ChannelComp: A General Method for Computation by Communications

Saeed Razavikia[†], José Mairton Barros Da Silva Júnior*, Carlo Fischione[†]

[†]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

*Department of Information Technology, Uppsala University, Sweden

Email: {sraz, carlofi}@kth.se, mairton.barros@it.uu.se

Abstract—Over-the-air computation (AirComp) is a well-known technique by which several wireless devices transmit by analog amplitude modulation to achieve a sum of their transmit signals at a common receiver. The underlying physical principle is the superposition property of the radio waves. Since such superposition is analog and in amplitude, it is natural that AirComp uses analog amplitude modulations. Unfortunately, this is impractical because most wireless devices today use digital modulations. It would be highly desirable to use digital communications because of their numerous benefits, such as error correction, synchronization, acquisition of channel state information, and widespread use. However, when we use digital modulations for AirComp, a general belief is that the superposition property of the radio waves returns a meaningless overlapping of the digital signals. In this paper, we break through such beliefs and propose an entirely new digital channel computing method named ChannelComp, which can use digital as well as analog modulations. We propose a feasibility optimization problem that ascertains the optimal modulation for computing arbitrary functions over-the-air. Additionally, we propose pre-coders to adapt existing digital modulation schemes for computing the function over the multiple access channel. The simulation results verify the superior performance of ChannelComp compared to AirComp, particularly for the product functions, with more than 10 **dB** improvement of the computation error.

I. INTRODUCTION

To realize the ubiquitous connectivity from the Internet of Things (IoT), the generations of wireless communications have been accompanied by a paradigm shift from human-type communications towards machine-type communications. On the one hand, the number of IoT devices is predicted to reach 75 billion by 2025 [2], much higher than that of mobile phone users. On the other hand, the various IoT

TABLE I: Reference list of commonly used variables in this survey. Ordered by case and alphabetically.

Variable	Definition
$\mathcal{D}_f$	Domain set of function f
$\mathcal{R}_f$	Range set of function f
$\mathcal{R}_s$	Set of all possible constellation points
$\mathscr{T}(\cdot)$	Tabular mapping to compute function f
$\mathscr{E}_k(\cdot)$	Encoder at node k
f	Desired function
$f^{(i)}$	Value of output i of desired function f
h_k	Channel coefficient between node k and the center point
K	Number of nodes in the network
q	Number of quantization level
$\vec{s}_i$	The induced value over-the-air corresponding to $f^{(i)}$
$\vec{x}_k$	Modulated signal of node k
$\vec{y}$	The received value at the center point
$\vec{z}$	Additive white Gaussian noise

applications based on machine learning (ML) are set to emerge in 6G [3], and these require the collection, transmission, and calculation of enormous amounts of data from many devices. Consequently, extensive connectivity needs to scale up radio and computation resources, which means swamping the capacity of the current systems. To improve the support of emerging compute-intensive ML applications, e.g., virtual reality, edge computing, federated edge learning, the over-the-air computation (AirComp) method is a promising concept to simultaneously collect and compute data at the edge network [4]–[7].

AirComp leverages the waveform superposition property of the multi-access channel (MAC) to realize the aggregation of data simultaneously transmitted by devices, allowing each device to access all radio resources, unlike the standard transmit-then-compute scheme. Moreover, AirComp integrates communication and computation steps, providing ultra-fast wireless data aggregation in IoT networks with high-spectrum efficiency. AirComp reduces the required energy of each device for transmission while it can bring a high rate of communication of that device by harnessing interference to help functional computation. Besides preserving the privacy and security of data, the coverage area can also be enlarged since more devices can transmit simultaneously.

A. Literature review

Two different AirComp approaches exist in the literature: the uncoded analog aggregation [6], [11] and coded digital AirComp [12], [13]. The coded approaches use the linearity

TABLE II: Digital computation over-the-air methods for a star network with K nodes.

Methods	Bandwidth/Computation	Modulation scheme	Function
ChannelComp (proposed)	1	Any digital modulation	Any
AirComp [5]	1	Analog amplitude modulation	Nomographic
OFDMA	K	Any digital modulation	Any
OBDA [8]	1	BPSK and QPSK	Sign
OBDA-FSK [9]	2	FSK	Sign
OAC-Balanced [10]	$\approx \log q$	PAM	Mean

of nested lattice codes to encode the input data for obtaining computation over Gaussian channels [14], [15]. The analog AirComp has been widely studied and progressed from different points of view, such as information theory [6], signal processing [5], transceiver design [16], channel state information acquisition [17], or synchronization issues [18]. Compared with standard schemes, analog technique AirComp can dramatically reduce the required communication resources, particularly in distributed learning, where it has also attracted growing attention for federated edge learning systems [19], [20].

Although AirComp is a promising concept for data aggregation in communication systems (e.g., federated edge learning), it entirely depends on analog communication, which is difficult for reliable communications due to channel ramifications [10]. Furthermore, AirComp needs analog hardware systems for utilizing analog modulations, which is a drawback due to the limited number of current wireless devices that support analog modulations. Accordingly, the use of digital modulation is deemed more advantageous due to its outstanding properties in channel correction, source, and channel coding, and widespread adoption. This, however, is believed to be extremely difficult due to that the overlapping of digitally modulated signals returns, in general, meaningless or incomprehensible signals for function computation [4], [21], [22].

Recently, there have been some attempts to devise digital aggregation methods, e.g., one-bit broadband digital aggregation (OBDA) [8] and the other-based majority vote frequency-shift keying (FSK) [9]. Moreover, in [23], a phase asynchronous OFDM-based version of OBDA has been proposed by designing joint channel decoding and aggregation decoders tailored for digital AirComp. Further, AirComp's non-coherent communication solution for single and multi-cell using pulse-position modulation and FSK have been studied in [9], [24], [25]. All the aforementioned OBDA studies are limited to specific functions (sign or summation function) or specific ML training procedures (signSGD problem [26] for computing majority vote). Note that [10] has proposed to utilize the balanced number systems to compute the summation function while incurring higher bandwidth usage than the standard OBDA method due to allocating frequency for transmitting every quantized level. Similarly, another encoding-based numeral system is proposed in [27], where the decimal representation of the input bits is mapped to a pulse amplitude modulation (PAM) symbol to achieve a processing gain. The most representative methods of the AirComp literature are summarised in Table II, including their bandwidth usage, computation, and which functions can be computed.

Consequently, unlike function computations in the AirComp method, existing attempts to use digital modulations with AirComp cannot compute larger classes of functions beyond Nomographic functions [15], [28] and are unsuitable for general digital modulations beyond the simple BPSK or FSK. Indeed, the existing digital aggregation methods enforce analog AirComp over digital communication. Such enforcement works in very few, and particular cases of digital modulations where it appears highly inefficient regarding communication resource usage, such as satellite communications [29], unmanned aerial vehicles [30], distributed consensus [31].

In this paper, we propose a new channel computation method termed *ChannelComp* that is fully compatible with existing digital communication systems, such as those currently available on any smartphone or IoT system. We investigate the broad set of functions that ChannelComp can compute, and we investigate how ChannelComp makes it possible for digital modulation schemes to perform computations.

B. Contributions

How do we design or adapt modulations so that a valid computation is performed over-the-air? We answer this question by proposing the ChannelComp method. This paper establishes the conditions for computing functions over-the-air using digital modulation. For a given function, these conditions lead to feasibility optimization problems whose solutions give us the parameters of the modulation yielding acceptable computation over-the-air. Moreover, we show how to adapt existing digital or analog modulation schemes using pre-coders to compute the desired function over the MAC. Note that the rationale behind ChannelComp draws from the benefits of digital modulations, yet its scope can also be extended to encompass analog modulations.

Specifically, our contributions are as follows:

- **ChannelComp:** we propose and establish a novel method, the ChannelComp method, to calculate any function over the MAC by digital communications.
- **Simple communication architecture:** one of the key benefits of ChannelComp is its innate compatibility with existing digital systems. By contrast, if we wish to implement AirComp on existing digital systems, it is necessary to enforce analog communication systems to work on top of current digital systems, which is very difficult and impractical [21], [32].
- **Tractable complexity:** we investigate the fundamental limits for valid computations over MAC and devise an optimization to find the parameters of the digital modulation, which is an *NP-hard* problem. To address such complexity, we propose a convex relaxation that can be solved using a solver such as CVX [33].
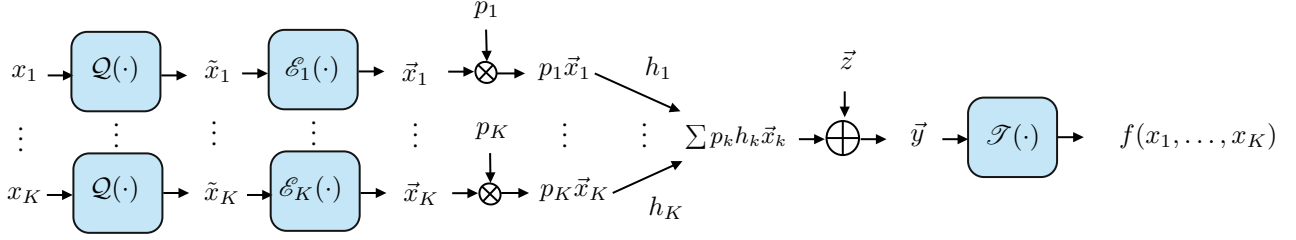
Figure 1: The proposed ChannelComp method. Here x_k, $\tilde{x}_k$, $\vec{x}_k$ are the input, quantized, and modulated values at node k in the network, respectively. The edge server, or computation point, receives $\vec{y} = \sum_{k=1}^{K} h_k p_k \vec{x}_k + \vec{z}$ by the superposition nature of waves. Here, $\vec{z}$ denotes the noise at the receiver, and h_k represents the channel attenuation between node k and the CP server. The tabular mapping $\mathscr{T}(\vec{y})$ maps (or decodes) the value of $\vec{y}$ into the value of function f corresponding to the function's inputs x_k's. The scheme can also apply to analog modulations.

- **All finite-valued functions:** ChannelComp can compute any function with a finite cardinality input domain, which is the case of any digital system because of the digital representation of values. As long as it is possible to count the output of the function, ChannelComp allows us to compute the desired function. For instance, to compute the function that returns the maximum of its input variables, the AirComp method uses the log sum function to approximate the maximum, which returns an approximate value. By contrast, ChannelComp, for a finite input domain of the function, can calculate exactly the maximum of the given input variables.
- **Any digital modulations:** ChannelComp is not only compatible with digital modulation such as quadratic phase shift key (QPSK), quadrature amplitude modulation (QAM) $4, 16, 64, \ldots$, but also FSK modulation, amplitude-shift keying (ASK), or phase modulation (PM), and other digital modulations.
- **Low latency:** ChannelComp provides a wireless aggregation communication system as fast as the AirComp or even faster. This is because ChannelComp adapts the parameters of the digital modulation format such that the receiver computes the desired function, leading to a low latency computation over-the-air.

Besides the mentioned benefits, in the numerical experiments we present in this paper, ChannelComp outperforms AirComp in terms of computation error for various critical functions while consuming the same communication resources. For example, for computing the product function, ChannelComp obtains a 10 dB performance improvement compared to AirComp in terms of the normalized mean square error without using analog modulations while relying only on currently widespread digital modulations.

C. Organization of the paper

The rest of the paper is organized as follows: in Section II, we explain the system model and present the ChannelComp method. Then, we investigate the fundamental limits of the ChannelComp computation over the MAC in Section III. Next, based on our obtained fundamental limits, we characterize in Section IV how to select the digital modulation formats for computing the desired function over the MAC with AWGN and fading effects over the wireless channel. We further describe the receiver architecture for ChannelComp in

Section IV-C. We present the numerical results and the performance comparison between ChannelComp, AirComp, and the traditional OFDMA in Section V. Finally, we conclude the paper in Section VI.

D. Notation

Throughout this paper, scalars are denoted by lowercase letters x, vectors and matrices by lower $\boldsymbol{x}$, and uppercase boldface letters $\boldsymbol{X}$, respectively. We use $\vec{x}$ to represent modulated band-pass signals. Operators are represented by calligraphic notations such as $\mathcal{X}$. The transpose and Hermitian of a matrix $\boldsymbol{X}$ are represented by $\boldsymbol{X}^{\mathsf{T}}$ and $\boldsymbol{X}^{\mathsf{H}}$, respectively. We further use $\otimes$ to show the Kronecker product. For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|_\infty$ are defined as the element-wise ℓ_1 and ℓ_∞ norms, respectively. We define $\|\boldsymbol{X}\|_2$ and $\|\boldsymbol{X}\|_{\mathrm{F}}$ as the spectral and Frobenius of the matrix $\boldsymbol{X}$, respectively. Also, $\|\boldsymbol{X}\|_{1 \to 1}$ is defined as $\max_{j \in [N]} \sum_i |x_{i,j}|$ where $x_{i,j}$ denotes (i, j) element of matrix $\boldsymbol{X}$.

We define $\mathcal{R}_f$ as the range of function f, and its cardinality by $|\mathcal{R}_f|$. For an integer N, $[N]$ corresponds to the set $\{1, 2, \ldots, N\}$. The finite field of size q with subset of the integers $\{0, 1, 2, \ldots, q-1\}$ is represented by $\mathbb{F}_q \subset \mathbb{Z}$. We use $\boldsymbol{X} \succeq \boldsymbol{0}$ to show that $\boldsymbol{X}$ is a positive semidefinite matrix. Finally, we define the $\mathrm{diag} : \mathbb{C}^N \mapsto \mathbb{C}^{N \times N}$ operator for a vector $\boldsymbol{x} \in \mathbb{C}^N$ that puts the input vector on the output matrix main diagonal and zero elsewhere.

II. SYSTEM MODEL AND METHOD FORMULATION

Consider a communication network in which there exists a computation point (CP) server with K nodes. The nodes communicate with the CP through a shared communication channel. The CP aims to compute the desired function $f(x_1, x_2, \ldots, x_K)$ whose input $x_k \in \mathbb{F}_q$ is a value owned by node k. In particular, we assume that the nodes transmit x_k's by digital communications. In such digital communication systems, all the nodes transmit their values over the MAC to compute the function f at the CP.

To perform a digital transmission, the usual procedure is the following: each value x_k is quantized into a scalar $\tilde{x}_k := \mathcal{Q}(x_k)$ with q possible values, where $\mathcal{Q}(\cdot)$ is the quantizer and q equals 2 to the power of the number of quantization bits. Then, the resultant vector is mapped into the digitally modulated signal $\vec{x}_k$ using encoder $\mathscr{E}_k(\cdot)$, i.e., $\vec{x}_k = \mathscr{E}_k(\tilde{x}_k)$. The signal $\vec{x}_k$ is what node k transmits
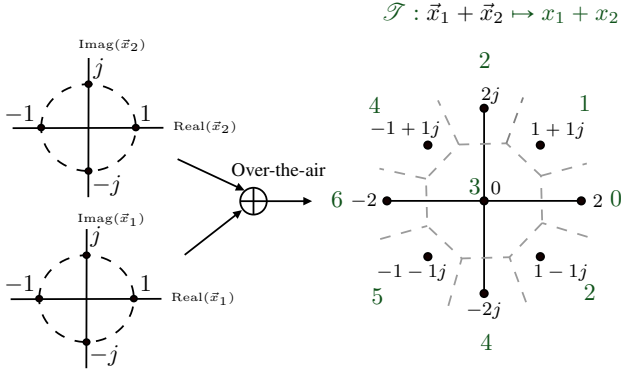
Figure 2: QPSK for 2 bit quantization and two nodes $K = 2$. On the left, there are two constellations (each per node); on the right, the resulting constellation at CP under the assumption of a noise-free channel.
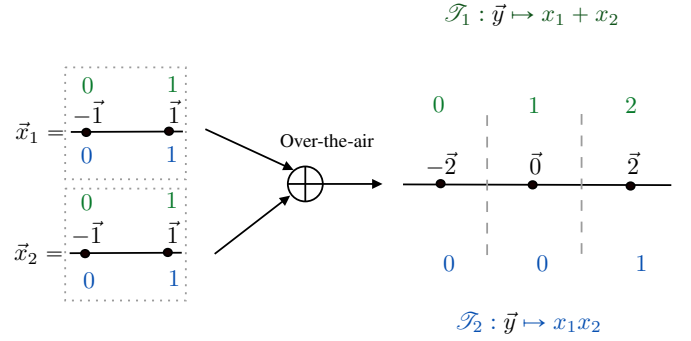

Figure 3: The sum and product computation using the BPSK modulation. The correspondence between the constellation points and the corresponding input/output values may differ based on the defined function. The corresponding values for the summation and product functions are represented using green and blue color codes, respectively.

over the communication channel. Since all the nodes transmit simultaneously[1] and over the same frequency or codes, the CP server receives the summation of all $\vec{x}_k$'s, over the MAC during the one-time slot [1], i.e.

$$\vec{y} = \sum_{k=1}^{K} h_k p_k \vec{x}_k + \vec{z}, \qquad (1)$$

where $\vec{y}$ is physically generated by the superposition nature of electromagnetic waves, h_k denotes the channel coefficient between node k and the CP server, p_k is the transmit power used by node k, and $\vec{z}$ is a receiver noise that the receivers unavoidably experience. Following the usual modeling in digital communications, we model such noise as an additive white Gaussian noise (AWGN) process with zero-mean and variance σ_z^2, which is circularly symmetric. To compute the function f, we need to use mapping (Tabular mapping) $\mathscr{T}(\vec{y})$ based on the resultant constellation diagram of signal $\vec{y}$ (see Figure 1). Note that this system model is identical to the AirComp system model over the MAC [5], except for digital modulation, instead of analog modulation, to communicate and for the trivial tabular mapping. Hence, this system model facilitates low-latency communication for a large number of nodes, as demonstrated by [4]. Moreover, the ChannelComp system model can perform *general* function computations faster than AirComp, as the measured value obtained via tabular mapping $\mathscr{T}(\cdot)$ corresponds exactly to the computation result, whereas the desired value in AirComp requires post-coders for computation [15].

Note that each digital value $\vec{x}_k$ is selected from finite q possible outputs of $\mathcal{Q}(\cdot)$, thereby the received signal $\vec{y}$ has finite constellation points in the absence of AWGN. The summation in (1) induces a specific constellation diagram of $\vec{y}$ that depends on the number of nodes K and on which modulations have been used for each $\vec{x}_k$. Note that such a resulting constellation diagram is a deformation of the original constellation diagram of the transmitting nodes. Figure 2 shows this deformation for the case of using QPSK for two

nodes, i.e., $K = 2$. Recall that our goal is to compute the desired function f. Then, the mapping $\mathscr{T}(\vec{y})$ must be determined such that its output approaches the value of function f as much as possible. Mathematically, at time t, the modulation vectors $\vec{x}_1, \ldots, \vec{x}_K$ with the map $\mathscr{T}$ can be found from the following optimization

$$\mathscr{T}^*, \vec{x}_1^*, \ldots, \vec{x}_K^* = \underset{\mathscr{T}, \vec{x}_1, \ldots, \vec{x}_K}{\operatorname{argmin}} \sum_{x_1, \ldots, x_K \in \mathcal{D}_f} \left| f(x_1, \ldots, x_K) - \mathscr{T}(\vec{y}) \right|^2,$$
$$(2)$$

where $\mathcal{D}_f$ is the domain of function f. To illustrate optimization problem (2), let us consider a simple case with BPSK modulation where the scalar value x_k is one-bit quantized to $\tilde{x}_k \in \{0, 1\}$. Then, we have $\tilde{x}_k = x_k + e_k$, where e_k denotes the quantization error at node k. Hence, the resultant modulated symbol is

$$\vec{x}_k = \mathscr{E}_k(\tilde{x}_k) = \begin{cases} A_c, & \text{if } \tilde{x}_k = 1, \\ -A_c, & \text{if } \tilde{x}_k = 0. \end{cases} \qquad (3)$$

Here, A_c is the amplitude of the carrier signal[2]. Then, assuming a noiseless MAC, the received signal by the CP experiences a constellation formed by the summation of $\vec{y} = \sum_{k=1}^{K} \mathscr{E}_k(x_k)$. For computing the summation function $f(x_1, x_2) = x_1 + x_2$, the tabular $\mathscr{T}_1$ is a simple map that assigns values of the constellation diagram of $\vec{y}$ to the corresponding output of f, i.e.,

$$\mathscr{T}_1(\{-\vec{2}\}) = 0, \quad \mathscr{T}_1(\{\vec{0}\}) = 1, \quad \mathscr{T}_1(\{\vec{2}\}) = 2, \qquad (4)$$

see Figure 3. Similarly, for the product function $f(\boldsymbol{x}) = x_1 x_2$, we can check that this function can be computed by using the following tabular map $\mathscr{T}_2$,

$$\mathscr{T}_2(\{-\vec{2}\}) = 0, \quad \mathscr{T}_2(\{\vec{0}\}) = 0, \quad \mathscr{T}_2(\{\vec{2}\}) = 1. \qquad (5)$$

This example is shown in Figure 3 using blue color. Therefore, we are able to compute a binary summation and product over

---

[1]It is assumed that the synchronization among all the nodes and the CP is perfect. However, the existing techniques of AirComp for solving imperfect synchronization, e.g., [34] can be applied to our system model due to its similarity to the AirComp model.

[2]We let $\mathscr{E}(\cdot)$ be any modulation, whether it has an analog or digital carrier or if the input of the data is digital or analog. For the digital carriers, such as PAM and pulse position modulation (PPM), we can encode and decode the digital data for each symbol per time. For the analog data, the quantization step (see Figure 1) allows us to perform the ChannelComp method for either analog modulation, such as amplitude modulation (AM), PM, and frequency modulations (FM), or digital modulation, such as ASK, PSK, FSK, and QAM.

the noiseless MAC. As long as input data are finite points, the resultant overlapped signal would read to finite constellation points, which lets the desired function f be computed by designing the proper Tabular mapping $\mathscr{T}$ to map the finite constellation points to the output of function f. In this regime, the communication system does not need to be changed, and the only requirement is to add a quantizer to the transmitter and a Tabular mapping to the receiver side.

However, if we increase the order of modulation, e.g., to QAM 16, the constellation points shaped by nodes cannot be uniquely mapped to either summation or product function. Moreover, even for a low-order modulation such as BPSK, computing the simple function $f(x_1, x_2) = x_1 + 2x_2$ is not possible (without errors). In the following section, we overcome this limitation and propose a method to adapt modulations to compute any function.

From the previous examples, the question is: which functions are computable using the ChannelComp communication architecture? Therefore, in the next section, we investigate the characteristics of the modulation and desired function f.

III. PRELIMINARY RESULTS FOR NOISELESS MACs

In this section, we introduce the basic ideas of Channel-Comp, and we give some preliminary results for noiseless MACs, symmetric functions, and the special case where all the nodes use an identical modulation. Based on these results, in the next section, we present ChannelComp for general functions, not necessarily symmetric, and for the case where the nodes are allowed to use different modulations.

In the following, we establish the important result that when identical modulation encoders are used, i.e., $\mathscr{E}_k(\cdot) := \mathscr{E}(\cdot)$ for all $k \in [K]$, it restricts the degrees of freedom in Equation (1). Consequently, this limits the class of functions that can be computed over the MAC. Hence, utilizing identical modulation enforces a class of functions f with specific features. Furthermore, we propose a necessary condition on the function f to compute it by ChannelComp uniquely by using an appropriate tabular mapping $\mathscr{T}$.

Proposition 1 (Necessary condition)**.** *Let the K multivariate function $f(x_1, x_2, \ldots, x_K)$ with domain $\mathcal{D}_f$, where $x_k \in \mathcal{D}_f$ for $k \in [K]$ be a symmetric function, i.e.,*

$$f(x_1, \ldots, x_K) = f(\pi(x_1), \ldots, \pi(x_K)), \qquad (6)$$

for all possible permutations by $\pi : \{1, \ldots, K\} \mapsto \{1, \ldots, K\}$. Let each node use the identical modulation $\mathscr{E}$. Then, function f can be computed by the constellation diagram of $\sum_{k=1}^{K} \mathscr{E}(x_k)$.

Proof. The proof is by contradiction. We assume that $K = 2$, and that $f(x_1, x_2)$ is an asymmetric function, i.e., $f(a, b) \neq f(b, a)$ where $a, b \in \mathcal{D}_f$. Then, for a case where $x_1 = a$ and $x_2 = b$, we have $\vec{a}$ and $\vec{b}$ as modulated signal and $\vec{a} + \vec{b}$ would be received by the CP, respectively. For the reverse scenario, i.e., $x_1 = b$ and $x_2 = a$ the CP also observes $\vec{a} + \vec{b}$. Therefore, we have the same constellation point for different values of the asymmetric function f, and it is impossible to assign the same vector $\vec{a} + \vec{b}$ to the two different values of $f(a, b)$ and $f(b, a)$. $\square$

x_1	x_2	$x_1 x_2$	$\vec{x}_1 + \vec{x}_2$
0	0	0	$\vec{0}$
0	1	0	$\vec{1}$
1	1	1	$\vec{2}$
0	2	0	$\vec{2}$
0	3	0	$\vec{3}$
1	2	2	$\vec{3}$
1	3	3	$\vec{4}$
2	2	4	$\vec{4}$

TABLE III: A counterexample showing that the symmetric property is insufficient for a correct computation with identical modulations.

Corollary 1. *The condition in Proposition 1 becomes sufficient for function f with binary domain, i.e., $x_K \in \{0, 1\}$. Hence, any symmetric function f with binary domains can be computed perfectly via one-bit communication.*

Remark 1. *In Proposition 1, we do not make assumptions on the modulation and the domain. Therefore, the proposition is valid for arbitrary modulation with different bits for each node as long as the modulations are identical.*

To see why the condition in Proposition 1 is not sufficient in general, one can check a simple product function $f(x_1, x_2) = x_1 x_2$, where the nodes use the PAM modulation with two bits. Then, some of the function's outputs f overlap their constellation points. Table III shows this case in detail. In Table III, $\mathscr{E}(\cdot)$ is the PAM modulation, and $x_1 x_2$ is a symmetric function concerning x_1 and x_2. This function cannot be computed because a conflict occurs on points $\vec{2}, \vec{3}$ and $\vec{4}$. The constellation points $\vec{2}, \vec{3}$ and $\vec{4}$ need to be assigned simultaneously to the three different values $\{0, 1\}, \{0, 2\}$ and $\{3, 4\}$, respectively.

To give more insights, let us consider the summation $\sum_{k=1}^{K} \mathscr{E}(x_k)$ as a function that maps the q^K points in the domain of a function to a lower number of constellation points over-the-air. Then, we establish in the following Proposition 2 the lower and upper bound on the cardinality of $\mathcal{R}_s$ when using identical modulation.

Proposition 2. *Let $\mathcal{R}_s$ be the range set of summation $\sum_{k=1}^{K} \vec{x}_k$ and $\mathcal{R}_f$ be the range of desired function f. Assume that we have K different modulation points $\vec{x}_k, k = 1, \ldots, K$, where each one consists of q distinct points in 2D space. Then, the aggregation of the modulated signals, i.e., $\sum_{k=1}^{K} \mathscr{E}(x_k)$ cannot impose lower than $(q-1)K + 1$ number of distinct constellation points and greater than $\binom{K+q-1}{q-1}$. Therefore, the following lower and upper bounds for the range of summation $\mathcal{R}_s$ holds*

$$(q-1)K + 1 \leqslant |\mathcal{R}_s| \leqslant \binom{K+q-1}{q-1}. \qquad (7)$$

Proof. See appendix A. $\square$

The upper and lower bounds in Proposition 2 are the limitations of the range of the symmetric functions. As a result, if the range of the desired function f is greater than the number of the constellation points, i.e., $|\mathcal{R}_f| \geqslant |\mathcal{R}_s|$, it is impossible to cover the overall range of function f.

x_1	x_2	$x_1 + 1$	$\vec{x}_1 + \vec{x}_2$
0	0	1	$\vec{0}$
0	1	1	$\vec{1}$
1	0	2	$\vec{1}$
1	1	2	$\vec{2}$

TABLE IV: A counter-example to demonstrate that $|R_s| \geqslant |\mathcal{R}_f|$ is not enough to compute function f.

Based on the above results, the use of identical modulation among all the nodes restricts us to compute only the symmetric functions. To go beyond symmetric functions, we need to change the modulations to allow us to compute any class of functions. In the following section, we propose a method to adapt or design the modulations under different scenarios to obtain a reliable function computation over the MAC.

IV. MODULATION SELECTION FOR NOISY MACS

We start this section by introducing a general notation to model that each node is allowed to use any modulation, not necessarily identical among the nodes. Moreover, this section assumes that the channel is subject to fading and that the receiver experiences noise. We propose a feasibility check that allows us to design which modulation is needed at each node for a correct computation, which we term as *G-ChannelComp* solution algorithm, where G stands for "general". Then, we introduce how to adapt existing modulations' power and phases, resulting in a correct computation, which we term as *E-ChannelComp* solution algorithm, where E stands for "equal". We conclude the section by giving the general architecture of ChannelComp.

Following the power control universally adopted in the AirComp literature [35], i.e., selecting the transmit power as the inverse of the channel as $p_k = h_k^*/|h_k|^2$, we rewrite (1) as follows[3]

$$\vec{y} = \sum_{k=1}^{K} \vec{x}_k + \vec{z}. \tag{8}$$

Therefore, the received signal by the CP, i.e., $\vec{y}$, is equal to $\sum_k \vec{x}_k$, which has finite constellation points. Using the complex/real vector representation of each $\boldsymbol{x}_k$ for digital/analog modulation, we can represent all the possible constellation points as:

$$\vec{s} := \boldsymbol{A}\boldsymbol{x}, \tag{9}$$

where $\boldsymbol{A} \in \{0,1\}^{M \times N}$ ($N := K \times q$ and $M := q^K$) is a binary matrix that selects all the possible cases of nodes to send their bits, and the vector $\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K]^{\mathsf{T}} \in \mathbb{C}^{N \times 1}$ consists of all K blocks of complex vectors, where block k is a modulation of the corresponding node k. Also, the vector $\vec{s} \in \mathbb{C}^{M \times 1}$ denotes the constellation points over the MAC.

Now, equipped with the definitions in (9), we can establish the conditions to ensure the correctness and viability
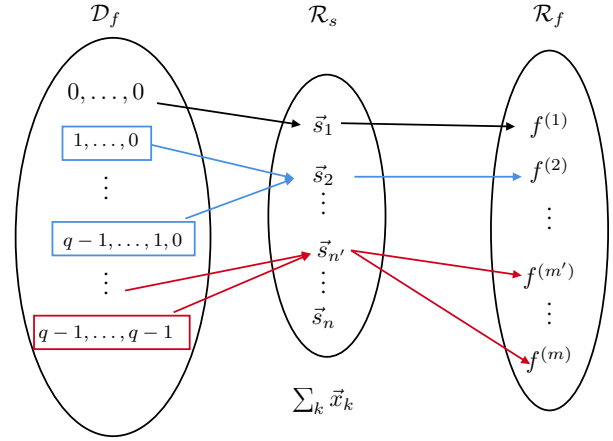
Figure 4: This figure gives the main idea of ChannelComp. It reports the domain $\mathcal{D}_f$, the range of the summation $\mathcal{R}_s$ returned by a noise-free channel, and the desired function $\mathcal{R}_f$ on left, middle, and right, respectively. Two points (blue lines) in the domain of function f create a constellation point $\vec{s}_2$ while the function's output for these two points is the same and equal to $f^{(2)}$. However, for $\vec{s}_{n'}$ the corresponding values of the function are different $f^{(m)} \neq f^{(m')}$. Accordingly, we cannot assign the point $\vec{s}_{n'}$ to these points (red lines) unless we enforce a splitting of $\vec{s}_{n'}$ by a proper selection of the modulations.

of computation via wireless transmission and formulate the modulation vector $\boldsymbol{x}$ to adhere to these established conditions.

A. Modulation Design with Channel Inversion

To compute the function f whose input corresponds to the modulated signal $\vec{x}_1, \ldots, \vec{x}_K$, we need to make sure that the employed modulations allow such computation with adequate accuracy. As mentioned after Proposition 2, one necessary condition under which perfect computation can be possible is related to the range of function R_f and the range of the summation over the channel R_s. Indeed, the number of constellation points at the receiver CP has to be greater than the range of function f, i.e., $|\mathcal{R}_s| \geqslant |\mathcal{R}_f|$. To illustrate that this condition is necessary but not sufficient, we can consider function f to be $x_1 + 1$, for $x_1, x_2 \in \{0,1\}$, where the signals are modulated using BPSK. In Table IV, we illustrated the range points of the function f and summation over the MAC, i.e., $\sum_{k=1}^{K} \vec{x}_k$. While we have $|\mathcal{R}_s| = 3 > 2 = |\mathcal{R}_f|$, the function f cannot be computed by this modulation because of the overlapping on point $\vec{1}$.

Furthermore, the sum $\sum_{k=1}^{K} \vec{x}_k$ must cover the entire range of function f. In fact, for $i \in [M]$, let $f^{(i)}$ be an output of function f for a certain value of inputs $x_1, x_2 \ldots, x_K$, where all x_k's have the same q possible values. Then, if $f^{(i)}$ is different from $f^{(j)}$, the corresponding constellation point $\vec{s}_i$ must not be the same as $\vec{s}_j$ for $i \neq j$, see Figure 4. Hence, to verify if given modulation signals are suitable for computing the desired function f, we pose the following problem:

$$\mathcal{P}_1 = \text{find} \quad \boldsymbol{x}$$
$$\text{s.t. if } f^{(i)} \neq f^{(j)} \Rightarrow \vec{s}_i \neq \vec{s}_j, \ \forall (i,j) \in [M]^2, \tag{10a}$$
$$\|\boldsymbol{x}\|_2^2 \leqslant 1. \tag{10b}$$

Recall that $\boldsymbol{x} \in \mathbb{C}^{N \times 1}$ is the complex modulation vector, and $[M]^2$ means $[M] \times [M]$. Problem $\mathcal{P}_1$ is a feasibility problem

to find a point that satisfies the constraints, which are non-convex and non-smooth. To simplify problem $\mathcal{P}_1$, we replace such constraints by a smooth condition,

$$\mathcal{P}_2 = \text{find} \quad \boldsymbol{x}$$
$$\text{s.t.} \quad |\vec{s}_i - \vec{s}_j|^2 \geqslant \epsilon |f^{(i)} - f^{(j)}|^2, \forall (i,j) \in [M]^2, \quad (11a)$$
$$\|\boldsymbol{x}\|_2^2 \leqslant 1, \quad (11b)$$

where $\epsilon > 0$ is a positive constant. Note that for any sufficiently small ϵ, the solution to Problem $\mathcal{P}_1$ equals the solution to Problem $\mathcal{P}_2$.

Remark 2. *Problem $\mathcal{P}_2$ not only satisfies the constraints of Problem $\mathcal{P}_1$, but also designs the transmit constellation points for achieving the acceptable computation error in noisy communications. The reason is that the right side of the constraints is the computation error, in which a higher computation error due to noise enforces a larger distance of constellation points and enforces more energy. In other words, the distances among constellation points are penalized based on possible computation errors.*

Remark 3. *The constraints in Problem $\mathcal{P}_2$ are compatible with digital or analog modulation, where vector $\boldsymbol{x}$ must be a complex or real vector, respectively. Also, the obtained modulation can compute a larger class of functions than Nomographic functions. The solution to Problem $\mathcal{P}_2$ for the special case of the summation function, i.e., $f = \sum_{k=1}^{K} x_k$, results in AirComp. Indeed, for the computation of Nomographic functions, pre-coder, and post-coder functions are utilized in the AirComp method [36], which can also be incorporated at the beginning and end of our system model illustrated in Figure 1.*

To determine how small ϵ must be such that the solution to Problem $\mathcal{P}_2$ is equal to the solution to Problem $\mathcal{P}_1$, we have the following lemma.

Lemma 1. *Let $\epsilon^{-1} \geqslant \max_{(i,j)\in[M]^2} |f^{(i)} - f^{(j)}|^2$. Then, Problem $\mathcal{P}_2$ is feasible, and thus there exists a modulation vector $\boldsymbol{x}$ satisfying the constraints.*

Proof. See Appendix B. $\qquad\square$

If we rewrite Problem $\mathcal{P}_2$ in terms of variable $\boldsymbol{x}$, we have

$$\mathcal{P}_2 = \text{find} \quad \boldsymbol{x}$$
$$\text{s.t.} \quad |(\boldsymbol{a}_i - \boldsymbol{a}_j)^\mathsf{T} \boldsymbol{x}|^2 \geqslant \epsilon |f^{(i)} - f^{(j)}|^2, \quad (12a)$$
$$\|\boldsymbol{x}\|_2^2 \leqslant 1, \quad (12b)$$

for all $(i,j) \in [M]^2, i \neq j$. Problem $\mathcal{P}_2$ is now a quadratically constrained quadratic programming (QCQP) problem with non-convex constraints. From the literature [37], we note that Problem $\mathcal{P}_2$ is *NP-hard*. Toward overcoming the non-convex constraints, we can use the *lifting trick* [38], in which we recast the constraints (12) as

$$\mathcal{P}_2 = \text{find} \quad \boldsymbol{x}$$
$$\text{s.t.} \quad \text{trace}(\boldsymbol{x}\boldsymbol{x}^\mathsf{H} \boldsymbol{B}_{i,j}) \geqslant \gamma_{i,j}, \quad (13a)$$
$$\text{trace}(\boldsymbol{x}\boldsymbol{x}^\mathsf{H}) \leqslant 1, \quad (13b)$$

Algorithm 1 Approximated G-ChannelComp (solution to $\mathcal{P}_4$)

1: **Input:** Function $f(x_1, \ldots, x_K)$, δ
2: Set $N = qK$, $M = q^K$
3: **Output:** Modulation vector $\boldsymbol{x} \in \mathbb{C}^N$
4: **procedure** DUAL-PRIMAL DC($\{f^{(i)}\}_{i=1}^M$)
5:     **for** $t \leftarrow 1, 2, \ldots,$ **do**
6:         Compute Eigendecomposition $\boldsymbol{X}^{t-1} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\mathsf{H}$
7:         Set $\boldsymbol{G} = \boldsymbol{u}_1 \boldsymbol{u}_1^\mathsf{H}$ where $\boldsymbol{u}_1$ is the first column of matrix $\boldsymbol{U}$
8:         Compute $\boldsymbol{Q} = \boldsymbol{G} + \mu \boldsymbol{X}^{t-1}$
9:         Obtain $\boldsymbol{X}^t = \underset{\boldsymbol{X} \in \mathcal{C}}{\text{argmin}} \quad \frac{\mu}{2}\|\boldsymbol{X}\|_\text{F}^2 + \langle \boldsymbol{X}, \boldsymbol{I}_N - \boldsymbol{Q} \rangle$
10:        Compute $\theta = \frac{\mu}{2}\|\boldsymbol{X}^t\|_\text{F}^2 + \langle \boldsymbol{X}^t, \boldsymbol{I}_N - \boldsymbol{Q} \rangle$
11:        **if** $|\theta| \leqslant \delta$ **then**
12:            $\boldsymbol{X}^* = \boldsymbol{X}^t$
13:            **break**
14:        **end if**
15:     **end for**
16: **end procedure**

---

where $\gamma_{i,j} = \epsilon |f^{(i)} - f^{(j)}|^2$ for all $(i,j) \in [M]^2, i \neq j$ and $\boldsymbol{B}_{i,j} = (\boldsymbol{a}_i - \boldsymbol{a}_j)(\boldsymbol{a}_i - \boldsymbol{a}_j)^\mathsf{T}$. Note that problem (13) is equivalent to problem (11), and thus we denote problem (13) as $\mathcal{P}_2$. Now, if we consider $\boldsymbol{x}\boldsymbol{x}^\mathsf{H}$ as a matrix $\boldsymbol{X} \in \mathbb{C}^{N \times N}$, then problem (13) can be reformulated as

$$\mathcal{P}_2 = \text{find} \quad \boldsymbol{X}$$
$$\text{s.t.} \quad \text{trace}(\boldsymbol{X}\boldsymbol{B}_{i,j}^\mathsf{T}) \geqslant \gamma_{i,j}, \ \text{trace}(\boldsymbol{X}) \leqslant 1, \quad (14a)$$
$$\boldsymbol{X} \succeq \boldsymbol{0}, \quad \text{rank}(\boldsymbol{X}) = 1, \quad (14b)$$

in which the inequality $\boldsymbol{X} \succeq \boldsymbol{0}$ means that the matrix $\boldsymbol{X}$ is symmetric positive semi-definite (PSD). Constraints (14) are linear and convex with respect to matrix $\boldsymbol{X}$, except for the non-convex rank. One possible way to handle such tricky rank constraint is to relax problem (14) by dropping the rank-one constraint in (14) [37], which leads to

$$\mathcal{P}_3 = \text{find} \quad \boldsymbol{X}$$
$$\text{s.t.} \quad \text{trace}(\boldsymbol{X}\boldsymbol{B}_{i,j}^\mathsf{T}) \geqslant \gamma_{i,j}, \ \text{trace}(\boldsymbol{X}) \leqslant 1 \quad (15a)$$
$$\boldsymbol{X} \succeq \boldsymbol{0}. \quad (15b)$$

Problem $\mathcal{P}_3$ is a semi-definite programming problem that can be solved using CVX [33]. Moreover, we term the solution algorithm to problem $\mathcal{P}_3$ as the G-ChannelComp methods. Afterward, if $\boldsymbol{X}^*$, as the solution to Problem $\mathcal{P}_3$, results to be a rank one matrix, $\boldsymbol{x}^*$, the optimal modulation vector solution to Problem $\mathcal{P}_2$ can be obtained via Cholesky decomposition of $\boldsymbol{X}^*$ [39]. Otherwise, if the solution to $\mathcal{P}_3$ does not give a rank one matrix, we can recover a sub-optimal solution to Problem $\mathcal{P}_2$ by the Gaussian randomization method [39]. The Gaussian randomization method has a guaranteed optimality gap [40]. The suboptimality of Problem $\mathcal{P}_3$ may occur when the dimension of the optimization variables becomes large, i.e., $N \gg 1$, which means $K \gg 1$ or $q \gg 1$. To overcome such drawback, the rank-one constraint can be replaced by another equivalent function, which would be successively solved using the primal and dual problems, see [41]. Particularly, the rank-one constraint can be translated to $\text{trace}(\boldsymbol{X}) - \|\boldsymbol{X}\|_2$.

Consequently, we can replace the rank-one constraint in (14) with a penalty term as follows

$$\mathcal{P}_4 = \min_{\boldsymbol{X}} \ \operatorname{trace}(\boldsymbol{X}) - \|\boldsymbol{X}\|_2$$

$$\text{s.t. } \operatorname{trace}(\boldsymbol{X}\boldsymbol{B}_{i,j}^{\mathsf{T}}) \geqslant \gamma_{i,j}, \qquad (16a)$$

$$\boldsymbol{X} \succeq \boldsymbol{0}, \quad \operatorname{trace}(\boldsymbol{X}) \leqslant 1. \qquad (16b)$$

As Problem $\mathcal{P}_4$ remains non-convex, the optimization can be solved using the difference-of-convex (DC) programming [41]. The DC algorithm for Problem $\mathcal{P}_4$ is presented in Algorithm 1.

The relaxation in $\mathcal{P}_3$ disregards the rank constraint in $\mathcal{P}_2$ and can occasionally result in a suboptimal solution, as our manuscript acknowledges. However, when the solution to $\mathcal{P}_3$ (denoted as $\hat{\boldsymbol{X}}$) is a rank-one matrix, it is considered optimal and not suboptimal with respect to problem $\mathcal{P}_2$. In fact, $\mathcal{P}_3$ has the potential to provide optimal solutions to $\mathcal{P}_2$ under certain conditions, particularly when q (level of quantization) or K (number of nodes) are small. The reason is that $\mathcal{P}_3$ is a relaxation of $\mathcal{P}_2$, and if a relaxed problem gives an optimal solution that is feasible to the original problem $\mathcal{P}_2$ (that is defined as a subset of $\mathcal{P}_3$), then it is also optimal to problem $\mathcal{P}_2$. Thus, the rank-one $\hat{\boldsymbol{X}}$ from $\mathcal{P}_3$ is the optimal solution to $\mathcal{P}_2$.

To counter the potential suboptimality of $\mathcal{P}_3$, we proposed $\mathcal{P}_4$ aiming to ensure a low-rank solution as much as possible. This approach is designed to verify the feasibility of rank-one constraints accurately and encourage rank-one solutions. As such, if the solution is a rank-one matrix, it can be regarded as an optimal solution satisfying the constraints of $\mathcal{P}_4$. The main advantage of using $\mathcal{P}_4$ is its ability to reach the stationary point solution, albeit with higher iteration and complexity than $\mathcal{P}_3$. However, since $\mathcal{P}_4$ is not a convex problem, the uniqueness of the solution cannot be guaranteed.

Remark 4. *Optimization Problem $\mathcal{P}_4$ presented in (16) must be solved only once and offline at the CP. To compute the solution, the CP only needs to know which function f is in use and how many nodes are present. Once the CP solves optimization Problem $\mathcal{P}_4$ and sends the encoder $\mathscr{E}_k(\cdot)$ to the nodes, node k can utilize its modulation vector $\vec{x}_k$ to communicate and compute the intended function f over the MAC. Therefore, no optimization must be solved during the computation over-the-air, and the modulation and encoder blocks are integrated.*

B. Adapting Phase and Power of Existing Modulations

To this point, we have established a method for designing the modulation vector for computation over the MAC. However, altering the nodes' modulation formats may sometimes be difficult. In situations where the modulation formats are fixed, we show that it is possible to alter the power and phase of each modulation vector to reshape the constellation diagram and ensure correct and feasible function computation. In order to do so, we do not assume a power control as in AirComp that

enforces channel inversion in (9), but we introduce a general power control. Consequently, the received signal at the CP is

$$\vec{y}_n = \sum_{k=1}^{K} h_k p_k \vec{x}_k + \vec{z}, \qquad (17)$$

where $p_k \in \mathbb{C}$ determines the power and the phase for node k, h_k is the channel coefficient form node k to CP, and $\vec{z} \in \mathbb{C}$ denotes circularly symmetric AWGN noise with zero mean and variance σ_z^2. To alleviate the notation, let us define vectors $\boldsymbol{p} := [p_1, \ldots, p_K]^{\mathsf{T}} \in \mathbb{C}^K$ and $\boldsymbol{h} := [h_1, \ldots, h_K]^{\mathsf{T}} \in \mathbb{C}^K$. Also, we need to define the operator $\mathcal{H}_q : \mathbb{C}^K \mapsto \mathbb{C}^{N \times N}$ as follows

$$\mathcal{H}_q(\boldsymbol{h}) := \operatorname{diag}(\boldsymbol{h}) \otimes \boldsymbol{I}_q = \operatorname{diag}(h_1 \boldsymbol{I}_q, \ldots, h_K \boldsymbol{I}_q), \quad (18)$$

where $\otimes$ denotes Kronecker product and $\boldsymbol{I}_q$ represents the $q \times q$ identity matrix. Then, the constellation points induced by $\boldsymbol{x}$ can be represented as

$$\boldsymbol{A} \underbrace{\mathcal{H}(\boldsymbol{h})\operatorname{diag}(\boldsymbol{x})(\boldsymbol{I}_K \otimes \mathbb{1}_q)}_{:= \boldsymbol{C}} \boldsymbol{p} = \tilde{\boldsymbol{s}}, \qquad (19)$$

where $\mathbb{1}_q$ stands for a vector of size $q \times 1$ whose elements are one, and $\tilde{\boldsymbol{s}} = [\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_M]^{\mathsf{T}} \in \mathbb{C}^M$ denotes the constellation points in the presence of fading. Also, $\boldsymbol{x}$ is the given modulation whose elements can be either complex-valued (digital modulation) or real-valued (analog modulation).

Accordingly, by incorporating the constraints in (12), we propose the following optimization problem to find the minimum modulation powers $\boldsymbol{p}$ for achieving a correct computation without changing the modulation formats:

$$\mathcal{P}_5 = \min_{\boldsymbol{p} \in \mathbb{C}^K} \ \|\boldsymbol{p}\|_2^2$$

$$\text{s.t. } |(\boldsymbol{a}_i - \boldsymbol{a}_j)^{\mathsf{T}} \boldsymbol{C} \boldsymbol{p}|^2 \geqslant \gamma_{i,j}, \ \forall \ (i,j) \in [M]^2. \quad (20)$$

Problem $\mathcal{P}_5$ is a QCQP problem and difficult to solve. Therefore, by following similar steps from (13) to (15), we can pose the following relaxed version of Problem $\mathcal{P}_5$:

$$\mathcal{P}_6 = \min_{\boldsymbol{P} \in \mathbb{C}^{K \times K}} \operatorname{trace}(\boldsymbol{P})$$

$$\text{s.t. } \operatorname{trace}(\boldsymbol{P}\boldsymbol{C}_{i,j}^{\mathsf{T}}) \geqslant \gamma_{i,j}, \ \boldsymbol{P} \succeq \boldsymbol{0}, \qquad (21)$$

where $\boldsymbol{C}_{i,j} := \boldsymbol{C}(\boldsymbol{a}_i - \boldsymbol{a}_j)(\boldsymbol{a}_i - \boldsymbol{a}_j)^{\mathsf{T}} \boldsymbol{C}^{\mathsf{T}}$. Similar to Problem $\mathcal{P}_3$, the solution to Problem $\mathcal{P}_6$, denoted as $\boldsymbol{P}^*$, can give $\boldsymbol{p}^*$ as an optimal solution to $\mathcal{P}_5$ by using Cholesky decomposition or a suboptimal solution by applying the Gaussian randomization method. Finally, the k-th element of $\boldsymbol{p}^*$ is used to determine the power and phase allocation for node k, leading to improved accuracy in over-the-air computations. Moreover, we term the solution algorithm to Problem $\mathcal{P}_6$ as E-ChannelComp.

Remark 5. *Due to the dependence of Problem $\mathcal{P}_6$ on the channel coefficients (h_k) in the case of a fast-fading channel, solving the optimization problem every time the channel changes can be computationally expensive. One alternative solution is separating the channel compensation and modulation adaptation steps. This means that optimization problem (21) only needs to be solved once for a given set of modulations and function f, similar to the approach in (16). However, this*

approach may be less power-efficient compared to the output of optimization problem (21).

In the next section, we describe the structure of the decoder of ChannelComp.

C. Receiver Architecture in ChannelComp

After solving optimization problem $\mathcal{P}_3$ or $\mathcal{P}_6$ at the CP, the modulation vector $\boldsymbol{x}^*$ is determined. Then, the CP broadcasts the whole modulation vector $\boldsymbol{x}^*$'s to the nodes, and node k uses $\boldsymbol{x}_k^*$ for the communication. Afterward, the CP needs to determine the decoder $\mathscr{T}(\cdot)$ based on the obtained modulation vectors $\boldsymbol{x}_k^*$s. The decoder can be straightforwardly determined as long as the solution to either problems $\mathcal{P}_4$ or $\mathcal{P}_6$ is feasible. In particular, we can use the maximum likelihood estimator (MLE) for the decoder and design the decision boundaries based on the deformed constellation points at the CP server. Next, using the Tabular mapping $\mathscr{T}(\cdot)$ on the $\vec{y}(t)$ yields the desired value of the function f. More precisely, let us define $\vec{g}_i := \sum_{k=1}^{K} \vec{x}_k$, the constellation point corresponding to the function $f^{(i)}$. Then, the problem is to find which $\vec{g}_i$'s values were transmitted while we have received $\vec{y}$. Hence, using the MLE estimator, we have

$$\hat{f}^{(i)} = \underset{i}{\operatorname{argmax}} \Pr(\vec{y}|\vec{g}_i), \qquad (22)$$

where $\Pr(\vec{y}|\vec{g}_i) = 1/\sqrt{2\pi\sigma_z^2} \exp\left[-\|\vec{y} - \vec{g}_i\|_2^2/2\sigma_z^2\right]$ follows a Gaussian distribution. Next, taking logarithm results in the following expression

$$\hat{f}^{(i)} = \underset{i}{\operatorname{argmin}} \|\vec{y} - \vec{g}_i\|_2^2. \qquad (23)$$

The last expression generates a Voronoi diagram of the set of all possible constellation points $\{\vec{g}_1, \ldots, \vec{g}_M\}$ with the corresponding Voronoi cells $\{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ [42]. Therefore, the desired value is given by $\hat{f} = \sum_{j=1}^{M} \mathscr{T}^{(j)}(\vec{y})$, where

$$\mathscr{T}^{(j)}(\vec{g}) := \begin{cases} f^{(j)}, & \vec{g} \in \mathcal{V}_j, \\ 0, & \text{otherwise.} \end{cases} \qquad (24)$$

In the case where we cannot satisfy the computation condition in (10), there may be points in which $f^{(i)}$ and $f^{(i+1)}$ have the same constellation point $\vec{g}_i$. In other words, these points have the same Voronoi cell, or $\mathcal{V}_i = \mathcal{V}_{i+1}$, and the decision rule can be replaced by their mean:

$$\mathscr{T}^{(i)}(\vec{g}) = \mathscr{T}^{(i+1)}(\vec{g}) = \begin{cases} \frac{f^{(i)}+f^{(i+1)}}{2}, & \vec{g} \in \mathcal{V}_i, \mathcal{V}_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \qquad (25)$$

To illustrate, consider one simple BPSK example in Figure 3 where there are $K = 2$ nodes for computing the summation function, $f_1(x_1, x_2) = x_1 + x_2$ and product function, $f_2(x_1, x_2) = x_1 x_2$. Let $\mathscr{T}_1$ and $\mathscr{T}_2$ be the tabular maps corresponding to functions f_1 and f_2, respectively. Then, for function f_1 the resultant constellation points are restricted to the set $\{-\vec{2}, \vec{0}, \vec{2}\}$, consequently, it gives the following Voronoi cells,

$$\mathcal{V}_1 := \{\vec{g} \,|\mathrm{Real}(\vec{g}) \leqslant -1\}, \qquad (26a)$$

$$\mathcal{V}_2 := \{\vec{g} \,|-1 < \mathrm{Real}(\vec{g}) \leqslant 1\}, \qquad (26b)$$

$$\mathcal{V}_3 := \{\vec{g} \,|\mathrm{Real}(\vec{g}) > 1\}. \qquad (26c)$$

Hence, by substituting (26) into (24), we can obtain $\mathscr{T}_1 := \sum_{i=1}^{3} \mathscr{T}_1^{(i)}$ for function f_1 as follows

$$\hat{f}_1 := \mathscr{T}_1(\vec{g}) = \begin{cases} 0, & \mathrm{Real}(\vec{g}) \leqslant -1, \\ 1, & -1 < \mathrm{Real}(\vec{g}) \leqslant 1, \\ 2, & \mathrm{Real}(\vec{g}) > 1. \end{cases} \qquad (27)$$

Similarly for the $\mathscr{T}_2$, we have

$$\mathscr{T}_2(\vec{g}) = \begin{cases} 0, & \mathrm{Real}(\vec{g}) \leqslant 1, \\ 1, & \mathrm{Real}(\vec{g}) > 1. \end{cases} \qquad (28)$$

Note that the encoder and decoder in ChannelComp have a similar overhead compared to AirComp. This is because we only map the input and output using the modulation vectors obtained from Problem $\mathcal{P}_4$. The main complexity comes from solving optimization in (15). This optimization must be done offline once before setting up the communication system. Note that thanks to the tailored modulation by ChannelComp, the latency of the communication does not depend on the number of nodes K; accordingly, ChannelComp can handle a massive number of devices with a low latency communication system.

In the following section, we assess the performance of ChannelComp numerically.

V. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of G-ChannelComp (solution to Problem $\mathcal{P}_3$), which designs the modulation among the nodes, and E-ChannelComp (solution to Problem $\mathcal{P}_6$), which adapts phase and power of modulations for the nodes, for several functions with different numbers of nodes. We compare the performance of G-ChannelComp and E-ChannelComp on the following two settings: 1. the standard digital transmission, with the computing method using the orthogonal frequency division multiple access (OFDMA), in which each node uses different frequency channels; and 2. to the AirComp method [5]. In the last subsection, we repeat the comparison of ChannelComp with AirComp and OFDMA methods in the presence of fading.

A. Performance Evaluation of ChannelComp

In the first numerical experiment, we solve optimization problem (15), i.e., (G-ChannelComp) Problem $\mathcal{P}_3$, for four different functions: summation, product, maximum, and fractional function. We consider different numbers of nodes and quantization values for each node, namely $K = 2$ with $q = 8$ and $K = 4$ with $q = 4$. For the case of $K = 4$, we obtained an output $\boldsymbol{X}^*$, which is no longer a rank-one matrix. Hence, we solved optimization problem (16), i.e., Problem $\mathcal{P}_4$, with the penalty parameter to get a better approximation via Algorithm 1. The parameters of Algorithm 1 are set as $\mu = 10^{-4}$ and $\delta = 10^{-3}$ in the simulations.

The resultant modulation vectors are depicted in Figure 5. Figure 5 shows the resultant modulation vectors for the first three functions, i.e., summation, product, and maximum (symmetric functions), and the last fractional function results in different modulation vectors. In the case of the summation function, it turns out that the PAM modulation is the resultant

(a) $f(\boldsymbol{x}) = \sum_{k=1}^{K} x_k$　(b) $f(\boldsymbol{x}) = \prod_{k=1}^{K} x_k$　(c) $f(\boldsymbol{x}) = \max_k x_k$　(d) $f(\boldsymbol{x}) = \frac{x_1+1}{x_2+1}$
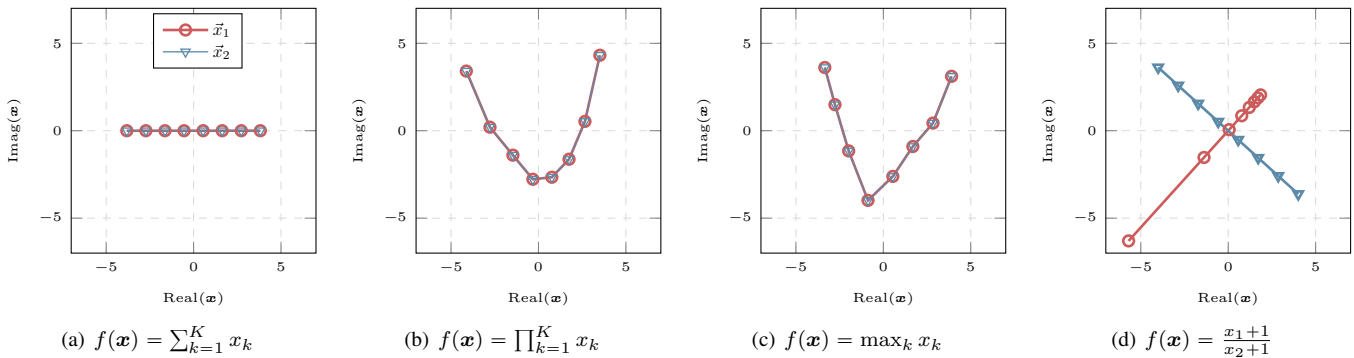
Figure 5: Constellation diagram of the modulation vector for $K = 2$ nodes with $q = 8$ (3 bits). The axes show the real and imaginary values corresponding to the baseband components.
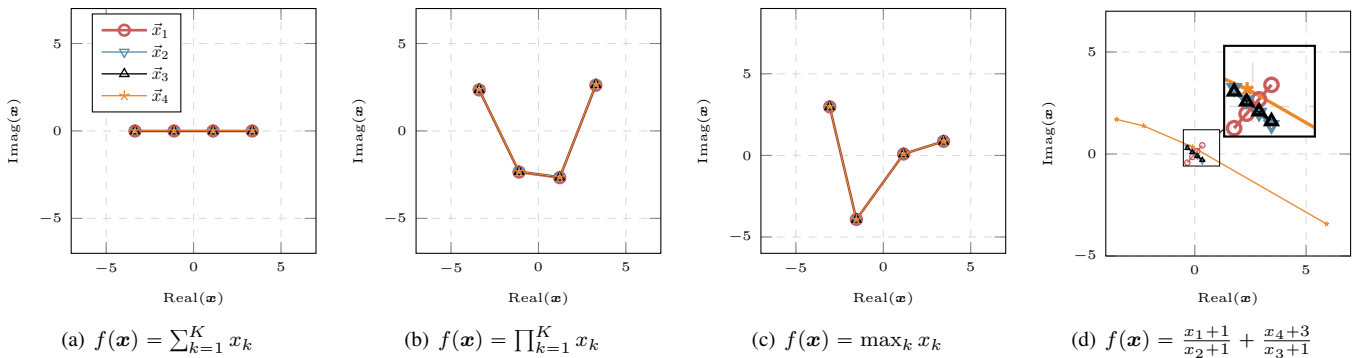

(a) $f(\boldsymbol{x}) = \sum_{k=1}^{K} x_k$　(b) $f(\boldsymbol{x}) = \prod_{k=1}^{K} x_k$　(c) $f(\boldsymbol{x}) = \max_k x_k$　(d) $f(\boldsymbol{x}) = \frac{x_1+1}{x_2+1} + \frac{x_4+3}{x_3+1}$

Figure 6: Constellation diagram of the modulation vector for $K = 4$ nodes with $q = 4$ (2 bits). The axes show the real and imaginary values corresponding to the baseband components.

modulation vector (see Figure 5(a)). Specifically, $q = 4$ is a misconception that QPSK modulation is a more power-efficient alternative for computing the summation. However, this is not the case. It is shown that the PAM modulation is the optimal choice for computing the summation function, as it results in a lower computation error compared to other alternatives.

In Figure 6, we still obtain similar modulations diagram results by increasing the number of nodes to $K = 4$ with $q = 4$. Furthermore, we observe that the modulation vector $\boldsymbol{x}$ of higher order, i.e., larger q, is an interpolated version of the lower order modulation. Similarly, it shows that the lower order modulation can be obtained by sub-sampling from the higher order modulation instead of solving Problem $\mathcal{P}_3$.

In the second experiment, we check the performance of E-ChannelComp for adapting the modulations as in problem (21), i.e., Problem $\mathcal{P}_6$, for the product function $f(\boldsymbol{x}) = \prod_{k=1}^{K} x_k$ with $K = 4$ using QPSK modulation, i.e., $q = 4$ and $\vec{x} = [1, j, -1, j]^\mathsf{T}$. Moreover, to understand the effect of channel fading on the output of Problem $\mathcal{P}_6$, we repeat the experiment for fading channels whose coefficients are generated with normal Gaussian distribution $h_k \sim \mathcal{N}(0, 1)$.

Figure 7 shows the resultant modulations whose power and phases are given by the solution to problem (21) for the ideal and fading channels in Figures 7(a) and 7(b), respectively. We observe that if the fading channel is not considered, E-ChannelComp needs to adapt the power and phase of the
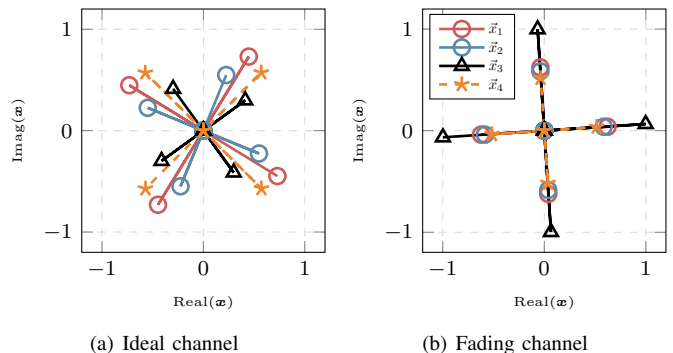

(a) Ideal channel　(b) Fading channel

Figure 7: The resultant modulation vector of (21) for computing the product function $f(\boldsymbol{x}) = \prod_k x_k$ with $K = 4$ nodes and quantization $q = 4$, when the nodes use QPSK modulation. Figures 7(a) and 7(b) show the modulation each node should use, in the case of ideal and fading channels, respectively. The x- and y-axis of the plot illustrate the real and imaginary parts of the baseband components, respectively.

modulations to avoid that destructive overlapping occurring over the MAC. Conversely, in the presence of fading, the channel coefficients can alter the powers and phase of the modulations obtained at the receiver, resulting in a more power-efficient manner than standard OFDMA over the MAC. Specifically, in Figure 7(a), node 2 needs to reduce its power to satisfy the computation constraints in (21). At the same time, a low value of h_2 suggests not compensating the channel effect and allocating more power to this node.

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2023.3324999
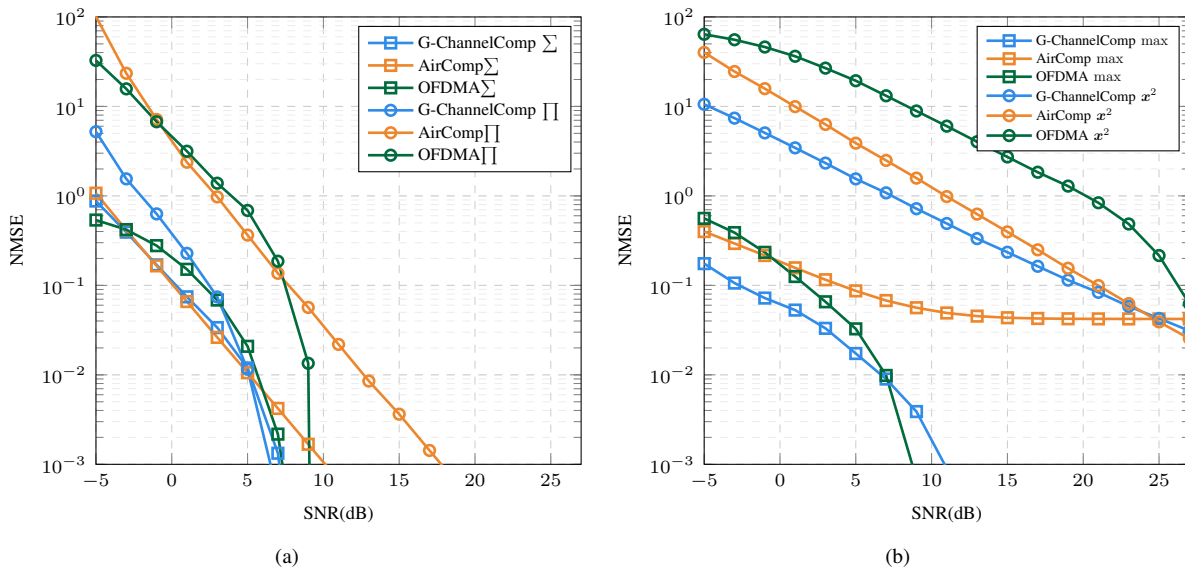
11

(a)

(b)

Figure 8: Performance comparison between G-ChannelComp, AirComp, and OFDMA in terms of NMSE error averaged over $N_s = 100$, when values of the function to be computed are originally quantized. Such input values are the input value $x_k = \{0, 1, \ldots, 7\}$ and the desired functions are $f_1 = \sum_{k=1}^{4} x_k, f_2 = \prod_{k=1}^{4} x_k$, $f_3 = \max_k x_k$, and $f_4 = \sum_{k=1}^{4} x_k^2$.

B. Comparison to AirComp

We compare G-ChannelComp to OFDMA and AirComp, which is analog and mostly intended for summation and product functions. Hence, we compare G-ChannelComp, AirComp, and OFDMA using these functions in different scenarios where the input data has either continuous or discrete values. Specifically, we consider the function's input values originally quantized in the first scenario. Thus, the input values are naturally associated with digital values modulated over the AWGN channel. In other words, we consider the summation function $f = \sum_{k=1}^{4} x_k$, where $x_k \in \{0, 1, 2, \ldots, 8\}$ over a network with $K = 4$ nodes. For a fair comparison, the peak of modulated signals' amplitude of G-ChannelComp is set to be the same as the maximum power for AirComp, which is the maximum value of input value or 8. Furthermore, to characterize the computation error in G-ChannelComp, AirComp, and OFDMA, we use the normalized mean square error (NMSE) metric, which is defined as $\text{NMSE} := \sum_{j=1}^{N_s} |f^{(i)} - \hat{f}_j^{(i)}|^2 / N_s |f^{(i)}|$, where N_s denotes the number of Monte Carlo trials, $f^{(i)}$ denotes the value of the desired function we wish to compute, and $\hat{f}_i^{(j)}$ is the j-th estimated value of $f^{(i)}$ for $j \in [N_s]$.

Figure 8(a) shows the NMSE for different signal-to-noise ratios (SNRs), which is defined as $\text{SNR} := 20 \log(\|\boldsymbol{x}\|_2/\sigma_z)$. We note that G-ChannelComp has a similar performance to AirComp for computing the summation function at the low SNR (less than 5 dB), but as the SNR increases, G-ChannelComp outperforms AirComp. For the product function, G-ChannelComp shows approximately a 10 dB improvement compared to the other methods, even in low SNR scenarios. We repeat this experiment for the maximum $f = \max_k x_K$ and quadratic $f = \sum_{k=1}^{K} x_k^2$ functions in Figure 8(b).

We observe that G-ChannelComp exhibits superior performance compared to the other methods for both functions under examination, thanks to the constructive overlapping
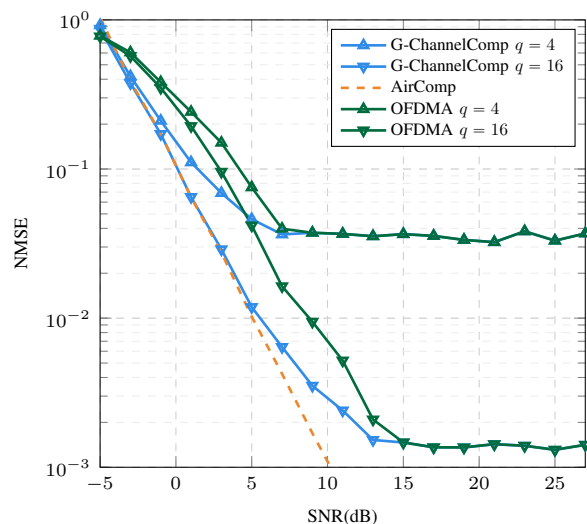

Figure 9: Performance comparison between G-ChannelComp, AirComp, and OFDMA in terms of NMSE error for computing the summation function $f = \sum_{k=1}^{4} x_k$ with the continuous input values x_k in the interval $[0, 7]$, averaged over $N_s = 100$.

of constellation points of all the nodes. Furthermore, it is notable that AirComp cannot accurately compute the maximum function even in low SNR scenarios, likely due to its approximation techniques applied to the maximum function [4]. For the quadratic function, the results suggest that G-ChannelComp outperforms both other methods across a wide range of SNR, with a particularly noticeable advantage in low SNR scenarios.

[4]AirComp approximates the maximum using the log sum function. Indeed, $f(x_1, \ldots, x_K) = \log\left(\sum_{k=1}^{K} \exp(x_k)\right) \approx \max_k x_k$ [32] where we need to set $\mathscr{E}(x_k) := \exp(x_k)$ and $\mathscr{T}(y) := \log(y)$ as encoders and Tabular mapping, respectively. However, ChannelComp computes the exact value for such a function in a noise-free communication by solving optimization in Problem $\mathcal{P}_3$.

In the next experiment reported in Figure 9, we generate uniform random values as continuous numbers, $x_k \sim [0,7]$. Afterward, for G-ChannelComp, these values are quantized with $q = 4$ and $q = 16$ levels (or equivalently 2 and 4 bits) and transmitted over the MAC. Figure 9 shows that the performance of G-ChannelComp is saturated by quantization noise level for low noise cases. However, increasing the number of bits can mitigate this issue. The improved performance observed for AirComp can be attributed to the fact that it does not employ quantization, thus resulting in a transmitted signal with a lower noise level and preserving the signal's quality. Moreover, when the noise variance is very high (SNR less than -3 dB), OFDMA performs better than G-ChannelComp in the summation function, which comes from the input values having limited domain [5] (x_ks are between 0 and 7).

The results of this subsection show that the designed modulation by G-ChannelComp outperforms the other approaches, specifically for non-summation functions. G-ChannelComp and AirComp obtain similar performance for the summation function because AirComp can be considered a special case of G-ChannelComp (Remark 3).

C. E-ChannelComp with QAM Modulations

In this subsection, we analyze the performance of E-ChannelComp for QAM modulation with multiple orders and compare it to OFDMA. We consider a network of $K = 10$ nodes for computing the summation function $f = \sum_k x_k$. Figure 10(a) shows the NMSE performance of ChannelComp and OFDMA using QAM 4, QAM 16, QAM 64, QAM 256 modulations' depicted by $N_s = 1000$ samples randomly selected from the range of the function $f(x_1, \ldots, x_K)$, i.e., $\mathcal{R}_f$. The results show that E-ChannelComp performs better than OFDMA, thanks to constructive overlaps of E-ChannelComp, where multiple signals are transmitted simultaneously and overlap in the frequency domain, resulting in a stronger signal than the individual signals in OFDMA. In addition, increasing the order of modulation results in lower performance for both E-ChannelComp and OFDMA due to reaching the channel's capacity.

In Figure 10(b), we analyze the effect of the number of nodes K on the performance of ChannelComp using QAM $64, 256$ and 1024 modulation in an AWGN channel with SNR $= 10$ dB for computing the summation function. We note that increasing the number of nodes decreases the NMSE error, which comes from constructive interference of node signals.

For the last experiment, we analyze the effect of fading channels on the performance using QPSK modulation for computing the product and max functions of $K = 4$ nodes and $K = 2$, i.e., $f = \prod_{k=1}^{4} x_k$ and $f = \max_k x_k$. Figure 11(a) shows that ChannelComp can achieve better NMSE performance than AirComp in high SNR scenarios, while it has
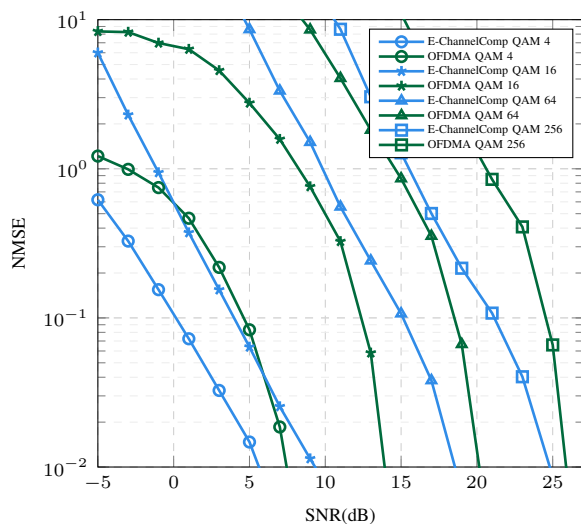
(a)
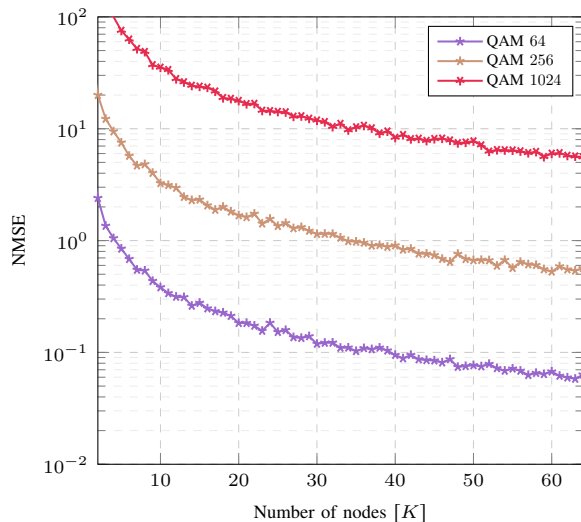
(b)

Figure 10: Performance of adapting modulation using E-ChannelComp over the MAC for the QAM modulation. The function is summation $f = \sum_{k=1}^{K} x_k$ over $N_s = 1000$ samples randomly selected from $\mathcal{R}_f$. Figure 10(a) shows the performance of the E-ChannelComp compared to OFDMA for $k = 10$ nodes. Figure 10(b) shows the performance of the E-ChannelComp for different numbers of nodes K when the SNR $= 10$ dB. Increasing the number of nodes decreases the error thanks to the constructive interference.

a similar performance to OFDMA in low SNR scenarios. We note that the deterioration of E-ChannelCompp performance in this experiment comes from the fact that QPSK modulation is not the optimal modulation for computing the product function (see Figure 6(b)). Similarly, for the max function in Figure 11(b), E-ChannelComp performs better than AirComp for $K = 2$ and $K = 4$ using the same communication resources. Also, OFDMA shows the best performance here, using more bandwidths than the other methods.

We observe that E-ChannelComp enables existing modulation to compute functions over the MAC; however, the performance drops compared to the designed modulation by G-ChannelComp.
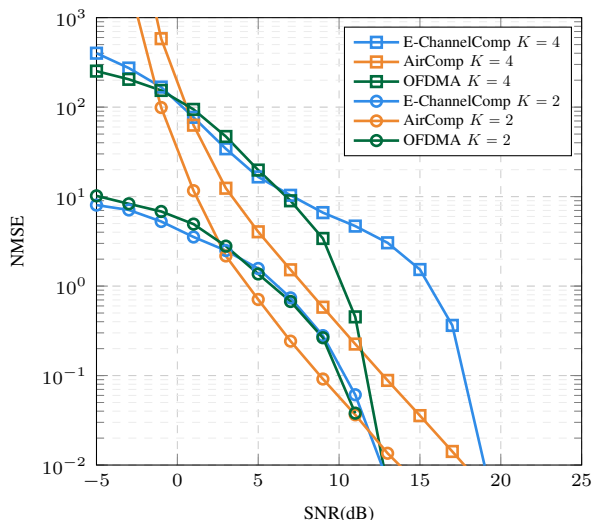
(a) $f = \prod x_k$
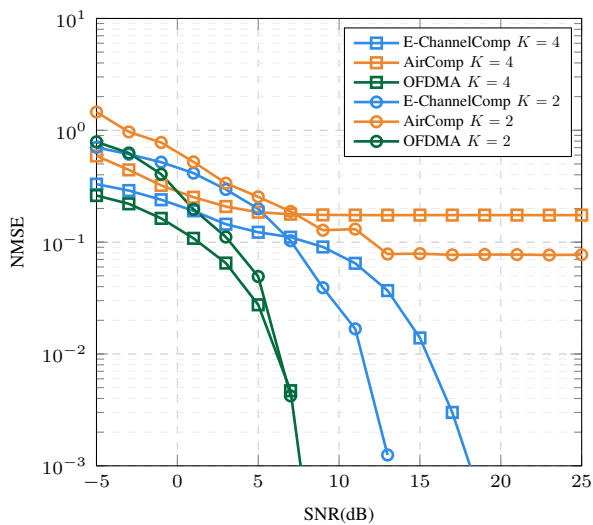
(b) $f = \max x_k$

Figure 11: Performance comparison between E-ChannelComp, AirComp, and OFDMA in terms of NMSE versus the SNR values in the presence of fading channels. We considered $K = 2$, and 4 nodes, and computing the product and the max function using QPSK in 11(a) and 11(b), respectively. The channel coefficients are generated by the Gaussian distribution, i.e., $h_k \sim \mathcal{N}(0, 1)$.

VI. CONCLUSION

In this study, we presented a novel over-the-air computation principle and method called ChannelComp, which utilizes digital or analog modulation to compute functions over multiple access channels. Our proposed method can compute a much broader class of functions compared to the AirComp method, which is restricted to analog modulations. ChannelComp can handle massive devices simultaneously, ensuring strict computation time constraints.

We also proposed an offline modulation selection method based on optimizing a feasibility function and a tabular mapping capable of computing functions using digital modulation. We showed how to adapt existing digital modulation schemes to compute the desired function over the MAC. The Simulation results showed that ChannelComp outperforms analog

AirComp and OFDMA methods in terms of normalized mean squared error with a notable improvement of around 10 dB in computation error.

There are numerous potential avenues for further investigation of ChannelComp, including, but not limited to, the following:

- **Robust solution for stochastic fading channel:** we plan to extend ChannelComp to accommodate general functions through different (digital and analog) modulations for each node and evaluate the impact of stochastic fading.
- **Other optimization perspectives:** we solved a feasibility problem to find modulation vectors for a valid computation over-the-air. One can use the proposed computation to optimize the other objectives, e.g., minimizing the maximum or average computation error.
- **Time variant systems:** Both G-ChannelComp and E-ChannelComp can be extended for designing or adapting the modulation vector for a general time-varying system model. As a result, it enables either computing a sequence of functions over time or stabilizing the computation against the channel changes.
- **MIMO extension for matrix computation:** we intend to expand the current single narrowband antenna system at the transmitters and receiver to broadband multiple-input and multiple-output systems, enabling vector-based computations for several applications, such as matrix computation or federated learning.
- **Sequential transmission:** we plan to introduce repetitions in the transmissions and look at the sequence of received symbols across a trellis diagram to tradeoff modulation complexity with computation time and computation precision.
- **Machine learning applications:** we aim to demonstrate that ChannelComp has the potential to enhance applications such as federated edge learning significantly.

APPENDIX

A. Proof of Proposition 2

In the case where all nodes use identical modulation $\mathscr{E}_k = \mathscr{E}$ for $k \in [K]$, the summation over-the-air can be considered as a symmetric function $\mathscr{G}(\cdot) = \sum_{k=1}^{K} \mathscr{E}(\cdot)$. Consequently, to compute the upper bound, the outputs of the function need to be as much as possible distinguishable, which means that all the outputs of $\mathscr{E}(\cdot)$ are distinct. Therefore, the range of $\mathscr{E}(x)$ would be q for $x \in \{0, 1, \ldots, q - 1\}$. Next, we need to count the number of possible distinct outputs for the function $\mathscr{G}(x_1, \ldots, x_K) = \sum_{k=1}^{K} \mathscr{E}(x_k)$. Each x_k has q distinct values, and we need to count the number of ways to select one level of x_k such that the output of function $\mathscr{G}$ be different. This is exactly the combinatorial balls and bins problem [43], in which we seek the number of possible ways to put n indistinguishable balls into k distinguishable bins. Here, bins are the possible values of variables x_ks, and balls are the modulation map $\mathscr{E}$. Therefore, using the formula of balls and bins, we get the following upper bound $|\mathcal{R}_s| \leqslant \binom{K+q-1}{q-1}$.

For the lower bound, we should consider a scenario where the outputs of each $\mathscr{E}(x_k)$ overlap with each other. The most overlapping occurs when $\mathscr{E}$ is a linear function. As a result, for the function $\mathscr{G}$, we have that

$$\mathscr{G}(x_1, \ldots, x_K) = \sum_{k=1}^{K} \mathscr{E}(x_k) = \mathscr{E}\left(\sum_{k=1}^{K} x_k\right). \qquad (29)$$

Now, because $\mathscr{E}(\cdot)$ is also a bijective mapping, the cardinality of the range of the function $\mathscr{G}(\cdot)$ equals to the cardinality of the its input domain, i.e. $\sum_{k=1}^{K} x_k$. Using that $x_k \in \{0, 1, \ldots, q-1\}$, we note that the number of possible values of the range of $\sum_{k=1}^{K} x_k$ is $K(q-1) + 1$.

B. Proof of Lemma 1

We first rewrite problem (11) with a new constraint on $\|s\|_2^2 \leqslant \tilde{\epsilon}$ as follows:

$$\tilde{\mathcal{P}} = \text{find} \quad x$$
$$\text{s.t.} \quad |\vec{s}_i - \vec{s}_j|^2 \geqslant |f^{(i)} - f^{(j)}|^2, \quad \forall(i,j) \in [M]^2, \quad (30a)$$
$$\|s\|_2^2 \leqslant \tilde{\epsilon}. \qquad (30b)$$

Then, let $\mathcal{L}(x, \Lambda, \rho)$ be the Lagrangian function associated to problem (30) as

$$\mathcal{L}(x, \Lambda, \rho) = \langle \mathcal{D}(Ax), \Lambda \rangle + \langle \mathcal{D}(f), \Lambda \rangle + \rho(x^{\top} A^{\top} Ax - \tilde{\epsilon}), \qquad (31)$$

where $\mathcal{D} : \mathbb{R}^M \mapsto \mathbb{R}^{M \times M}$ denotes a distance operator, which for vector $u \in \mathbb{R}^M$ is defined as

$$\mathcal{D}(u) := (u \odot u)\mathbb{1}_M^{\top} + \mathbb{1}_M(u \odot u)^{\top} - 2uu^{\top}, \qquad (32)$$

in which $\mathbb{1}_M$ is a vector of size $M \times 1$ whose all elements are equal to one. Also, the Λ matrix includes Lagrangian multipliers such that $[\Lambda]_{i,j} = \lambda_{i,j}$. Hence, the Lagrangian cost function $\mathcal{L}(x, \Lambda, \rho)$ in (31) can be equivalent to the following term

$$\mathcal{L} = \langle (Ax \odot Ax)\mathbb{1}_M^{\top}, \Lambda \rangle + \langle \mathbb{1}_M(Ax \odot Ax)^{\top}, \Lambda \rangle - 2\langle Axx^{\top} A^{\top} \Lambda \rangle + \langle \mathcal{D}(f), \Lambda \rangle + \rho(x^{\top} A^{\top} Ax - \tilde{\epsilon}). \qquad (33)$$

By reformulating the previous equations, one can reach the following expression.

$$\mathcal{L}(x, \Lambda, \rho) = \langle xx^{\top}, \underbrace{A^{\top}(\rho I + \Lambda - \text{Diag}(\Lambda \mathbb{1}_M))A}_{Z} \rangle + \langle \mathcal{D}(f), \Lambda \rangle - \rho\tilde{\epsilon}. \qquad (34)$$

Next, minimizing the above Lagrangian cost function over variable x gives us

$$\mathcal{L}(x, \Lambda, \rho) = \begin{cases} \langle \mathcal{D}(f), \Lambda \rangle - \rho\tilde{\epsilon}, & Z \succeq 0, \\ -\infty, & \text{otherwise}, \end{cases} \qquad (35)$$

in which the constraint $Z \succeq 0$ or equivalently $Y := \rho I + \Lambda - \text{Diag}(\Lambda \mathbb{1}_M) \succeq 0$ forces the maximum summation of row of Lagrangian multipliers Λ to be upper bounded by ρ,

i.e., $\|\Lambda\|_{1 \to 1} \leqslant \rho$. Strictly speaking, the diagonal element i of matrix Y is given by

$$[Y]_{ii} = [\rho I + \Lambda - \text{Diag}(\Lambda \mathbb{1}_M)]_{i,i}$$
$$= \rho + \lambda_{i,i} - \sum_{j=1}^{M} \lambda_{i,j}, \qquad (36)$$

where $\lambda_{i,j}$ denotes the (i,j) element of Λ. Since all $\lambda_{i,j}$s are positive numbers, to make the diagonal elements of the matrix Y positive, we need to set $\rho \geqslant \|\Lambda\|_{1 \to 1}$. Accordingly, we have

$$\rho \geqslant \|\Lambda\|_{1 \to 1} = \max_i \sum_{j=1}^{M} \lambda_{i,j}$$
$$\geqslant \max_i \sum_{j=1, j\neq i}^{M} \lambda_{i,j}$$
$$\geqslant \sum_{j=1, j\neq i}^{M} \lambda_{i,j}$$
$$= \sum_{j=1}^{M} \lambda_{i,j} - \lambda_{i,i}, \qquad (37)$$

where the last inequality means that all diagonal elements in (36) are positive. Then, one can write the matrix Y as $Y = \Lambda + \Gamma$ where $\Gamma = \rho I - \text{Diag}(\Lambda \mathbb{1}_M)$ is a diagonal matrix with positive elements, i.e., $\Gamma_{ii} \geqslant \lambda_{ii} \geqslant 0$. On the one hand, matrix Γ is a diagonal matrix with positive elements, which means it is a PSD matrix. On the other hand, matrix Λ is a PSD matrix since it is a matrix of the Lagrangian multiplier. Therefore, because Λ and Γ both are PSD matrices, accordingly, Y is also a PSD matrix.

Next, using this fact yields the following upper bound on the Lagrangian for the PSD case of Z where we use Hölder's inequality on the inner product in (35) to obtain

$$\mathcal{L}(x, \Lambda, \rho) \leqslant \|\Lambda\|_{1 \to 1} \sum_{i=1}^{M} \max_j [\mathcal{D}(f)]_{i,j} - \rho\epsilon$$
$$\leqslant \|\Lambda\|_{1 \to 1} M\|\mathcal{D}(f)\|_{\infty} - \rho\epsilon$$
$$\leqslant \rho(M\|\mathcal{D}(f)\|_{\infty} - \epsilon).$$

Hence, to make the problem feasible, the Lagrangian needs to be negative, otherwise maximizing over Λ leads to ∞ for $\mathcal{L}(x, \Lambda, \rho)$. Therefore, by setting $\tilde{\epsilon} \geqslant M\|\mathcal{D}(f)\|_{\infty}$, or equivalently $\epsilon^{-1} \geqslant \frac{\|s\|_2^2}{\|A\|} = \|\mathcal{D}(f)\|_{\infty}$, the problem is always feasible.

REFERENCES

[1] S. Razavikia, J. M. B. d. Silva, and C. Fischione, "Computing functions over-the-air using digital modulations," in *IEEE ICC*, 2023.

[2] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, 2021.

[3] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland *et al.*, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, 2021.

[4] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2019.

[5] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Sig. Proc.*, vol. 61, no. 20, pp. 4893–4906, 2013.

[6] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Info. Theo.*, vol. 53, no. 10, 2007.

[7] H. Hellström, J. M. B. da Silva Jr, M. M. Amiri, M. Chen, V. Fodor *et al.*, "Wireless for machine learning: A survey," *Foundations and Trends® in Signal Processing*, vol. 15, no. 4, pp. 290–399, 2022.

[8] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, 2020.

[9] A. Şahin, B. Everette, and S. S. M. Hoque, "Distributed learning over a wireless network with FSK-based majority vote," in *International Conference on Advanced Communication Technologies and Networking*. IEEE, 2021, pp. 1–9.

[10] A. Şahin and R. Yang, "Over-the-air computation over balanced numerals," in *IEEE Globecom Workshops*, 2022, pp. 347–352.

[11] M. Gastpar, "Uncoded transmission is exactly optimal for a simple Gaussian "sensor" network," *IEEE Trans. Info. Theo.*, vol. 54, no. 11, pp. 5247–5251, 2008.

[12] R. Soundararajan and S. Vishwanath, "Communicating linear functions of correlated Gaussian sources over a MAC," *IEEE Trans. Info. Theo.*, vol. 58, no. 3, pp. 1853–1860, 2012.

[13] A. B. Wagner, S. Tavildar, and P. Viswanath, "Rate region of the quadratic Gaussian two-encoder source-coding problem," *IEEE Trans. Info. Theo.*, vol. 54, no. 5, pp. 1938–1961, 2008.

[14] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Info. Theo.*, vol. 57, no. 10, pp. 6463–6486, 2011.

[15] M. Goldenbaum, H. Boche, and S. Stańczak, "Nomographic functions: Efficient computation in clustered Gaussian sensor networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2093–2105, 2014.

[16] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Over-the-air computation for IoT networks: Computing multiple functions with antenna arrays," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5296–5306, 2018.

[17] F. Ang, L. Chen, N. Zhao, Y. Chen, and F. R. Yu, "Robust design for massive CSI acquisition in analog function computation networks," *IEEE Trans. Veh. Tech.*, vol. 68, no. 3, pp. 2361–2373, 2019.

[18] M. Goldenbaum and S. Stanczak, "Robust analog function computation via wireless multiple-access channels," *IEEE Trans. on Commun.*, vol. 61, no. 9, pp. 3863–3877, 2013.

[19] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[20] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[21] Z. Wang, Y. Zhao, Y. Zhou, Y. Shi, C. Jiang *et al.*, "Over-the-air computation: Foundations, technologies, and applications," *arXiv preprint arXiv:2210.10524*, 2022.

[22] K. Huang, "How does edge AI change the principles of communications and computing?" 2022, distinguished lecture. [Online]. Available: https://www.digitalfutures.kth.se/event/distinguished-lecture-kaibin-huang-the-university-of-hong-kong/

[23] X. Zhao, L. You, R. Cao, Y. Shao, and L. Fu, "Broadband digital over-the-air computation for asynchronous federated edge learning," in *IEEE ICC*, 2022, pp. 5359–5364.

[24] A. Sahin, B. Everette, and S. S. M. Hoque, "Over-the-air computation with DFT-spread OFDM for federated edge learning." in *WCNC*, 2022, pp. 1886–1891.

[25] M. H. Adeli and A. Şahin, "Multi-cell non-coherent over-the-air computation for federated edge learning," in *IEEE ICC*, 2022.

[26] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Int. Conf. Mach. Learn.* PMLR, 2018, pp. 560–569.

[27] M. Tang, S. Cai, and V. K. Lau, "Radix-partition-based over-the-air aggregation and low-complexity state estimation for IoT systems over wireless fading channels," *IEEE Trans. Sig. Proc.*, vol. 70, pp. 1464–1477, 2022.

[28] M. Goldenbaum, H. Boche, and S. Stańczak, "Reliable computation of Nomographic functions over Gaussian multiple-access channels," in *IEEE ICASSP*, 2013, pp. 4814–4818.

[29] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar *et al.*, "Satellite communications in the new space era: A survey and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 70–109, 2020.

[30] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[31] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz *et al.*, "When internet of things meets blockchain: Challenges in distributed consensus," *IEEE Network*, vol. 33, no. 6, pp. 133–139, 2019.

[32] A. Şahin and R. Yang, "A survey on over-the-air computation," *IEEE Communications Surveys & Tutorials*, 2023.

[33] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," 2014.

[34] S. Razavikia, J. A. Peris, J. M. B. d. Silva, and C. Fischione, "Blind asynchronous over-the-air federated edge learning," in *IEEE Globcom Workshops*, 2022, pp. 1834–1839.

[35] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimal power control for over-the-air computation," in *IEEE Globcom Workshops*, 2019, pp. 1–6.

[36] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Sig. Proc.*, vol. 61, no. 20, pp. 4893–4906, 2013.

[37] N. Sidiropoulos, T. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Sig. Proc.*, vol. 54, no. 6, pp. 2239–2251, 2006.

[38] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[39] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Sig. Processing Mag.*, vol. 27, no. 3, pp. 20–34, 2010.

[40] J. M. B. d. Silva, H. Ghauch, G. Fodor, M. Skoglund, and C. Fischione, "Smart antenna assignment is essential in full-duplex communications," *IEEE Trans. on Commun.*, vol. 69, no. 5, pp. 3450–3466, 2021.

[41] Z. Wang, Y. Shi, Y. Zhou, H. Zhou, and N. Zhang, "Wireless-powered over-the-air computation in intelligent reflecting surface-aided IoT networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1585–1598, 2020.

[42] K. Sugihara and M. Iri, "Construction of the Voronoi diagram for 'one million' generators in single-precision arithmetic," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1471–1484, 1992.

[43] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis.* Cambridge university press, 2017.

Saeed Razavikia (Graduate Student Member, IEEE) received the B.Sc. degree from the Iran University of Science and Technology, in 2016, and the M.Sc. degree in electrical engineering from the Sharif University of Technology, Iran, in 2018. He is currently pursuing his Ph.D. at KTH Royal Institute of Technology as a doctoral student funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP). His research interests include various topics in optimization, theory of machine learning, and statistical signal processing.

José Mairton Barros da Silva, Jr (Member, IEEE) is currently Assistant Professor in the Division of Computer Systems at Uppsala University, Sweden. He received the Ph.D. from KTH Royal Institute of Technology, Stockholm, Sweden, in 2019. He was a postdoctoral researcher at KTH Royal Institute of Technology, Stockholm, Sweden, between 2019-2021. He was a Marie Skłodowska-Curie Postdoctoral Fellow with Princeton University, USA, and KTH Royal Institute of Technology, Sweden, between 2022-2023. He received the B.Sc. (Hons.) and M.Sc. degrees in telecommunications engineering from the Federal University of Ceará, Brazil, in 2012 and 2014, respectively. From July 2012 to March 2015, he was a Research Engineer with the Wireless Telecommunication Research Group (GTEL), Brazil. From Fall/Winter 2013 to 2014, he worked in an internship at Ericsson Research, Stockholm. In Spring/Fall 2018, he was a Visiting Researcher with Rice University, TX, USA. He has served as the Secretary of the IEEE Communications Society Emerging Technology Initiative on Full Duplex Communications between 2018-2021. He has been involved in the organization of many IEEE conferences and workshops, including co-chairing IEEE ICMLCN 2024, IEEE SECON 2022-2023 and IEEE GLOBECOM Workshop on Wireless Communications for Distributed Intelligence 2022-2023. He gave several tutorials at many IEEE flagship conferences, including ICASSP, PIMRC, ICC, and GLOBECOM. His research interests include distributed machine learning and optimization over wireless communications.

Carlo Fischione (Senior Member, IEEE) is full Professor at KTH Royal Institute of Technology, Electrical Engineering and Computer Science, Division of Network and Systems Engineering, Stockholm, Sweden. He is Director of the KTH-Ericsson Data Science Micro Degree Program directed to Ericsson globally, and Chair of the IEEE Machine Learning for Communications Emerging Technologies Initiative. He is distinguished lecturer of the IEEE Communication Society, and the funding Chair of the IEEE International Conference on Machine Learning for Communication and Networking (IEEE ICMLCN 2024). He received the Ph.D. degree in Electrical and Information Engineering (3/3 years) in May 2005 and the Laurea degree in Electronic Engineering (Laurea, Summa cum Laude, 5/5 years) in April 2001, both from University of L'Aquila, Italy, He has held research positions at Massachusetts Institute of Technology, Cambridge, MA (2015, Visiting Professor); Harvard University, Cambridge, MA (2015, Associate); and University of California at Berkeley, CA (2004-2005, Visiting Scholar, and 2007-2008, Research Associate). He is Honorary Professor at University of L'Aquila, Italy, Department of Mathematics, Information Engineering, and Computer Science.

His research interests include applied optimization, wireless, sensor networks, Internet of things, and machine learning. He received a number of awards, such as the "IEEE Communication Society S. O. Rice" award for the best IEEE Transactions on Communications paper of 2018, Best Paper award of IEEE Transactions on Industrial Informatics (2007). He is Editor of IEEE Transactions on Communications (Machine Learning for Communications area) and IEEE Transactions on Machine Learning for Communication and Networking, and has served as Associated Editor of IFAC Automatica (2014-2019). He is Member of IEEE (the Institute of Electrical and Electronic Engineers), and Ordinary Member of DASP (the Italian academy of history Deputazione Abruzzese di Storia Patria).