c} \mathbf{1}_{\Gamma_{in}(v) \cap \Omega \neq \emptyset} m_v \log q}{nH(f(\mathbf{S}))} \\ &\stackrel{(b)}{=} \frac{\min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \sum_{v \in \Omega^c} \mathbf{1}_{\Gamma_{in}(v) \cap \Omega \neq \emptyset} C^+ \left( \frac{1}{|\Gamma_{in}(v)|} + \min_{u \in \Gamma_{in}(v)} h_{u,v}^2 P \right)}{H(f(\mathbf{S}))} \end{aligned} \quad (69)$$

is achievable, where (a) follows since  $\text{rank}(H_\Omega(\{i\}))$  assuming the channel (68) is the same as that assuming the channel (66) and  $\text{rank}(H_\Omega(\{i\})) = \sum_{v \in \Omega^c} \mathbf{1}_{\Gamma_{in}(v) \cap \Omega \neq \emptyset} m_v$  and (b) follows from (67). Note that  $q$ , which is the largest prime number among  $[1 : n \log n]$ , becomes arbitrarily large as  $n \rightarrow \infty$  and, as a result, satisfies the condition  $q > (p-1)^2 K$  in Lemma 2 for  $n$  sufficiently large. In conclusion, Theorem 5 holds.

2) *Proof of Corollary 1:* Denote

$$i^* = \arg \min_{i \in [1:K]} C_{\text{mac}}^+(\{i\}) \quad (70)$$

and

$$\Omega^* = \arg \min_{\Omega \in \Lambda(\{i^*\})} \sum_{v \in \Omega^c} \mathbf{1}_{\Gamma_{in}(v) \cap \Omega \neq \emptyset} C^+ \left( \frac{1}{|\Gamma_{in}(v)|} + \min_{u \in \Gamma_{in}(v)} h_{u,v}^2 P \right). \quad (71)$$

Then

$$\begin{aligned}
& H(f(\mathbf{S}))(R_{\text{mac}}^{(u)} - R_{\text{mac}}^{(l)}) \\
&= \bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} C_{\text{mac}}^+(\{i\}) \\
&\stackrel{(a)}{\leq} \bar{C}_{\text{mac}}(\{i^*\}) - C_{\text{mac}}^+(\{i^*\}) + \left( \bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} \bar{C}_{\text{mac}}(\{i\}) \right) \\
&\stackrel{(b)}{\leq} \sum_{v \in \Omega^{*c}} \mathsf{C} \left( \left( \sum_{u \in \Gamma_{\text{in}}(v), u \in \Omega^*} h_{u,v} \right)^2 P \right) - \sum_{v \in \Omega^{*c}} \mathbf{1}_{\Gamma_{\text{in}}(v) \cap \Omega^* \neq \emptyset} \mathsf{C}^+ \left( \frac{1}{|\Gamma_{\text{in}}(v)|} + \min_{u \in \Gamma_{\text{in}}(v)} h_{u,v}^2 P \right) \\
&\quad + \left( \bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} \bar{C}_{\text{mac}}(\{i\}) \right) \\
&\stackrel{(c)}{\leq} \sum_{v \in \Omega^{*c}} \mathbf{1}_{\Gamma_{\text{in}}(v) \cap \Omega^* \neq \emptyset} (\mathsf{C}(|V|^2 h_{u,v}^2 P) - \mathsf{C}^+(h_{u,v}^2 P)) \\
&\quad + \left( \bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} \bar{C}_{\text{mac}}(\{i\}) \right) \\
&\stackrel{(d)}{\leq} |V|(\log(|V|^2) + 1) + \left( \bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} \bar{C}_{\text{mac}}(\{i\}) \right), \tag{72}
\end{aligned}$$

where (a) follows from the definition of  $i^*$ , (b) follows from the definition of  $\Omega^*$ , (c) follows from the condition that  $h_{u,v}$  are the same for all  $u \in \Gamma_{\text{in}}(v)$ , and (d) follows since  $\mathsf{C}(x) - \mathsf{C}^+(x) \leq 1$  for all  $x \geq 0$ . Finally, from the condition  $\bar{C}_{\text{mac}}([1 : K]) - \min_{i \in [1:K]} \bar{C}_{\text{mac}}(\{i\}) \leq c_1 |V| \log |V|$ , we have (45), which completes the proof.

## VII. EXTENSIONS

In this section, we apply our computation code to other interesting scenarios for computing over Gaussian networks.

### A. Modulo- $p$ Sum Computation

Consider the modulo- $p$  sum computation over Gaussian networks with orthogonal components. In [6], [8], achievable computation rates have been derived assuming an arbitrarily large source field size, i.e.,  $p \rightarrow \infty$  as the block length  $n$  increases. When the source field size is fixed, it is hard to apply the previous work to find a better computation rate than the separation-based computation. From our framework, on the other hand, one naive approach is for the fusion center to first compute the corresponding arithmetic sum over networks and then take the modulo- $p$  operation in order to obtain the desired modulo- $p$  sum. Hence the achievable computation rates for the arithmetic sum in this paper can also be achievable computation rates for the corresponding modulo- $p$  sum. Obviously, this approach is not optimal but we can easily find examples that it outperforms the separation-based computation.

### B. Superposition Approach for Unequal Channel Coefficients

One drawback of the achievability in Theorem 5 is when the channel coefficients have different values. For the single-hop case, i.e., the Gaussian MAC in which the length- $n$  time-extended input-output is given by (5), Theorem 5 provides

$$R = \frac{\mathsf{C}^+ \left( \frac{1}{K} + \min_{i \in [1:K]} h_i^2 P \right)}{H(f(\mathbf{S}))}. \tag{73}$$

As shown in (73), the achievable computation rate is bounded by the minimum of the channel gains. In order to achieve computation rates scalable with  $P$  from the compute-and-forward framework [7], the transmit power of each lattice should be reduced to let the received power for each lattice be the same at the receiver side, resulting  $\min_{i \in [1:K]} h_i^2$  in (73). An improved computation rate is achievable by the superposition of multiple codes and then

allocating residual transmit power to high layer codes. The following theorem provides an improved computation rate for the two-user case.

*Theorem 7:* Consider the 2-user Gaussian MAC with  $h_2^2 \geq h_1^2$ . Then the computation rate satisfying

$$\begin{aligned} R &\leq \frac{C^+ \left( \frac{1}{2} + h_1^2 P \right)}{H(S_1|S_2)}, \\ R &\leq \frac{C^+ \left( \frac{1}{2} + h_1^2 P \right)}{H(f(\mathbf{S}))} + \frac{H(f(\mathbf{S})) - H(S_1|S_2)}{H(S_2)} \frac{C \left( \frac{(h_2^2 - h_1^2)P}{1 + 2h_1^2 P} \right)}{H(f(\mathbf{S}))} \end{aligned} \quad (74)$$

is achievable.

*Proof:* Let  $\mathbf{x}_i^{\text{lattice}}$  denote a dither-added transmit signal from lattice codebook of the  $i$ th sender for compute-and-forward in [7], which satisfies the power constraint  $P$ . Let  $\mathbf{x}^{\text{random}}$  denote the transmit signal from a capacity-achieving point-to-point Gaussian codebook, which satisfies the power constraint  $P$ . Then the first sender transmits  $\mathbf{x}_1 = \mathbf{x}_1^{\text{lattice}}$  and the second sender transmits  $\mathbf{x}_2 = \frac{h_1}{h_2} \mathbf{x}_2^{\text{lattice}} + \frac{\sqrt{h_2^2 - h_1^2}}{h_2} \mathbf{x}^{\text{random}}$ . Let  $R_2$  denote the message rate delivered by  $\mathbf{x}^{\text{random}}$ . The receiver first decodes the message delivered by  $\mathbf{x}^{\text{random}}$  that is reliably decodable if  $R_2 \leq C \left( \frac{(h_2^2 - h_1^2)P}{1 + 2h_1^2 P} \right)$ . The receiver then subtracts  $\mathbf{x}^{\text{random}}$  from the received signal. Then, from the same argument in the proof of Theorem 5, we can construct the modulo- $q$  sum channel in (49) with  $m = nC^+ \left( \frac{1}{K} + h_1^2 P \right) (\log q)^{-1}$ , where  $q$  is set to be the largest prime number among  $[1 : n \log n]$ . By utilizing the length  $nR_1(\log q)^{-1}$  of this modulo- $q$  sum channel, the first sender is able to transmit its message at the rate of  $R_1$  if  $R_1 \leq C^+ \left( \frac{1}{2} + h_1^2 P \right)$ . The remaining modulo- $q$  sum channel has the length of  $n \left( C^+ \left( \frac{1}{2} + h_1^2 P \right) - R_1 \right) (\log q)^{-1}$ .

For the bit-pipe channels with the rates  $R_1$  and  $R_2$ , we apply Slepian–Wolf source coding to deliver two sources separately, then compute the desired function. For the remaining modulo- $q$  sum channel, we apply linear source coding for computation in Lemma 2. This approach achieves the compute rate represented by the following rate constraints:

$$\begin{aligned} R &= R' + R'', \\ R_1 &\leq C^+ \left( \frac{1}{2} + h_1^2 P \right), \\ R_2 &\leq C \left( \frac{(h_2^2 - h_1^2)P}{1 + 2h_1^2 P} \right), \\ R' &\leq \frac{R_1}{H(S_1|S_2)}, \\ R' &\leq \frac{R_2}{H(S_2|S_1)}, \\ R' &\leq \frac{R_1 + R_2}{H(S_1, S_2)}, \\ R'' &\leq \frac{C^+ \left( \frac{1}{2} + h_1^2 P \right) - R_1}{H(f(\mathbf{S}))}. \end{aligned} \quad (75)$$

After Fourier–Motzkin elimination, we have (74), which completes the proof.  $\blacksquare$

*Example 7 (Arithmetic Sum of I.I.D. Binary Sources):* Suppose that  $K = 2$ ,  $h_1 = 1$ ,  $h_2 \geq 1$ , and  $S_1, S_2$  are independently and uniformly drawn from  $\{0, 1\}$ . The receiver wishes to compute  $\{f(\mathbf{s}[j]) = s_1[j] + s_2[j]\}_{j=1}^k$ . Figure 11 plots the achievable computation rate by the superposition in Theorem 7. As  $h^2$  increases, the separation-based computation outperforms the computation scheme in Theorem 5 and the gap to the cut-set upper bound decreases. As shown in the figure, the superposition approach can attain both of the computation in Theorem 5 and the separation-based computation.

*Remark 15 (Multiple Layering for More Than Two Users):* For more than two users, we can apply the same superposition approach by layering multiple lattice codes. We refer to [8, Section III. F] for the detailed lattice code construction and encoding, decoding procedure.

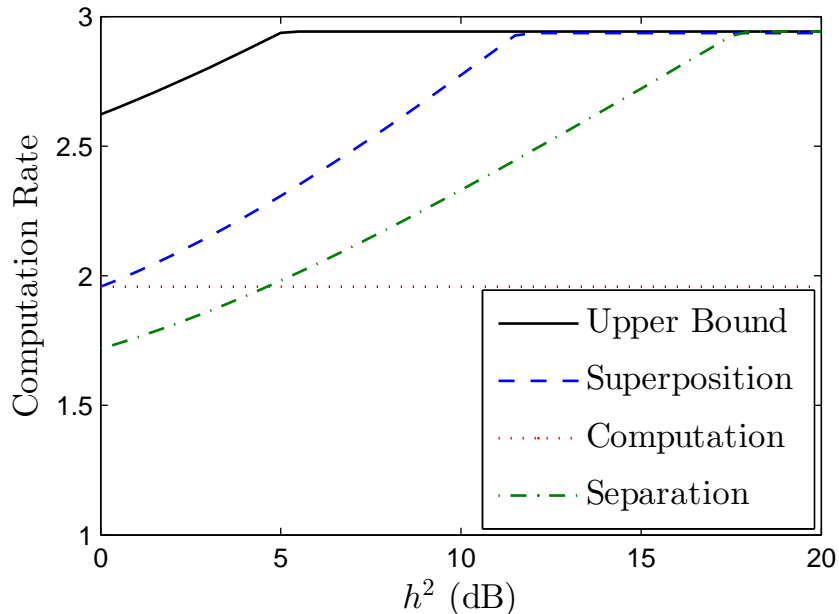


Fig. 11. Computation of  $S_1 + S_2$  for the two-user Gaussian MAC with unequal channel gains when  $P = 15$  dB.

### C. Fading Networks

Consider fading Gaussian MAC in which channel coefficients vary independently over time. The length- $n$  time-extended input-output is given by

$$\mathbf{y} = \sum_{i=1}^K \mathbf{H}_i \mathbf{x}_i + \mathbf{z}, \quad (76)$$

where  $\mathbf{H}_i = \text{diag}(h_i^{(1)}, \dots, h_i^{(n)})$  and  $h_i^{(t)}$  denotes the complex channel coefficient at time  $t$  from the  $i$ th sender to the receiver. We assume that  $\{h_i^{(t)}\}$  are independently drawn from  $\mathcal{CN}(0, 1)$  and also independent over time. We further assume that the elements of  $\mathbf{z}$  are independently drawn from  $\mathcal{CN}(0, 1)$ . Global channel state information is assumed to be available at each sender and the receiver. Since  $\{h_i^{(t)}\}$  are i.i.d. over time, we drop the time index hereafter for simplicity.

If we simply apply Theorem 5 for each time slot, then due to  $\min_{i \in [1:K]} |h_i|^2$  in Theorem 5, the achievable computation rate for fading will decrease as  $K$  increases.<sup>1</sup> We can fix this problem and achieve an approximate computation capacity similar to Corollary 1 and Remark 9 for fading Gaussian MAC.

To achieve this, we modify Theorem 5 to the fading scenario in a similar approach as given by Goldsmith and Varaiya [43]. At each time slot, the  $j$ th sender transmits with power  $\frac{\min_{i \in [1:K]} |h_i|^2 P}{|h_j|^2 \mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_j|^2]}$ , which satisfies the average power constraint. Then from the same analysis in Theorem 5, the computation rate

$$R = \frac{1}{H(f(\mathbf{S}))} \mathbb{E} \left[ \mathsf{C}^+ \left( \frac{1}{K} + \frac{\min_{i \in [1:K]} |h_i|^2 P}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]} \right) \right] := R_{\text{fading}}^{(l)} \quad (77)$$

is achievable, where  $\mathsf{C}^+(x) := \max\{\log(x), 0\}$ . On the other hand, the cut-set upper bound is given by

$$R \leq \frac{1}{H(f(\mathbf{S}))} \max_{\phi_1, \dots, \phi_K} \mathbb{E} \left[ \mathsf{C} \left( \left( \sum_{i=1}^K |h_i| \phi_i(h_1, \dots, h_K) \right)^2 \right) \right] := R_{\text{fading}}^{(u)}, \quad (78)$$

<sup>1</sup>Although we describe the paper including Theorem 5 based on the real channel model, the results in the paper can be straightforwardly extended to the complex channel model.

where  $\phi_i$  denotes the power allocation policy of the  $i$ th sender that should satisfy  $\mathbb{E}[\phi_i^2(h_1, \dots, h_K)] \leq P$  and  $C(x) := \log(1+x)$ . The following theorem establishes an approximate computation capacity for i.i.d. Rayleigh fading.

*Theorem 8 (Approximate Computation Capacity for Fading MAC):* Consider fading Gaussian MAC with time-varying channel coefficients. For i.i.d. Rayleigh fading,

$$R_{\text{fading}}^{(u)} - R_{\text{fading}}^{(l)} \leq \frac{3 \log K + 2 + \log e}{H(f(\mathbf{S}))} \quad (79)$$

for any  $P$ , where  $f(\mathbf{S}) = (U_1, \dots, U_L)$  for the arithmetic sum function and  $f(\mathbf{S}) = (B_0, \dots, B_{p-1})$  for the type function.

*Proof:* Let  $\{\phi_i^*\}$  denote an optimum power allocation policy maximizing (78). Then

$$\begin{aligned} & H(f(\mathbf{S})) (R_{\text{fading}}^{(u)} - R_{\text{fading}}^{(l)}) \\ & \stackrel{(a)}{\leq} \mathbb{E} \left[ \log \left( \frac{1 + \left( \sum_{i=1}^K |h_i| \phi_i^*(h_1, \dots, h_K) \right)^2}{1 + K \frac{\min_{i \in [1:K]} |h_i|^2}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]} P} \right) \right] + \log K \\ & \leq \mathbb{E} \left[ \log \left( \frac{K \frac{\min_{i \in [1:K]} |h_i|^2}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]} + \frac{1}{P} \left( \sum_{i=1}^K |h_i| \phi_i^*(h_1, \dots, h_K) \right)^2}{K \frac{\min_{i \in [1:K]} |h_i|^2}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]}} \right) \right] + \log K \\ & \stackrel{(b)}{\leq} 2 \mathbb{E} \left[ \log \left( \sqrt{K \frac{\min_{i \in [1:K]} |h_i|^2}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]}} + \frac{1}{\sqrt{P}} \left( \sum_{i=1}^K |h_i| \phi_i^*(h_1, \dots, h_K) \right) \right) \right] \\ & \quad + \log \left( \mathbb{E} \left[ \min_{i \in [1:K]} |h_i|^2 / |h_1|^2 \right] \right) - \mathbb{E} \left[ \log \left( \min_{i \in [1:K]} |h_i|^2 \right) \right] \\ & \stackrel{(c)}{\leq} 2 \log \left( \sqrt{K \frac{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2]}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]}} + \frac{1}{\sqrt{P}} \left( \sum_{i=1}^K \mathbb{E}[|h_i| \phi_i^*(h_1, \dots, h_K)] \right) \right) \\ & \quad + \log \left( \mathbb{E} \left[ \min_{i \in [1:K]} |h_i|^2 / |h_1|^2 \right] \right) - \mathbb{E} \left[ \log \left( \min_{i \in [1:K]} |h_i|^2 \right) \right] \\ & \stackrel{(d)}{\leq} 2 \log \left( \sqrt{K \frac{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2]}{\mathbb{E}[\min_{i \in [1:K]} |h_i|^2 / |h_1|^2]}} + K \right) + \log \left( \mathbb{E} \left[ \min_{i \in [1:K]} |h_i|^2 / |h_1|^2 \right] \right) - \mathbb{E} \left[ \log \left( \min_{i \in [1:K]} |h_i|^2 \right) \right] \\ & = 2 \log \left( \sqrt{K \mathbb{E} \left[ \min_{i \in [1:K]} |h_i|^2 \right]} + K \sqrt{\mathbb{E} \left[ \min_{i \in [1:K]} |h_i|^2 / |h_1|^2 \right]} \right) - \mathbb{E} \left[ \log \left( \min_{i \in [1:K]} |h_i|^2 \right) \right] \\ & \leq 2 \log(\sqrt{K} + K) - \mathbb{E} \left[ \log \left( \min_{i \in [1:K]} |h_i|^2 \right) \right]. \quad (80) \end{aligned}$$

where (a) follows since  $C^+ \left( \frac{1}{K} + P \right) \geq \frac{1}{2} \log(1 + KP) - \frac{1}{2} \log K$ , (b) follows since  $\log(a+b) \leq 2 \log(\sqrt{a} + \sqrt{b})$  for  $a \geq 0$  and  $b \geq 0$ , (c) follows from Jensen's inequality, and (d) follows since  $\mathbb{E}[|h_i| \phi_i^*(h_1, \dots, h_K)] \leq \sqrt{\mathbb{E}[|h_i|^2] \mathbb{E}[(\phi_i^*(h_1, \dots, h_K))^2]} \leq \sqrt{P}$ .

For Rayleigh fading,  $|h_i|^2$  is exponentially distributed. Thus, we have

$$\begin{aligned} \mathbb{E} \left[ \log \left( \min_{i \in [1:K]} |h_i|^2 \right) \right] &= \int_0^\infty \log(x) K \exp(-Kx) dx \\ &= -\log K + \int_0^\infty \log(u) \exp(-u) du \\ &\geq -\log K + \int_0^1 \log(u) du + \int_1^\infty \log(u) \exp(-u) du \\ &\geq -\log K - \log e. \quad (81) \end{aligned}$$



Therefore, from (80) and (81),

$$R_{\text{fading}}^{(u)} - R_{\text{fading}}^{(l)} \leq \frac{3 \log K + 2 + \log e}{H(f(\mathbf{S}))}, \quad (82)$$

which completes the proof.  $\blacksquare$

#### D. Scaling Laws

One interesting performance metric is to focus on how the computation rate scales as the number of sources  $K$  increases. The work [11] studied scaling laws on the computation rate under collocated collision networks assuming that concurrent transmission from multiple nodes causes collisions and, therefore, is not allowed. It was shown in [11] that the order of  $\frac{1}{K}$  rate scaling law is achievable for the type computation, which is the same scaling law achievable by the separation-based computation. Whereas, Theorem 5 provides the order of  $\frac{1}{\log K}$  rate scaling law for collocated Gaussian networks. The gain comes from a more efficient physical layer abstraction using the compute-and-forward framework and exploiting the superposition property of the abstracted channel for function computation, which is not allowed for collocated collision networks.

#### E. Function Multicast

Throughout the paper, we assume that a single receiver wishes to compute the desired function. Now consider the function multicasting problem in which multiple receivers wish to compute the same desired function. It has been shown in [9] that function alignment is essentially required for linear finite field single-hop networks in order to optimally compute the modulo sum function at multiple receivers. For the multihop case, if there is a relay node that is connected from all senders and also at the same time connected to all receivers, this relay node can first compute the desired function and then broadcast it to multiple receivers. For the first task, i.e., function computation at a single relay node, our coding schemes are applicable. For the second task, i.e., function broadcast to multiple receivers, quantize-map-and-forward [32] or noisy network coding [44] achieves a near-optimal rate. Although this approach is not applicable for any network topology, we can easily find examples of interest that it achieves a near-optimal computation rate. A similar approach has been also proposed in [33], analyzing the rate scaling law under a bit-pipe wired network represented by an undirected graph.

### VIII. CONCLUSION

In this paper, we studied the function computation over Gaussian networks assuming orthogonal components. We proposed a novel computation coding that is able to compute multiple weighted arithmetic sums including the type function. Computing the type function is very powerful since any symmetric function such as the same mean, maximum, minimum, and so on, can be obtained from the type function. Hence, the proposed computation coding is useful not only for the arithmetic sum computation, but for any symmetric function computation. The main ingredients of the proposed computation coding are the network transformation via lattice codes and linear network coding and then the computation based on linear Slepian–Wolf source coding for computing. In many cases, the proposed computation coding outperforms the separation-based computation, especially when the number of sources becomes large. We established the computation capacity for a class of orthogonal Gaussian networks and an approximate capacity for a class of Gaussian networks with multiple-access.

#### APPENDIX I HYBRID APPROACH

In this appendix, we prove Theorem 3. First consider a distributed source coding problem with rate tuple  $(R_1, \dots, R_K)$  to compute the desired function. The  $i$ th bit-pipe orthogonal channel with rate  $R_i$  can be treated as the orthogonal finite field channel with  $\mathbb{F}_q^{kR_i(\log q)^{-1}}$ . From the same argument in Lemma 2, setting  $q > (p-1)^2K$  and computing the corresponding modulo- $q$  sum function yields the desired function. We generalize the coding scheme in [5, Section VI] to the  $K$ -user case, (Also see [25, Theorem III.2] for the rate constraints for decoding the set of auxiliary random sequences). Then any rate tuple  $(R_1, \dots, R_K)$  satisfying

$$\sum_{i \in \Sigma} R_i \geq I(\{W_i\}_{i \in \Sigma}; \mathbf{S} | \{W_i\}_{i \in [1:K] \setminus \Omega}) + |\Sigma| H(f(\mathbf{S}) | W_1, \dots, W_K) \quad (83)$$

for all  $\Sigma \subseteq [1 : K]$  is a necessary condition for the desired function computation. By abstracting the  $i$ th orthogonal Gaussian channel using point-to-point capacity-achieving codes, we have error-free bit-pipe channel with rate  $C(h_i^2)$ . Hence if a rate tuple  $(\frac{C(h_1^2)}{R}, \dots, \frac{C(h_K^2)}{R})$  is located inside the region (83), the computation rate  $R$  is achievable, which provides the rate constraint on  $R$  as in (34). In conclusion, Theorem 3 holds.

## APPENDIX II NETWORK TRANSFORMATION

In this appendix, we prove Lemma 3.

### A. Layered Networks

In this subsection, we prove Lemma 3 for the layered case. For layered networks, we can partition the set of nodes into  $M$  layers. Let  $V[j] \subseteq V$  denote the set of nodes at the  $j$ th layer, where  $j \in [1 : M]$ . We assume that  $V[1]$  is the set of senders and the node at the  $M$ th layer is the receiver. That is,  $V[1] = \{t_i\}_{i=1}^K$  and  $V[M] = \{d\}$ . The encoding functions are set as follows:

- (Sender Encoding) Node  $v \in V[1]$  transmits  $\mathbf{x}_v = \mathbf{F}_v \mathbf{x}_v''$ , where  $\mathbf{F}_v \in \mathbb{F}_q^{n\alpha_v \times n\tau}$  and  $\mathbf{x}_v'' \in \mathbb{F}_q^{n\tau}$ . We will specify  $\tau$  later.
- (Relay Encoding) Node  $v \in \bigcup_{j=2}^{M-1} V[j]$  transmits  $\mathbf{x}_v = \mathbf{F}_v \mathbf{y}_v$ , where  $\mathbf{F}_v \in \mathbb{F}_q^{n\alpha_v \times n\beta_v}$ .

Then the receiver generates  $\mathbf{y}'_d = \mathbf{F}_d \mathbf{y}_d$ , where  $\mathbf{F}_d \in \mathbb{F}_q^{n\tau \times n\beta_d}$ .

Let  $\Gamma_{in}^u(v) = \{w : \text{there exists a direct path from } u \text{ to } w, w \in \Gamma_{in}(v)\}$ . From the definition,  $\Gamma_{in}^u(v) \subseteq \Gamma_{in}(v)$ . Then the input–output from  $\mathbf{x}'_u$ ,  $u \in V[1]$ , to  $\mathbf{y}'_d$  assuming  $\mathbf{x}'_v = \mathbf{0}$  for all  $v \neq u$  is given by

$$\begin{aligned}
& \mathbf{y}'_d |_{\mathbf{x}'_v = \mathbf{0}, \forall v \neq u} \\
&= \mathbf{F}_d \bigoplus_{v_{M-1} \in \Gamma_{in}^u(d)} \mathbf{H}_{v_{M-1}, d} \mathbf{x}_{v_{M-1}} \\
&= \mathbf{F}_d \bigoplus_{v_{M-1} \in \Gamma_{in}^u(d)} \mathbf{H}_{v_{M-1}, d} \mathbf{F}_{v_{M-1}} \bigoplus_{v_{M-2} \in \Gamma_{in}^u(v_{M-1})} \mathbf{H}_{v_{M-2}, v_{M-1}} \mathbf{F}_{v_{M-2}} \cdots \bigoplus_{v_2 \in \Gamma_{in}^u(v_3)} \mathbf{H}_{v_2, v_3} \mathbf{x}_{v_2} \\
&= \mathbf{F}_d \bigoplus_{v_{M-1} \in \Gamma_{in}^u(d)} \mathbf{H}_{v_{M-1}, d} \mathbf{F}_{v_{M-1}} \bigoplus_{v_{M-2} \in \Gamma_{in}^u(v_{M-1})} \mathbf{H}_{v_{M-2}, v_{M-1}} \mathbf{F}_{v_{M-2}} \cdots \bigoplus_{v_2 \in \Gamma_{in}^u(v_3)} \mathbf{H}_{v_2, v_3} \mathbf{F}_{v_2} \mathbf{H}_{u, v_2} \mathbf{F}_u \mathbf{x}_u''. \tag{84}
\end{aligned}$$

Let us denote

$$\mathbf{H}_u := \mathbf{F}_d \bigoplus_{v_{M-1} \in \Gamma_{in}^u(d)} \cdots \bigoplus_{v_2 \in \Gamma_{in}^u(v_3)} \mathbf{H}_{v_{M-1}, d} \mathbf{F}_{v_{M-1}} \cdots \mathbf{H}_{v_2, v_3} \mathbf{F}_{v_2} \mathbf{H}_{u, v_2} \mathbf{F}_u, \tag{85}$$

which is the  $n\tau \times n\tau$  dimensional end-to-end channel matrix from  $\mathbf{x}_u''$  to  $\mathbf{y}'_d$ . Then  $\mathbf{y}'_d$  can be represented as

$$\begin{aligned}
\mathbf{y}'_d &= \bigoplus_{u \in V[1]} \mathbf{H}_u \mathbf{x}_u'' \\
&= \bigoplus_{i=1}^K \mathbf{H}_{t_i} \mathbf{x}_{t_i}''. \tag{86}
\end{aligned}$$

The following theorem and corollary show that if the size of the end-to-end channel matrix  $\mathbf{H}_{t_i}$  is smaller than the corresponding minimum-cut value, then  $\mathbf{H}_{t_i}$  becomes a full-rank matrix for all  $i \in [1 : K]$  with probability approaching one as  $n$  increases.

*Theorem 9 (Avestimehr–Diggavi–Tse [32]):* For any  $i \in [1 : K]$ , let  $\{\mathbf{x}_{t_i}''(w)\}_{w \in [1:2^{n\tau \log q}]}$  be a set of randomly chosen  $2^{n\tau \log q}$  vectors in  $\mathbb{F}_q^n$  and  $\{\mathbf{y}'_d(w)\}_{w \in [1:2^{n\tau \log q}]}$  be the corresponding set of output vectors, i.e.,  $\mathbf{y}'_d(w) = \mathbf{H}_{t_i} \mathbf{x}_{t_i}''(w)$ . Suppose that the elements of  $\mathbf{F}_v$  are i.i.d. drawn uniformly from  $\mathbb{F}_q$  for all  $v \in V$ . Then there exists one-to-one correspondence between  $\{\mathbf{x}_{t_i}''(w)\}_{w \in [1:2^{n\tau \log q}]}$  and  $\{\mathbf{y}'_d(w)\}_{w \in [1:2^{n\tau \log q}]}$  with probability approaching one as  $n$  increases, provided that

$$\tau \leq \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})). \tag{87}$$

*Proof:* We refer to [32, Theorem 4.1] for the proof. ■

*Corollary 2:* Suppose that the elements of  $\mathbf{F}_v$  are chosen i.i.d. uniformly from  $\mathbb{F}_q$  for all  $v \in V$ . If  $\tau \leq \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\}))$ , then

$$\text{rank}(\mathbf{H}_{t_i}) = n\tau \quad (88)$$

for all  $i \in [1 : K]$  with probability approaching one as  $n$  increases.

*Proof:* Assume that  $\text{rank}(\mathbf{H}_{t_i}) < n\tau$ . Then, since the vector space spanned by  $\mathbf{H}_{t_i}$  is  $\mathbb{F}_q^{\text{rank}(\mathbf{H}_{t_i})}$  space, it contains strictly less than  $q^{n\tau}$  distinguishable vectors in  $\mathbb{F}_q^{n\tau}$ . On the other hand, Theorem 9 shows that it is possible to have  $2^{n\tau \log q} = q^{n\tau}$  distinguishable  $\mathbf{y}'_d$ 's with probability approaching one as  $n$  increases, which contradicts the assumption. In conclusion, from the union bound, (88) holds for all  $i \in [1 : K]$  with probability approaching one as  $n$  increases, which completes the proof. ■

Based on Corollary 2, we set such that the elements of  $\mathbf{F}_v$  are independently and uniformly chosen from  $\mathbb{F}_q$  for all  $v \in V$  and

$$\tau = \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})), \quad (89)$$

which guarantees the existence of  $\mathbf{H}_{t_i}^{-1}$  for all  $i \in [1 : K]$  with probability approaching one as  $n$  increases. Then by setting  $\mathbf{x}''_{t_i} = \mathbf{H}_{t_i}^{-1} \mathbf{x}'_{t_i}$ ,  $\mathbf{x}'_{t_i} \in \mathbb{F}_q^{n\tau}$ , for  $n$  sufficiently large, we have

$$\mathbf{y}'_d = \bigoplus_{i=1}^K \mathbf{x}'_{t_i} \quad (90)$$

from (86). In conclusion, Lemma 3 holds for layered networks.

### B. Arbitrary Networks

In this subsection, we prove Lemma 3 for a general linear finite field network (not necessarily layered). We can unfold the network  $G$  over time to establish the corresponding layered network. The underlying approach is similar to that proposed in [32, Section V. B]. Let

$$\bar{C}_{\min} := \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_\Omega(\{i\})) \log q \quad (91)$$

Define the  $T$  time-steps unfolded network  $G_{\text{TU}} = (V_{\text{TU}}, E_{\text{TU}})$  as follows.

- The network has  $T + 2$  stages, numbered from 0 to  $T + 1$ .
- Stage 0 has the senders  $t_1[0]$  to  $t_K[0]$ , which are the senders, and stage  $T + 1$  has node  $d[T + 1]$ , which is the receiver.
- Stage  $j$  has all nodes  $v \in V$  denoted by  $v[j]$ , where  $j \in [1 : T]$ . These nodes will act as relay nodes.
- There are finite-capacity links with the rate of  $T\bar{C}_{\min}$  between
  - $(t_i[0], t_i[1])$  for all  $i \in [1 : K]$  and  $(d[T], d[T + 1])$ .
  - $(v[j], v[j + 1])$  for all  $v \in V$  and  $j \in [1 : T]$ .
- Node  $v[j]$  is connected to node  $w[j + 1]$  with the linear finite field channel of the original network  $G$  for all  $(v, w) \in E, v \neq w$ .

The length- $n$  time-extended transmit signal of  $v[j], j \in [0 : T]$ , is given by the pair of  $(\mathbf{x}_{v[j]}^{(1)}, \mathbf{x}_{v[j]}^{(2)})$ , where  $\mathbf{x}_{v[0]}^{(1)} \in \mathbb{F}_q^{nT\bar{C}_{\min}/\log q}$  and  $\mathbf{x}_{v[0]}^{(2)} \in \mathbb{F}_q^{n\alpha_{v[0]}}$ . The length- $n$  time-extended received signal of  $v[1] \in \Gamma_{\text{out}}(t_i[0]), i \in [1 : K]$ , is given by  $\mathbf{y}_{v[1]}^{(1)} = \mathbf{x}_{v[0]}^{(1)}$ . The length- $n$  time-extended received signal of  $v[j], j \in [2 : T]$ , is given by  $(\mathbf{y}_{v[j]}^{(1)}, \mathbf{y}_{v[j]}^{(2)})$ , where  $\mathbf{y}_{v[j]}^{(1)} = \mathbf{x}_{v[j-1]}^{(1)}$  and  $\mathbf{y}_{v[j]}^{(2)} = \bigoplus_{v[j-1] \in \Gamma(v[j])} \mathbf{H}_{v[j-1], v[j]} \mathbf{x}_{v[j-1]}^{(2)}$  (see the input–output relation (48)). The length- $n$  time-extended received signal of  $v[T + 1]$  is given by  $\mathbf{y}_{v[T+1]}^{(1)} = \mathbf{x}_{v[T]}^{(1)}$ . For other unspecified received signals, they receive all-zero vectors. For a better understanding, Fig. 12 illustrates an example of an  $T$  time-steps unfolded network.

Since this unfolded network  $G_{\text{TU}}$  is layered, we can apply the same linear coding as in Appendix II. A to  $G_{\text{TU}}$ . Specifically,

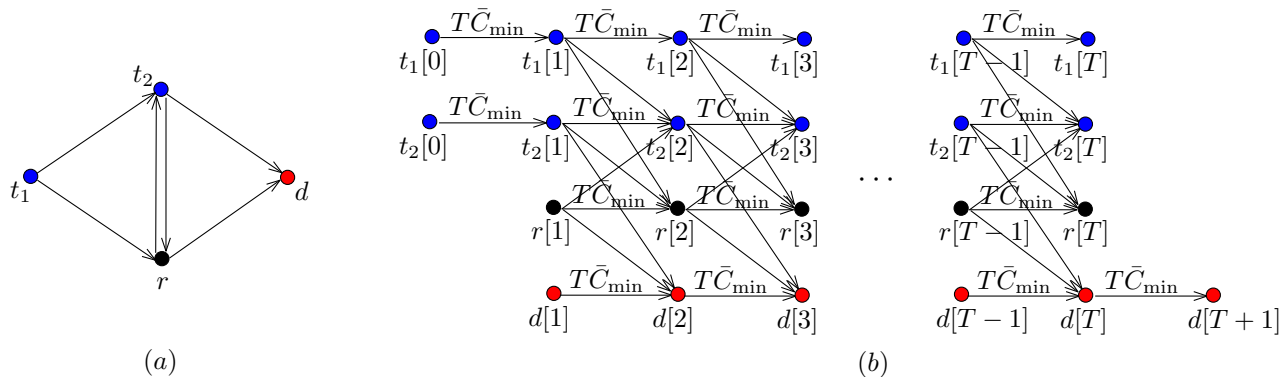


Fig. 12. Unfolded network example, where (a) is the original network and (b) is the corresponding unfolded network.

- (Sender Encoding) Node  $v[0]$  transmits  $\mathbf{x}_{v[0]}^{(1)} = \mathbf{F}_{v[0]}^{(1)} \mathbf{x}_{v[0]}''$  and  $\mathbf{x}_{v[0]}^{(2)} = \mathbf{F}_{v[0]}^{(2)} \mathbf{x}_{v[0]}''$ , where  $\mathbf{F}_{v[0]}^{(1)} \in \mathbb{F}_q^{nT\bar{C}_{\min}/\log q \times n\tau}$ ,  $\mathbf{F}_{v[0]}^{(2)} \in \mathbb{F}_q^{n\alpha_{v[0]} \times n\tau}$ , and  $\mathbf{x}_{v[0]}'' \in \mathbb{F}_q^{n\tau}$ . We will specify  $\tau$  later.
- (Relay Encoding) Node  $v[j]$ ,  $j \in [1 : T]$ , transmits  $\mathbf{x}_{v[j]}^{(1)} = \mathbf{F}_{v[j]}^{(1)} \left[ \mathbf{y}_{v[j]}^{(1)T}, \mathbf{y}_{v[j]}^{(2)T} \right]^T$  and  $\mathbf{x}_{v[j]}^{(2)} = \mathbf{F}_{v[j]}^{(2)} \left[ \mathbf{y}_{v[j]}^{(1)T}, \mathbf{y}_{v[j]}^{(2)T} \right]^T$ , where  $\mathbf{F}_{v[j]}^{(1)} \in \mathbb{F}_q^{nT\bar{C}_{\min}/\log q \times n(T\bar{C}_{\min}/\log q + \beta_{v[j]})}$  and  $\mathbf{F}_{v[j]}^{(2)} \in \mathbb{F}_q^{n\alpha_{v[j]} \times n(T\bar{C}_{\min}/\log q + \beta_{v[j]})}$ .

Then the receiver generates  $\mathbf{y}'_{v[T+1]} = \mathbf{F}_{v[T+1]}^{(1)} \mathbf{y}_{v[T+1]}^{(1)}$ , where  $\mathbf{F}_{v[T+1]}^{(1)} \in \mathbb{F}_q^{n\tau \times nT\bar{C}_{\min}/\log q}$ .

Similar to (86), the input–output from  $\{\mathbf{x}_{v[0]}''\}_{v[0] \in V[0]}$  to  $\mathbf{y}'_{v[T+1]}$  can be represented as

$$\mathbf{y}'_{v[T+1]} = \bigoplus_{v[0] \in V[0]} \mathbf{H}_{v[0]} \mathbf{x}_{v[0]}'', \quad (92)$$

where  $V[0]$  denotes the set of nodes at stage 0, which is the set of senders, and  $v[T+1]$  is the node at stage  $T+1$ , which is the receiver. Here  $\mathbf{H}_{v[0]}$  is the end-to-end channel matrix from  $\mathbf{x}_{v[0]}''$  to  $\mathbf{y}'_{v[T+1]}$ .

Now consider the minimum-cut value of  $G_{\text{TU}}$  with respect to a subset of nodes in  $V[0]$ . In the same manner in Section III-D, for  $\Sigma \subseteq V[0]$ , we can define  $G_{\text{TU}}(\Sigma)$  and  $\Lambda_{\text{TU}}(\Sigma)$ . Then the minimum-cut value is given by

$$\bar{C}_{\text{TU}}(\Sigma) := \min_{\Omega \in \Lambda_{\text{TU}}(\Sigma)} \text{rank} \left( H_{\Omega}^{\text{TU}}(\Sigma) \right) \log q, \quad (93)$$

where  $H_{\Omega}^{\text{TU}}(\Sigma)$  denotes the transfer matrix associated with the cut  $\Omega \in \Lambda_{\text{TU}}(\Sigma)$  on  $G_{\text{TU}}(\Sigma)$ .

Hence, from Theorem 9 and Corollary 2, by setting the elements of  $\mathbf{F}_{v[j]}^{(1)}$  and  $\mathbf{F}_{v[j]}^{(2)}$  i.i.d. drawn uniformly from  $\mathbb{F}_q$ , we can guarantee that

$$\text{rank}(\mathbf{H}_{v[0]}) = n\tau \quad (94)$$

for all  $v[0] \in V[0]$  with probability approaching one as  $n$  increases if

$$\tau \leq \min_{v[0] \in V[0]} \frac{\bar{C}_{\text{TU}}(\{v[0]\})}{\log q}. \quad (95)$$

For  $v[0] \in V[0]$ , the minimum-cut value is lower bounded by

$$\begin{aligned} \bar{C}_{\text{TU}}(\{v[0]\}) &= \min_{\Omega \in \Lambda_{\text{TU}}(\{v[0]\})} \text{rank} \left( H_{\Omega}^{\text{TU}}(\{v[0]\}) \right) \log q \\ &\geq (T - |V|) \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_{\Omega}(\{i\})) \log q, \end{aligned} \quad (96)$$

where  $t_i = v[0]$ . The inequality follows from the same analysis in [32, Lemma 5.2].

From (95) and (96), we set

$$\tau = (T - |V|) \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_{\Omega}(\{i\})) \quad (97)$$

and the elements of  $\mathbf{F}_{v[j]}^{(1)}$  and  $\mathbf{F}_{v[j]}^{(2)}$  i.i.d. drawn uniformly from  $\mathbb{F}_q$ . This guarantees the existence of  $\mathbf{H}_{v[0]}^{-1}$  for all  $v[0] \in V[0]$  with probability approaching one as  $n$  increases. Therefore from (92), setting  $\mathbf{x}''_{v[0]} = \mathbf{H}_{v[0]}^{-1} \mathbf{x}'_{v[0]}$ ,  $\mathbf{x}'_{v[0]} \in \mathbb{F}_q^{n\tau}$ , provides

$$\mathbf{y}_{v[T+1]}^{(1)} = \bigoplus_{v[0] \in V[0]} \mathbf{x}'_{v[0]}. \quad (98)$$

Finally, since any coding scheme for the  $T$  time-steps unfolded network  $G_{\text{TU}}$  can be performed in the original network  $G$  using  $nT$  time slots, see the argument in [32, Lemma 5.1], we have

$$\mathbf{y}'_d = \bigoplus_{i=1}^K \mathbf{x}'_{t_i}, \quad (99)$$

where  $\mathbf{x}'_{t_i} \in \mathbb{F}_q^{n\tau}$  for all  $i \in [1 : K]$  using  $nT$  time slots for the original network  $G$ . Here, we simply rewrite  $\{\mathbf{x}'_{t_i}\}_{i \in [1:K]} = \{\mathbf{x}'_{v[0]}\}_{v[0] \in V[0]}$  and  $\mathbf{y}'_d = \mathbf{y}_{v[T+1]}^{(1)}$  from (98) since  $V[0]$  is the set of senders and  $v[T+1]$  is the receiver. Then using  $n$  time slots, we have (99) with  $\mathbf{x}'_{t_i} \in \mathbb{F}_q^{n\tau/T}$ . From the fact that  $\lim_{T \rightarrow \infty} \frac{\tau}{T} \rightarrow \min_{i \in [1:K]} \min_{\Omega \in \Lambda(\{i\})} \text{rank}(H_{\Omega}(\{i\}))$ , Lemma 3 holds for any arbitrary networks.

## REFERENCES

- [1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
- [2] T. M. Cover, A. El Gamal, and M. Salehi, "Multiple access channels with arbitrarily correlated sources," *IEEE Trans. Inf. Theory*, vol. IT-26, pp. 648–657, Nov. 1980.
- [3] M. Gastpar, "Uncoded transmission is exactly optimal for a simple Gaussian "sensor" network," *IEEE Trans. Inf. Theory*, vol. 54, pp. 5247–5251, Nov. 2008.
- [4] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 219–221, Mar. 1979.
- [5] R. Ahlswede and T. S. 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. IT-29, pp. 396–412, May 1983.
- [6] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, pp. 3498–3516, Oct. 2007.
- [7] —, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, pp. 6463–6486, Oct. 2011.
- [8] J. Zhan, S. Y. Park, M. Gastpar, and A. Sahai, "Linear function computation in networks: Duality and constant gap results," *IEEE J. Select. Areas Commun.*, vol. 31, pp. 620–638, Apr. 2013.
- [9] C. Suh, N. Goela, and M. Gastpar, "Computation in multicast networks: Function alignment and converse theorems," in *Proc. 50th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2012.
- [10] R. Soundararajan and S. Vishwanath, "Communicating linear functions of correlated Gaussian sources over a MAC," *IEEE Trans. Inf. Theory*, vol. 58, pp. 1853–1860, Mar. 2012.
- [11] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 755–764, Apr. 2005.
- [12] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, Jul. 2000.
- [13] S.-Y. R. Li, R. W. Yueng, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, pp. 371–381, Feb. 2003.
- [14] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE Trans. Inf. Theory*, vol. 11, pp. 782–795, Oct. 2003.
- [15] W. Nam, S.-Y. Chung, and Y. H. Lee, "Nested lattice codes for Gaussian relay networks with interference," *IEEE Trans. Inf. Theory*, vol. 57, pp. 7733–7745, Dec. 2011.
- [16] A. Lapidath and S. Tinguely, "Sending a bivariate Gaussian over a Gaussian MAC," *IEEE Trans. Inf. Theory*, vol. 56, pp. 2714–2752, Jun. 2010.
- [17] S. Lim, P. Minero, and Y.-H. Kim, "Lossy communication of correlated sources over multiple access channels," in *Proc. 48th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Sep./Oct/ 2010.
- [18] S. I. Bross, A. Lapidath, and S. Tinguely, "Broadcasting correlated Gaussians," *IEEE Trans. Inf. Theory*, vol. 56, pp. 3057–3068, Jul. 2010.
- [19] C. Tian, S. Diggavi, and S. Shamai, "The achievable distortion region of sending a bivariate Gaussian source on the Gaussian broadcast channel," *IEEE Trans. Inf. Theory*, vol. 57, pp. 6419–6427, Oct. 2011.
- [20] M. Gastpar and M. Vetterli, "On the capacity of large gaussian relay networks," *IEEE Trans. Inf. Theory*, vol. 51, pp. 765–779, Mar. 2005.
- [21] T. S. Han and K. Kobayashi, "A dichotomy of functions  $F(X, Y)$  of correlated sources  $(X, Y)$  from the viewpoint of the achievable rate region," *IEEE Trans. Inf. Theory*, vol. IT-33, pp. 69–76, Jan. 1987.
- [22] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, pp. 903–917, Mar. 2001.
- [23] P. Cuff, H.-I. Su, and A. El Gamal, "Cascade multiterminal source coding," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seoul, South Korea, Jun./Jul. 2009.
- [24] N. Ma and P. Ishwar, "Some results on distributed source coding for interactive function computation," *IEEE Trans. Inf. Theory*, vol. 57, pp. 6180–6195, Sep. 2011.

- [25] S. Huang and M. Skoglund, "Polynomials and computing functions of correlated sources," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA, Jul. 2012.
- [26] —, "On linear coding over finite rings and applications to computing," Submitted to *IEEE Trans. Inf. Theory*, Oct. 2012.
- [27] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inf. Theory*, vol. IT-28, pp. 585–592, Jul. 1982.
- [28] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Network coding for computing: Cut-set bounds," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1015–1030, Feb. 2011.
- [29] R. Appuswamy and M. Franceschetti, "Computing linear functions by linear coding over networks," Submitted to *IEEE Trans. Inf. Theory*, Feb. 2011.
- [30] H. Kowshik and P. R. Kumar, "Optimal function computation in directed and undirected graphs," *IEEE Trans. Inf. Theory*, vol. 59, pp. 3407–3418, Jun. 2012.
- [31] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4413–4430, Oct. 2006.
- [32] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1872–1905, Apr. 2011.
- [33] S. Kannan and P. Viswanath, "Multi-terminal function multicasting in undirected graphs," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.
- [34] S. Kannan and P. Vishwanath, "Multi-session function computation and multicasting in undirected graph," *IEEE J. Select. Areas Commun.*, vol. 31, pp. 702–713, Apr. 2013.
- [35] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Linear codes, target function classes, and network computing capacity," Submitted to *IEEE Trans. Inf. Theory*, May 2011.
- [36] N. Ma, P. Ishwar, and P. Gupta, "Interactive source coding for function computation in collocated networks," *IEEE Trans. Inf. Theory*, vol. 58, pp. 4289–4305, Jul. 2012.
- [37] C.-Y. Wang, S.-W. Jeon, and M. Gastpar, "Multi-round computation of type-threshold functions in collocated Gaussian networks," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.
- [38] U. Erez and R. Zamir, "Achieving  $\frac{1}{2} \log(1 + \text{SNR})$  on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2293–2314, Oct. 2004.
- [39] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Trans. Inf. Theory*, vol. 51, pp. 3401–3416, Oct. 2005.
- [40] W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity of the Gaussian two-way relay channel to within  $\frac{1}{2}$  bit," *IEEE Trans. Inf. Theory*, vol. 56, pp. 5488–5494, Nov. 2010.
- [41] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 471–480, Jul. 1973.
- [42] S.-C. Chang and E. Weldon, "Coding for  $t$ -user multiple-access channels," *IEEE Trans. Inf. Theory*, vol. IT-25, pp. 684–691, Nov. 1979.
- [43] A. Goldsmith and P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, pp. 1986–1992, Nov. 1997.
- [44] S. H. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Trans. Inf. Theory*, vol. 57, pp. 3132 – 3152, May 2011.