# Blind Compute-and-Forward

Chen Feng, Danilo Silva *Member, IEEE*, and Frank R. Kschischang *Fellow, IEEE*

***Abstract*—Compute-and-forward (C&F) is a promising new approach to interference management, enjoying several advantages over other information-theoretic schemes. C&F usually requires channel state information (CSI) at the receivers so that an "optimal" scaling factor can be computed for the purposes of decoding. In this paper, a blind C&F scheme—i.e., one not requiring CSI—is developed. Rather than attempting to compute the optimal scaling factor, this new scheme seeks one or more "good" scalars, i.e., scalars which allow correct decoding despite possibly being sub-optimal. The region of all such good scalars is characterized. To find a good scalar, a computationally efficient scheme is proposed, which involves error-detection, a hierarchically organized list, as well as a use of the Smoothing Lemma from lattice theory. Simulation results show that our blind C&F scheme achieves—for a class of nested lattice codes—the same throughput as its CSI-enabled counterpart, at the expense of, approximately, a two-fold increase in computational complexity in the high-throughput region. Moreover, our blind C&F scheme can be applied to multi-source multi-relay networks with a good performance/complexity tradeoff.**

***Index Terms*—Physical-layer network coding, compute-and-forward, nested lattice codes, Smoothing Lemma.**

## I. INTRODUCTION

COMPUTE-AND-FORWARD (C&F) is a promising new approach to physical-layer network coding [1]. It enables wireless relay nodes to compute linear combinations of concurrently transmitted messages directly from interfering signals. As shown in [1], for some wireless relay networks, C&F outperforms other relaying strategies (such as compress-and-forward, amplify-and-forward, and decode-and-forward) in certain SNR regions when channel state information (CSI) is available at the relay nodes. If CSI is also known at the transmitters, the performance of C&F can be greatly improved, achieving the full degrees of freedom for single-hop relay networks [2] providing constant-gap-to-capacity result for multi-hop relay networks [3]. C&F can be further enhanced when wireless relays are equipped with multiple antennas [4].

In addition to wireless relay networks, C&F achieves competitive performance in many other practically-relevant wireless networks (such as distributed antenna systems and small-cell networks) compared to other information-theoretic schemes [5]. Also, C&F gives new rate regions for symmetric interference channels [6] and for many-to-one interference channels [7]. More importantly, C&F enjoys low-complexity decoding (even with off-the-shelf components) [8]–[13]. All of these make C&F particularly attractive for practical implementation. A recent survey of C&F can be found in [14].

In the simplest form of C&F, a relay node receives $\mathbf{y} = \sum_\ell h_\ell \mathbf{x}_\ell + \mathbf{z}$, where $h_\ell$ are channel gains, and $\mathbf{x}_\ell$ are points in a multidimensional lattice. Based on the fact that any integer combination of lattice points is again a lattice point, the relay node selects integer coefficients $a_\ell$ and a scalar $\alpha$, and then attempts to decode the lattice point $\sum_\ell a_\ell \mathbf{x}_\ell$ from the scaled signal

$$\begin{aligned} \alpha \mathbf{y} &= \sum_\ell \alpha h_\ell \mathbf{x}_\ell + \alpha \mathbf{z} \\ &= \sum_\ell a_\ell \mathbf{x}_\ell + \mathbf{n}_{\mathsf{eff}}, \end{aligned} \tag{1}$$

where $\mathbf{n}_{\mathsf{eff}} \triangleq \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha \mathbf{z}$ is the so-called "effective noise." The scalar $\alpha$ and integer coefficients $a_\ell$ are carefully chosen based on channel gains $h_\ell$ so that the effective noise is made (in some sense) small. Hence, the "optimal" scalar $\alpha$ and integer coefficients $a_\ell$ critically depend on CSI.

Prior work on C&F assumes that perfect CSI is available at the relay nodes. In reality, a relay node only has *imperfect* CSI or even no CSI at all. A conventional method of obtaining imperfect CSI is to apply training-based channel estimation. However, this method is not particularly suitable for C&F due to the following two reasons. First, C&F is sensitive to channel estimation error [15]. Second, the requirement of accurate CSI is quite demanding when the number of concurrent transmissions is large [16]. This motivates us to take a completely different approach that eliminates the need for CSI in C&F, hereafter called blind C&F.

A preliminary version of our blind approach is presented in [17], whose basic idea is as follows. Although the optimal scalar is nearly impossible to acquire without CSI, some "good" scalars (that allow correct decoding of linear combinations) can be obtained with a reasonable effort. The set of such good scalars turns out to be bounded and symmetric. Furthermore, when the underlying nested lattice codes of C&F are asymptotically-good (in the sense of [18]), the set of good scalars consists of a union of disks. Based on these properties of good scalars, we propose a blind C&F scheme in [17] that finds a good scalar by "probing" a list of points. This list is created in a hierarchical way in order to control the computational complexity.

In this work, we have greatly enhanced our previous blind approach by introducing two new strategies. The first strategy allows us to significantly reduce the complexity of the probing operation compared to [17]. This is achieved by using the

Manuscript received December 23, 2014; revised July 17, 2015 and January 6, 2016. This paper was presented in part at the IEEE International Symposium on Information Theory, Cambridge, MA, July 2012.

C. Feng was with the Dept. of Elec. & Comp. Eng., University of Toronto, Canada. He is now with the School of Eng., University of British Columbia, Kelowna, Canada, chen.feng@ubc.ca. D. Silva is with the Dept. of Elec. Eng, Federal U. of Santa Catarina, Brazil, danilo@eel.ufsc.br. F. R. Kschischang is with the Dept. of Elec. & Comp. Eng., University of Toronto, Canada, frank@comm.utoronto.ca. The work of D. Silva was supported in part by CNPq-Brazil.

Smoothing Lemma from lattice theory. The second strategy allows us to handle more general nested lattice codes without any increase in computational complexity. Based on these strategies, a new computationally-efficient blind C&F scheme is obtained. This scheme achieves almost the same performance as coherent C&F (its CSI-enabled counterpart) with a modest increase in computational complexity. In particular, our simulation results show that this scheme has roughly twice the complexity of coherent C&F in the high-throughput region. Furthermore, we have discussed the application of our blind C&F scheme to multi-source multi-relay networks. Our simulation results demonstrate the clear advantages of blind C&F over the conventional training-based approach.

## II. Coherent Compute-and-Forward

In this section, we briefly review coherent C&F, which serves as a natural benchmark for blind C&F. We begin with a single building block of coherent C&F, namely, a system of $L$ concurrent transmitters and a single receiver.

In such a system, each transmitter $\ell$ sends a length-$n$ complex vector $\mathbf{x}_\ell \in \mathbb{C}^n$, which satisfies an average power constraint $E\left[\|\mathbf{x}_\ell\|^2\right] \leq nP$. The receiver observes $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$, where $h_\ell \in \mathbb{C}$ are complex-valued channel gains and $\mathbf{z}$ is i.i.d. circularly-symmetric complex Gaussian noise, i.e., $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, N_0 \mathbf{I}_{n\times n})$. The goal of the receiver is to reliably recover a nontrivial linear combination of the transmitted messages based on the received signal $\mathbf{y}$ and the channel gains $h_\ell$, which are assumed to be perfectly known at the receiver.

Nazer and Gastpar [1] proposed an effective coding scheme for the above coherent C&F system. Their scheme makes use of asymptotically-good nested lattice codes proposed by Erez and Zamir [18]. Since asymptotically-good nested lattice codes require very long block lengths and almost unbounded complexity, several practical C&F schemes have recently been developed (e.g., [8]–[13]). Here, we present a generic coherent C&F scheme following our previous work [8].

### A. Lattices and Nested Lattice Codes

To describe our generic coherent C&F scheme, we need some definitions and notations related to complex lattices and nested lattice codes. More details can be found in [8]. Let $T$ be a discrete subring of $\mathbb{C}$ forming a principle ideal domain. Typical examples include Gaussian integers $\mathbb{Z}[i] = \{a + bi : a, b \in \mathbb{Z}\}$ and Eisenstein integers $\mathbb{Z}[\omega] = \{a + b\omega : a, b \in \mathbb{Z},\ \omega = e^{2\pi i/3}\}$. A complex $T$-lattice $\Lambda$ of dimension $n$ is a discrete submodule of $\mathbb{C}^n$, i.e., $\Lambda = \{\sum_{i=1}^{n} r_i \mathbf{g}_i : r_i \in T\}$ for some (linearly independent) basis vectors $\mathbf{g}_1, \ldots, \mathbf{g}_n \in \mathbb{C}^n$.

The nearest-neighbor-quantizer $\mathcal{Q}_\Lambda^{\mathrm{NN}} : \mathbb{C}^n \to \Lambda$ sends a point $\mathbf{x} \in \mathbb{C}^n$ to a nearest lattice point in Euclidean distance, i.e.,

$$\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{x}) = \boldsymbol{\lambda} \in \Lambda, \quad \text{if } \forall \boldsymbol{\lambda}' \in \Lambda : \|\mathbf{x} - \boldsymbol{\lambda}\| \leq \|\mathbf{x} - \boldsymbol{\lambda}'\|, \quad (2)$$

where ties are broken in a systematic manner. The *Voronoi cell* $\mathcal{V}_\Lambda(\boldsymbol{\lambda})$ associated with each $\boldsymbol{\lambda} \in \Lambda$ is defined as the set of all points in $\mathbb{C}^n$ that are closest to $\boldsymbol{\lambda}$, i.e., $\mathcal{V}_\Lambda(\boldsymbol{\lambda}) \triangleq \{\mathbf{x} \in \mathbb{C}^n : \mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{x}) = \boldsymbol{\lambda}\}$. The cell $\mathcal{V}_\Lambda(\mathbf{0})$ associated with the origin is often referred to as the *Voronoi region* of $\Lambda$. The modulo-$\Lambda$ operation (associated with $\mathcal{Q}_\Lambda^{\mathrm{NN}}$) is defined as

$$\mathbf{x} \bmod \Lambda = \mathbf{x} - \mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{x}). \quad (3)$$

Clearly, the modulo-$\Lambda$ operation always outputs a point in the Voronoi region of $\Lambda$.

A *$T$-sublattice* $\Lambda'$ of $\Lambda$ is a subset of $\Lambda$ which is itself a $T$-lattice. Two lattices $\Lambda'$ and $\Lambda$ are said to be *nested* if $\Lambda'$ is a sublattice of $\Lambda$. A *nested lattice code* $\mathcal{L}(\Lambda, \Lambda')$ consists of all the lattice points of $\Lambda$ inside the Voronoi region of $\Lambda'$, i.e.,

$$\mathcal{L}(\Lambda, \Lambda') = \Lambda \cap \mathcal{V}_{\Lambda'}(\mathbf{0}) = \{\boldsymbol{\lambda} \bmod \Lambda' : \boldsymbol{\lambda} \in \Lambda\}. \quad (4)$$

In practice, for reasons of energy-efficiency, it is often useful to consider a translated version of nested lattice codes. For any fixed translation vector $\mathbf{d} \in \mathbb{C}^n$ (which is also referred to as a fixed dither in the literature), a *translated nested lattice code* $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ is defined as

$$\mathcal{L}(\Lambda, \Lambda', \mathbf{d}) = (\Lambda + \mathbf{d}) \cap \mathcal{V}_{\Lambda'}(\mathbf{0}) = \{(\boldsymbol{\lambda} + \mathbf{d}) \bmod \Lambda' : \boldsymbol{\lambda} \in \Lambda\}. \quad (5)$$

### B. Encoding and Decoding

For a pair of nested lattices $\Lambda$ and $\Lambda'$, it is shown in [8] that there is a linear labeling $\varphi : \Lambda \to W$ from lattice points in $\Lambda$ to messages in some message space $W$ satisfying two conditions:

1) a lattice point $\boldsymbol{\lambda}$ receives label $\mathbf{0}$ if and only if it belongs to the sublattice $\Lambda'$;
2) the labeling $\varphi$ is $T$-linear, i.e., for all $r_1, r_2 \in T$ and all $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda$, $\varphi(r_1\boldsymbol{\lambda}_1 + r_2\boldsymbol{\lambda}_2) = r_1\varphi(\boldsymbol{\lambda}_1) + r_2\varphi(\boldsymbol{\lambda}_2)$.

In addition, there is an *inverse map* $\tilde{\varphi} : W \to \Lambda$ from $W$ to $\Lambda$ satisfying two conditions:

1) for all $\mathbf{w} \in W$, $\varphi(\tilde{\varphi}(\mathbf{w})) = \mathbf{w}$;
2) for all $\mathbf{w} \in W$, $\tilde{\varphi}(\mathbf{w}) + \mathbf{d}$ is a codeword in the nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$, i.e., $\tilde{\varphi}(\mathbf{w}) + \mathbf{d} \in \mathcal{V}_{\Lambda'}(\mathbf{0})$.

Equipped with this linear labeling $\varphi$ and the inverse map $\tilde{\varphi}$, we can now describe our coherent C&F scheme. The transmitter $\ell$ sends the signal $\mathbf{x}_\ell = \tilde{\varphi}(\mathbf{w}_\ell) + \mathbf{d}$. The receiver aims to recover a nontrivial linear combination $\mathbf{u} = \sum_\ell a_\ell \mathbf{w}_\ell$. In order to do so, it first processes the received signal $\mathbf{y}$, obtaining $\mathbf{y}' = \alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d}$, where $a_{\mathsf{sum}} = \sum_\ell a_\ell$; it then computes $\hat{\mathbf{u}} = \varphi(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{y}'))$. Decoding is correct if $\hat{\mathbf{u}} = \mathbf{u}$.

To understand the decoder, note that

$$\begin{aligned}\mathbf{y}' &= \sum_\ell \alpha h_\ell \mathbf{x}_\ell + \alpha\mathbf{z} - \sum_\ell a_\ell \mathbf{d} \\ &= \sum_\ell a_\ell(\mathbf{x}_\ell - \mathbf{d}) + \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z} \\ &= \sum_\ell a_\ell \tilde{\varphi}(\mathbf{w}_\ell) + \mathbf{n}_{\mathsf{eff}}, \end{aligned} \quad (6)$$

where $\mathbf{n}_{\mathsf{eff}} \triangleq \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}$ is the effective noise. In other words, the operation $\mathbf{y}' = \alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d}$ induces a "virtual" point-to-point channel with channel input $\sum_\ell a_\ell \tilde{\varphi}(\mathbf{w}_\ell)$ and channel noise $\mathbf{n}_{\mathsf{eff}}$. Since the labeling $\varphi$ is $T$-linear, we have

$$\varphi\left(\sum_\ell a_\ell \tilde{\varphi}(\mathbf{w}_\ell)\right) = \sum_\ell a_\ell \varphi(\tilde{\varphi}(\mathbf{w}_\ell)) = \sum_\ell a_\ell \mathbf{w}_\ell. \quad (7)$$

Hence, decoding is correct if and only if $\varphi(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}_{\mathsf{eff}})) = 0$, or equivalently, $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$.

In many cases, the receiver has the freedom to choose the coefficients $a_\ell$ and the scalar $\alpha$. Intuitively, $\alpha$ and $a_\ell$ should be chosen to "minimize" the effective noise $\mathbf{n}_{\mathsf{eff}}$. In fact, as shown in [1] and [8], this amounts to solving the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2}\right) \\ \text{subject to} \quad & \mathbf{0} \neq \mathbf{a} \in T^L \\ & \alpha \in \mathbb{C} \end{aligned} \tag{8}$$

where $\mathsf{SNR} = P/N_0$, $\mathbf{h} = (h_1, \ldots, h_L)$ and $\mathbf{a} = (a_1, \ldots, a_L)$. In particular, a computation rate

$$R_{\mathsf{comp}}(\mathbf{h}) = \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha^*\mathbf{h} - \mathbf{a}^*\|^2 + |\alpha^*|^2}\right)$$

is achievable, where $(\alpha^*, \mathbf{a}^*)$ is an optimal solution to Problem (8).

To summarize, in a coherent C&F scheme, the receiver first solves Problem (8) to obtain some optimal scalar $\alpha$ and coefficients $a_\ell$. The receiver then computes $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d})\right)$. The decoding is correct, i.e., $\hat{\mathbf{u}} = \mathbf{u}$, if and only if $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$.

### *C. Multi-Source Multi-Relay Network*

Next, we move from a building block of coherent C&F to a multi-source multi-relay network, where $L$ sources are communicating to a single destination through $L$ relays as illustrated in Fig. 1. We assume that the final destination has backhaul links (either wireless or wired) to the relays, but has no direct links to the sources. Such a network model and its variants have been studied in several recent papers [19]–[23].

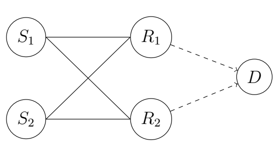



Fig. 1. Illustration for a multi-source multi-relay network with $L = 2$.

A C&F-based transmission scheme typically consists of two phases. In the first phase, all the sources are transmitting their messages simultaneously to all the relays. At the end of this phase, the $m$th relay ($m = 1, \ldots, L$) tries to recover some linear combination $\sum_\ell a_{m\ell}\mathbf{w}_\ell$ of the transmitted messages. In the second phase, the $m$th relay forwards its decoded linear combination to the destination via its backhaul link.

Let $\mathbf{a}_m = (a_{m1}, \ldots, a_{mL}) \in T^L$ be the coefficient vector at relay $m$. Let $\mathbf{A} \in T^{L \times L}$ be a matrix with $\mathbf{a}_m$ at its $m$th row. Clearly, if the matrix $\mathbf{A}$ is full rank and each linear combination associated with $\mathbf{a}_m$ is decoded correctly, then the destination can recover all the original messages by solving a system of linear equations. This leads to the following optimization problem [19], [21]:

$$\begin{aligned} \text{maximize} \quad & \min_{m=1,\ldots,L} \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha_m\mathbf{h}_m - \mathbf{a}_m\|^2 + |\alpha_m|^2}\right) \\ \text{subject to} \quad & \mathsf{rank}(\mathbf{A}) = L \\ & \alpha_1, \ldots, \alpha_L \in \mathbb{C} \end{aligned} \tag{9}$$

where $\mathbf{h}_m = (h_{m1}, \ldots, h_{mL})$ is the channel-gain vector at relay $m$. In particular, a symmetric rate

$$R_{\mathsf{sym}} = \min_{m=1,\ldots,L} \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha_m^*\mathbf{h}_m - \mathbf{a}_m^*\|^2 + |\alpha_m^*|^2}\right) \tag{10}$$

is achievable [21], where $\{(\alpha_m^*, \mathbf{a}_m^*)\}_{m=1}^L$ is an optimal solution to Problem (9). Note that Problem (9) can be solved by the destination via some lattice-based algorithms, if it knows all channel-gain vectors $\{\mathbf{h}_m\}$ [21] (i.e., full CSI). The destination then notifies each relay the optimal coefficient vector $\mathbf{a}_m^*$. Alternatively, Problem (9) can be solved jointly by all the relays at the cost of additional signaling overhead among the relays.

The above scheme can be extended to the case of multiple destinations [22] (via some time-sharing in the second phase, which essentially decomposes the original network into multiple sub-networks each with a single destination). The above scheme can also be extended to the case of non-symmetric rates, leading to a higher sum rate [24]. In addition, if full CSI is available at all the sources/transmitters, power allocation (see, e.g., [20] for $L = 2$ and [23] for general $L$) and network decomposition ( [5]) can be incorporated to boost the performance Since it is often difficult for all the transmitters to acquire full CSI, especially in a large network, in this paper we restrict our attention to the basic scheme described above (rather than its extensions and variants).

## III. Training-Based Approach

Training-based channel estimation is a conventional method of implementing C&F. However, it doesn't scale well with the number of users, as explained in [15]. In particular, with approximations and simulations, [15] shows that the rate loss due to (Gaussian distributed) channel estimation errors is significant under various scenarios for a building block of C&F, especially in the high SNR regime.

Here, we will present some simulation results in terms of the outage probability. Suppose that $\hat{\mathbf{h}} = \mathbf{h} + \mathbf{e}$, where $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_L)$ models the channel estimation error. This model is widely used in training-based channel estimation. For simplicity, assume that the channel gains follow Rayleigh fading, i.e., $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_L)$. Then the key parameter capturing the channel estimation error is

$$\mathsf{SNR}_{\mathsf{est}} = 1/\sigma^2.$$

Recall that without estimation error, the best computation rate for a building block is given by

$$R_{\mathsf{comp}}(\mathbf{h}) = \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha^*\mathbf{h} - \mathbf{a}^*\|^2 + |\alpha^*|^2}\right), \tag{11}$$

where $(\alpha^*, \mathbf{a}^*)$ is an optimal solution to Problem (8). For a fixed message rate $R$, the outage probability is defined as

$$\Pr[\text{outage}] = \Pr[R > R_{\mathsf{comp}}(\mathbf{h})], \tag{12}$$

where $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_L)$.

Similarly, with channel estimation error, the best computation rate is given by

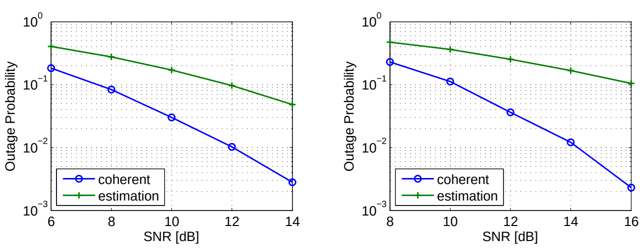
$$\hat{R}_{\mathsf{comp}}(\mathbf{h}) = \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\hat{\alpha}^*\mathbf{h} - \hat{\mathbf{a}}^*\|^2 + |\hat{\alpha}^*|^2}\right), \tag{13}$$

where $(\hat{\alpha}^*, \hat{\mathbf{a}}^*)$ is an optimal solution to the optimization problem

$$\begin{aligned} &\min\ \mathsf{SNR}\|\alpha\hat{\mathbf{h}} - \mathbf{a}\|^2 + |\alpha|^2 \\ &\text{s.t. } \mathbf{0} \neq \mathbf{a} \in \mathbb{Z}[i]^L \\ &\quad\ \ \alpha \in \mathbb{C} \end{aligned}$$

The resulting outage probability is $\Pr[\text{outage}] = \Pr[R > \hat{R}_{\mathsf{comp}}(\mathbf{h})]$.

Through extensive simulations, we observe that the outage probability is sensitive to channel estimation error $\mathbf{e}$, especially when $L$, the number of transmitters, is large. For example, Fig. 2(a) and 2(b) depict the outage probabilities of coherent C&F and estimation-based C&F when $\mathsf{SNR}_{\mathsf{est}}$ is set to $\mathsf{SNR}+6$ dB for the cases of $L = 3$ and $L = 4$, respectively. As we can see, the performance of estimation-based C&F decreases as the number of transmitters increases, even if the difference $\mathsf{SNR}_{\mathsf{est}} - \mathsf{SNR}$ remains the same. Hence, C&F with channel estimation does not scale well with the number of users.




(a) 3 transmitters (b) 4 transmitters

Fig. 2. Outage probabilities of coherent C&F and estimation-based C&F: (a) $L = 3$ and $R = 1$; (b) $L = 4$ and $R = 1$.

Finally, we would like to point out that estimation-based C&F can be applied to a multi-source multi-relay network in a way similar to coherent C&F. Specifically, the destination solves the following optimization problem

$$\begin{aligned} \text{maximize} \quad & \min_{m=1,\ldots,L} \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha_m\hat{\mathbf{h}}_m - \mathbf{a}_m\|^2 + |\alpha_m|^2}\right) \\ \text{subject to} \quad & \mathsf{rank}(\mathbf{A}) = L \\ & \alpha_1, \ldots, \alpha_L \in \mathbb{C} \end{aligned} \tag{14}$$

where $\hat{\mathbf{h}}_m$ is the channel estimation of $\mathbf{h}_m$. In this case, a symmetric rate

$$\hat{R}_{\mathsf{sym}}(\{\mathbf{h}_m\}) = \min_m \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\hat{\alpha}_m^*\mathbf{h}_m - \hat{\mathbf{a}}_m^*\|^2 + |\hat{\alpha}_m^*|^2}\right)$$

is achievable, where $\{(\hat{\alpha}_m^*, \hat{\mathbf{a}}_m^*)\}_{m=1}^L$ is an optimal solution to Problem (14). The corresponding outage probability is $\Pr[\text{outage}] = \Pr[R > \hat{R}_{\mathsf{sym}}(\{\mathbf{h}_m\})]$. The performance of this scheme will be evaluated in Sec. VI-C.

## IV. Blind Compute-and-Forward: General Framework

In this section, we present a general framework for blind C&F. At first glance, it seems very difficult to design a blind C&F scheme, since it is nearly impossible to acquire an optimal scalar $\alpha$ and coefficients $a_\ell$ without CSI. Our key observation is as follows: the receiver does not have to know an optimal scalar $\alpha$ and coefficients $a_\ell$ to ensure successful decoding; instead, it only needs to know some "good" scalars as well as $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$. As we will soon see, equipped with a good scalar and $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$, the receiver is always able to recover a linear combination $\sum_\ell a_\ell \mathbf{w}_\ell$ correctly.

### A. Properties of good scalars

Here, we formally define good scalars and study their basic properties.

*Definition 1:* A scalar $\alpha$ is said to be *good* if $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$ for some coefficients $(a_1, \ldots, a_L) \in T^L \setminus \{\mathbf{0}\}$, and is said to be *bad* otherwise.

Since the decoding is correct if and only if $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$, this justifies the above definition. Note that the effective noise $\mathbf{n}_{\mathsf{eff}}$ depends on the channel gains as well as $\mathbf{x}_\ell$'s and $\mathbf{z}$. Hence, whether a scalar $\alpha$ is good or bad relies on the channel gains and the realizations of $\mathbf{x}_\ell$'s and $\mathbf{z}$.

*Definition 2:* The *good region* of scalars, denoted by $\mathcal{G}_s$, is the set of all good $\alpha$'s, i.e., $\mathcal{G}_s = \{\alpha \in \mathbb{C} : \alpha \text{ is good}\}$.

The good region depends on the channel gains as well as the realizations of $\mathbf{x}_\ell$'s and $\mathbf{z}$. Although the good region is unknown to the receiver without CSI, it is still beneficial to understand some basic properties of the good region, which will play an important role in the design of our blind C&F schemes.

When the nested lattice code is asymptotically good (in the sense of [18]), the good region $\mathcal{G}_s$ has a number of interesting properties. Moreover, these properties still hold (or approximately hold) for commonly-used nested lattice codes.

We note that for asymptotically-good nested lattice codes, a scalar $\alpha$ is good if and only if the message rate $R$ is less than the computation rate

$$R(\alpha, \mathbf{a}) \triangleq \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2}\right) \tag{15}$$

for some $\mathbf{a} \in T^L \setminus \{\mathbf{0}\}$. This allows us to show that the good region $\mathcal{G}_s$ is bounded, symmetric, and consisting of a union of disks.

*Proposition 1:* The good region $\mathcal{G}_s$ is bounded: every good scalar $\alpha$ satisfies $|\alpha|^2 < \mathsf{SNR}/2^R$.

*Proof:* If $\alpha$ is good, then

$$\log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2}\right) > R \text{ for some } \mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

Since

$$\log_2\left(\frac{\mathsf{SNR}}{|\alpha|^2}\right) \geq \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha\mathbf{h}-\mathbf{a}\|^2+|\alpha|^2}\right),$$

we have

$$\log_2\left(\frac{\mathsf{SNR}}{|\alpha|^2}\right) > R.$$

Hence, every good $\alpha$ is bounded by $|\alpha|^2 < \mathsf{SNR}/2^R$. $\square$

*Proposition 2:* The good region $\mathcal{G}_s$ is *symmetric* with respect to rotations by some angle $\theta$. The angle $\theta$ is determined by $T$: $\theta = 90°$ when $T = \mathbb{Z}[i]$; $\theta = 60°$ when $T = \mathbb{Z}[\omega]$.

*Proof:* It suffices to show that if $\alpha$ is good, so is $e^{i\theta}\alpha$ for some angle $\theta$. We need the following fact (from abstract algebra): Let $T$ be a discrete subring of $\mathbb{C}$ forming a principle ideal domain. Let $\mathcal{U}$ be the set of all the units in $T$. Then $\mathcal{U} = \{e^{2\pi k i/n} : k = 0, 1, \ldots, n-1\}$ for some positive integer $n$. For example, when $T = \mathbb{Z}[i]$, the set of units $\mathcal{U} = \{e^0, e^{2\pi i/4}, e^{4\pi i/4}, e^{6\pi i/4}\}$; when $T = \mathbb{Z}[\omega]$, the set of units $\mathcal{U} = \{e^{2\pi k i/6} : k = 0, \ldots, 6\}$. Clearly, the units of $T$ are also the roots of unity.

Now let us choose a unit $u = e^{2\pi i/n}$. If $\alpha$ is good, then

$$\log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|\alpha\mathbf{h}-\mathbf{a}\|^2+|\alpha|^2}\right) > R \text{ for some } \mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

It follows that

$$\log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\|u\alpha\mathbf{h}-u\mathbf{a}\|^2+|u\alpha|^2}\right) > R \text{ for } u\mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

Therefore, $u\alpha$ is also good. In other words, the good region $\mathcal{G}_s$ is symmetric with respect to rotations by $360°/n$, where $n = 4$ when $T = \mathbb{Z}[i]$ and $n = 6$ when $T = \mathbb{Z}[\omega]$. $\square$

*Proposition 3:* The good region $\mathcal{G}_s$ consists of *a union of disks*. These disks are pairwise disjoint if the message rate $R \geq 2$.

*Proof:* Recall that $\alpha$ is good if and only if $R < R(\alpha, \mathbf{a})$ for some $\mathbf{a} \in T^L \setminus \{\mathbf{0}\}$, or equivalently,

$$\mathsf{SNR}\|\alpha\mathbf{h}-\mathbf{a}\|^2 + |\alpha|^2 < \mathsf{SNR}/2^R \text{ for some } \mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

Now observe that the term $\mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2$ is equal to the squared distance between two vectors $(\alpha h_1\sqrt{\mathsf{SNR}}, \ldots, \alpha h_L\sqrt{\mathsf{SNR}}, \alpha)$ and $(a_1\sqrt{\mathsf{SNR}}, \ldots, a_L\sqrt{\mathsf{SNR}}, 0)$. Hence, we have

$$\begin{aligned} &\mathsf{SNR}\|\alpha\mathbf{h}-\mathbf{a}\|^2 + |\alpha|^2 \\ =\ &\mathsf{SNR}\|\alpha^*\mathbf{h}-\mathbf{a}\|^2 + |\alpha^*|^2 + |\alpha-\alpha^*|^2(1+\mathsf{SNR}\|\mathbf{h}\|^2), \end{aligned}$$

where $\alpha^*$ is the MMSE coefficient given by

$$\alpha^* = \frac{\mathsf{SNR}\mathbf{a}\mathbf{h}^\mathsf{H}}{1+\mathsf{SNR}\|\mathbf{h}\|^2}. \tag{16}$$

Recall that $\mathsf{SNR}\|\alpha^*\mathbf{h}-\mathbf{a}\|^2 + |\alpha^*|^2 = \mathsf{SNR}/2^{R(\alpha^*,\mathbf{a})}$. Therefore, $\alpha$ is good if and only if

$$|\alpha-\alpha^*|^2 < \frac{\mathsf{SNR}}{1+\mathsf{SNR}\|\mathbf{h}\|^2}\left(\frac{1}{2^R} - \frac{1}{2^{R(\alpha^*,\mathbf{a})}}\right) \tag{17}$$

for some $\mathbf{a} \in T^L \setminus \{\mathbf{0}\}$. That is, a good $\alpha$ is in some disk of center $\alpha^*$. This proves the first part of Proposition 3.

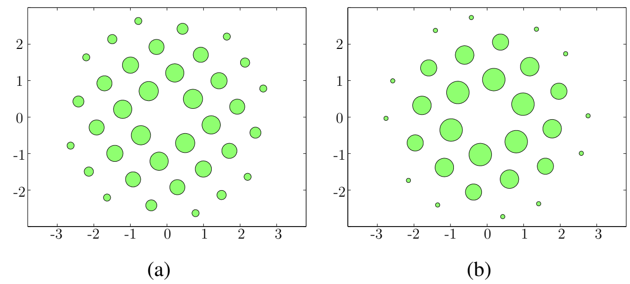



Fig. 3. Good regions for asymptotically-good nested lattice codes: (a) $T = \mathbb{Z}[i]$, $h_1 = -0.93 + 0.65i$, $h_2 = -0.04i$, and $\mathsf{SNR} = 20$dB; (b) $T = \mathbb{Z}[\omega]$, $h_1 = 0.72 + 0.61i$, $h_2 = -0.05i$, and $\mathsf{SNR} = 20$dB.

We proceed to the second part. If two disks overlap, then there exists a common good scalar $\alpha$ within the two disks. That is,

$$\mathsf{SNR}\|\alpha\mathbf{h}-\mathbf{a}\|^2 + |\alpha|^2 < \mathsf{SNR}/2^R \tag{18}$$

and

$$\mathsf{SNR}\|\alpha\mathbf{h}-\mathbf{b}\|^2 + |\alpha|^2 < \mathsf{SNR}/2^R \tag{19}$$

for some $\mathbf{a}, \mathbf{b} \in T^L \setminus \{\mathbf{0}\}$ with $\mathbf{a} \neq \mathbf{b}$.

On the one hand, adding (18) and (19), we have

$$\mathsf{SNR}(\|\alpha\mathbf{h}-\mathbf{a}\|^2 + \|\alpha\mathbf{h}-\mathbf{b}\|^2) + 2|\alpha|^2 < \mathsf{SNR}/2^{R-1}. \tag{20}$$

On the other hand, we have

$$\begin{aligned} \mathsf{SNR}(\|\alpha\mathbf{h}-\mathbf{a}\|^2 + \|\alpha\mathbf{h}-\mathbf{b}\|^2) &\geq \mathsf{SNR}\|\mathbf{a}-\mathbf{b}\|^2/2 && (21)\\ &\geq \mathsf{SNR}/2, && (22) \end{aligned}$$

where (21) follows from the fact that $\|\alpha\mathbf{h}-\mathbf{a}\| + \|\alpha\mathbf{h}-\mathbf{b}\| \geq \|\mathbf{a}-\mathbf{b}\|$, and (22) follows from the fact that the norm of any nonzero element in $T$ must be no less than 1.

Combining (20) and (22), we have

$$\mathsf{SNR}/2 \leq \mathsf{SNR}(\|\alpha\mathbf{h}-\mathbf{a}\|^2 + \|\alpha\mathbf{h}-\mathbf{b}\|^2) < \mathsf{SNR}/2^{R-1}. \tag{23}$$

Therefore, we have $\mathsf{SNR}/2 < \mathsf{SNR}/2^{R-1}$, or equivalently, $R < 2$. In other words, if the message rate $R \geq 2$, there are no overlapping disks. $\square$

Fig. 3(a) and 3(b) show some typical good regions for asymptotically-good nested lattice codes with $T = \mathbb{Z}[i]$ and $T = \mathbb{Z}[\omega]$, respectively. The rotation angles in Fig. 3(a) and 3(b) are 90° and 60°, respectively, as explained in Proposition 2.

Fig. 4 depicts a typical good region for a simple nested lattice code $\mathcal{L}(\mathbb{Z}[i]^{400}, 2\mathbb{Z}[i]^{400}, \mathbf{d})$ with $\mathbf{d} = \frac{1}{2}(1+i, \ldots, 1+i)$, which is also known as uncoded 4-QAM with four constellation points $\left\{\frac{1}{2}(\pm 1 \pm i)\right\}$. Since this nested lattice code is not asymptotically good, the disjoint areas are not quite disk-like. That is, Proposition 3 approximately holds here. Nevertheless, the good region is still bounded and symmetric (with respect to rotations by 90°). That is, Propositions 1 and 2 still hold here.

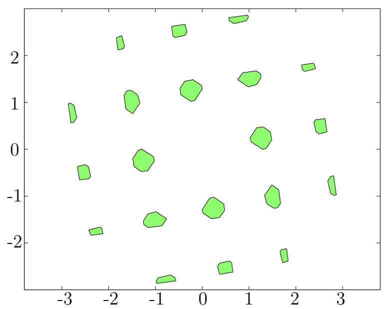



Fig. 4. A good region for the nested lattice code $\mathcal{L}(\mathbb{Z}[i]^{400}, 2\mathbb{Z}[i]^{400}, \mathbf{d})$, where $h_1 = 0.11 + 0.73i$, $h_2 = 0.78 + 0.19i$, and $\mathsf{SNR} = 35$dB.

### *B. The use of* $a_{\mathsf{sum}}\mathbf{d}$ *mod* $\Lambda'$ *in the decoding*

Here, we explain why knowledge of $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$ is sufficient for successful decoding. Recall that a good scalar $\alpha$ ensures successful decoding, i.e., $\varphi\left(\mathcal{Q}_{\Lambda}^{\text{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d})\right) = \sum_\ell a_\ell \mathbf{w}_\ell$ for some coefficients $(a_1, \ldots, a_L) \in T^L \setminus \{\mathbf{0}\}$. Note that if the term $a_{\mathsf{sum}}\mathbf{d}$ is replaced by $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$, the decoding is still successful. To see this, recall that $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda' = a_{\mathsf{sum}}\mathbf{d} - \boldsymbol{\lambda}'$ for some $\boldsymbol{\lambda}' \in \Lambda'$. Hence, we have

$$\begin{aligned} &\varphi\left(\mathcal{Q}_{\Lambda}^{\text{NN}}(\alpha\mathbf{y} - (a_{\mathsf{sum}}\mathbf{d} - \boldsymbol{\lambda}'))\right) \\ &= \varphi\left(\mathcal{Q}_{\Lambda}^{\text{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d}) + \boldsymbol{\lambda}'\right) \\ &= \varphi\left(\mathcal{Q}_{\Lambda}^{\text{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d})\right) + \varphi(\boldsymbol{\lambda}') \\ &= \sum_\ell a_\ell \mathbf{w}_\ell. \end{aligned} \tag{24}$$

Therefore, the receiver only needs to know $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$ after obtaining a good scalar $\alpha$.

We observe that $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$ often has a limited number of choices for commonly-used nested lattice codes, especially those obtained from linear codes [8]. For instance, we can construct a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ via an $[n, k]$ binary code $\mathcal{C}_{n,k}$ through lifted Construction A [8]. In particular, we have $\Lambda' = 2\mathbb{Z}[i]^n$, $\mathbf{d} = \frac{1}{2}(1+i, \ldots, 1+i)$, and $\boldsymbol{\lambda} \in \Lambda$ if and only $\text{Re}\{\boldsymbol{\lambda}\}$ mod 2, $\text{Im}\{\boldsymbol{\lambda}\}$ mod 2 are codewords in $\mathcal{C}_{n,k}$. Such a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ (known as coded 4-QAM) admits only eight possible choices of $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$, namely, $\{0\mathbf{d}, \pm\mathbf{d}, \pm i\mathbf{d}, , (1\pm i)\mathbf{d}, 2\mathbf{d}\}$. This greatly reduces the search space of $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$ for our blind C&F schemes.

### *C. Generic blind C&F scheme*

In previous sections, we have observed a number of properties of good scalars. We now present a generic blind C&F scheme based on these observations.

The first idea behind our generic scheme is to reduce the search space of good scalars as much as possible. Proposition 1 suggests that, to find a good scalar, it suffices to consider a bounded region. Proposition 2 shows that, to find a good scalar, it suffices to "ignore" some unnecessary areas. For instance, only the region in the first quadrant is worth investigating for nested lattice codes with $T = \mathbb{Z}[i]$. Proposition 3 implies that, to find a good scalar, it suffices to "probe" a discrete set of points. The denser this set, the better the performance.

The second idea is to use error detection to "probe" a given point, deciding whether this point is good or bad. Specifically, the transmitters embed a linear error-detecting code $\mathcal{C}$ into the message space $W$ so that each *valid* message $\mathbf{w}_\ell$ (as well as any linear combinations) is a codeword in $\mathcal{C}$. The receiver performs a basic probing operation as described in Algorithm 1.

**Algorithm 1** *Basic probing operation*

*Input:* a point $\alpha$.
*Output:* $\alpha$ is bad, or a good $\alpha$ with its associated $\hat{\mathbf{u}}$.

1. **for** each $\mathbf{t} \in \{a\mathbf{d}$ mod $\Lambda' : a \in T\}$ **do**
2. Compute $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_{\Lambda}^{\text{NN}}(\alpha\mathbf{y} - \mathbf{t})\right)$.
3. **if** $\hat{\mathbf{u}}$ is a non-zero codeword in $\mathcal{C}$ **then**
4. Declare $\alpha$ is good, output $\hat{\mathbf{u}}$, and then stop.
5. **end if**
6. **end for**
7. Declare $\alpha$ is bad.

If the point $\alpha$ is good, Algorithm 1 always declares $\alpha$ to be good and outputs some non-zero codeword $\hat{\mathbf{u}}$. If the point $\alpha$ is bad, Algorithm 1 might declare that $\alpha$ is good due to an undetected error. Such an error is called a Type-I error. The probability of a Type-I error can be made very small in practice by endowing $\mathcal{C}$ with sufficiently many parity checks.

When Algorithm 1 finds a good scalar $\alpha$, it does not necessarily mean that its output $\hat{\mathbf{u}} = \sum_\ell a_\ell \mathbf{w}_\ell$ for some $a_\ell$. This is because the associated $\mathbf{t}$ can be different from $a_{\mathsf{sum}}\mathbf{d}$ mod $\Lambda'$ due to an undetected error. We call such an error a Type-II error. As we will see in Sec. V, Type-II errors can be eliminated for certain nested lattice codes.

Now we are ready to describe a generic blind C&F scheme. The input to the scheme is an ordered list containing a discrete set of points. The scheme probes the points in the list one by one until it finds a good scalar or until it reaches the end of the list. The output is either a non-zero codeword $\hat{\mathbf{u}}$ (when the scheme finds a good scalar) or nothing (when the scheme finds no good scalars).

We note that the performance of the above generic scheme depends on the points in the list (but not on their probing order), whereas the computational cost of the scheme depends on the probing order of these points. In other words, two ordered lists containing exactly the same points achieve the same performance with possibly quite different computational complexity. We also note that the computational cost of the basic probing operation can be greatly reduced for some commonly-used nested lattice codes. All of these will be discussed in the next section.

## V. Blind Compute-and-Forward: Efficient Algorithms

In this section, we propose three (complementary) strategies to reduce the computational complexity of the generic blind C&F scheme presented in Sec. IV. The first strategy attempts to create some "smart" probing lists. The second strategy aims to detect bad scalars at a low cost. The third strategy further reduces the complexity of the basic probing operation.

### A. Hierarchically-organized list-building

The choice of the probing list is crucial to attaining good performance with low complexity. For instance, when the good region consists of many large disjoint areas, the probing points can be made relatively sparse. On the other hand, when the good region consists of a few small disjoint areas, the probing points should be made relatively dense. Based on this observation, we propose a heuristic method for creating the list.

First, we choose a well-shaped region $\mathcal{R}$ to avoid unnecessary probing (see discussions in Sec. IV-C). For nested lattice codes with $T = \mathbb{Z}[i]$, we note that $\mathcal{R}$ can be chosen heuristically as $[0, \log_{10}(\mathsf{SNR})/R] \times [0, \log_{10}(\mathsf{SNR})/R]$ (where $R$ is the message rate). For example, if $\mathsf{SNR} = 10\text{dB}$ and $R = 1$, then $\mathcal{R} = [0, 1] \times [0, 1]$.

Then, we construct an $m$-level lattice-partition chain [25] $\mathcal{L}_0/\mathcal{L}_1/\ldots/\mathcal{L}_m$ in $\mathbb{C}$ (i.e., each $\mathcal{L}_j$ is a one-dimensional complex lattice and $\mathcal{L}_0 \supset \mathcal{L}_1 \supset \cdots \supset \mathcal{L}_m$). Note that the lattice-partition chain, together with the region $\mathcal{R}$, induces $m+1$ probing grids $\{\mathcal{L}_j \cap \mathcal{R}\}$ satisfying $\{\mathcal{L}_m \cap \mathcal{R}\} \subset \cdots \subset \{\mathcal{L}_0 \cap \mathcal{R}\}$ (see Fig. 5 for a concrete example). For nested lattice codes with $T = \mathbb{Z}[i]$, we heuristically set $\mathcal{L}_j = \frac{1}{16}\log_{10}(\mathsf{SNR})(1+i)^j\mathbb{Z}[i]$, where $j = 0, \ldots, 8$.

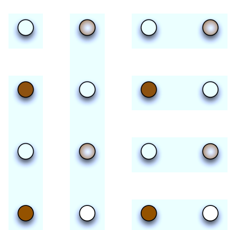


Fig. 5. An illustration of three (self-similar) probing grids. We choose $\mathcal{L}_j = (1+i)^j\mathbb{Z}[i]$ $(j = 0, 1, 2)$ and $\mathcal{R} = [0, 3] \times [0, 3]$. The sparsest grid consists of 4 solid points. The second sparsest grid consists of 4 solid points and 4 partially solid points.

With these grids, a list-building algorithm is described in Algorithm 2. The basic idea is to design a list such that the points in the sparser grids will appear before the points in the denser grids. That is, the points in $\mathcal{L}_j \cap \mathcal{R}$ are ordered to appear before the points in $\mathcal{L}_{j+1} \cap \mathcal{R}$.

**Algorithm 2** *Hierarchically-organized list-building algorithm*

*Input:* a lattice-partition chain $\mathcal{L}_0/\ldots/\mathcal{L}_m$ with a region $\mathcal{R}$.
*Output:* an ordered list of probing points.

```
Set list = ∅.
Set j = m and set L_{m+1} = {0}.
while j ≥ 0 do
    Let S = (L_j \ L_{j+1}) ∩ R.
    while |S| > 0 do
        Find a point α in S of the smallest L_1-norm.
        Set list = list ∪ {α}. Set S = S \ {α}.
    end while
    Set j = j − 1.
end while
```

### B. Quick detection for bad scalars

Note that the basic probing operation needs to compute $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda')\right)$ for each possible $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$. Such computations are often costly, due to the use of the nearest-neighbor-quantizer $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$. Here, we propose a new probing method that identifies certain bad scalars without the use of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$. Our new method only involves very simple computations, which is inspired by the *Smoothing Lemma* from lattice theory [26], [27].

*Definition 3 (Smoothing parameter):* For a complex lattice $\Lambda$ and for any $\epsilon > 0$, the *smoothing parameter* $\eta_\Lambda(\epsilon)$ is the smallest $\sigma > 0$ such that $\sum_{\boldsymbol{\lambda}^* \in \Lambda^* \setminus \{\mathbf{0}\}} e^{-\pi^2\sigma^2\|\boldsymbol{\lambda}^*\|^2} \leq \epsilon$, where $\Lambda^*$ is the dual lattice of $\Lambda$.

Clearly, $\eta_\Lambda(\epsilon)$ is a monotonically decreasing function of $\epsilon$. That is, for $\epsilon_1 < \epsilon_2$, we have $\eta_\Lambda(\epsilon_1) > \eta_\Lambda(\epsilon_2)$. The smoothing parameter bounds the variational distance between the Gaussian distribution mod $\Lambda$ and the uniform distribution $u_\Lambda$ on the Voronoi region $\mathcal{V}_\Lambda(\mathbf{0})$.

*Lemma 1 (Smoothing Lemma):* Let $\mathbf{n}$ be an i.i.d. circularly-symmetric complex Gaussian random vector with mean $\mu$ and variance $\sigma^2$, i.e., $\mathbf{n} \sim \mathcal{CN}(\mu\mathbf{1}, \sigma^2\mathbf{I}_{n\times n})$. Let $f_\Lambda(\cdot)$ be the probability density function of $\mathbf{n} \bmod \Lambda$. Then for any $\sigma > \eta_\Lambda(\epsilon)$, the variational distance between $f_\Lambda$ and $u_\Lambda$ is bounded by $\epsilon$, i.e.,

$$\int_{\mathcal{V}_\Lambda(\mathbf{0})} |f_\Lambda(\mathbf{t}) - u_\Lambda(\mathbf{t})| d\mathbf{t} \leq \epsilon. \tag{25}$$

The Smoothing Lemma says that $\mathbf{n} \bmod \Lambda$ tends to be uniform over the Voronoi region $\mathcal{V}_\Lambda(\mathbf{0})$ as $\sigma$ grows. This facilitates the detection of bad scalars. For any nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$, we have

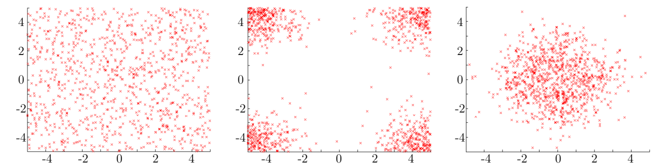
$$\begin{aligned}\alpha\mathbf{y} \bmod \Lambda &= \left(\sum_\ell a_\ell\tilde{\varphi}(\mathbf{w}_\ell) + \mathbf{n}_{\mathsf{eff}} + a_{\mathsf{sum}}\mathbf{d}\right) \bmod \Lambda \\ &= (\mathbf{n}_{\mathsf{eff}} + a_{\mathsf{sum}}\mathbf{d}) \bmod \Lambda. \end{aligned} \tag{26}$$

More generally, let $\Lambda_0$ be a lattice that contains $\Lambda$ (i.e., $\Lambda \subset \Lambda_0$), then we have

$$\alpha\mathbf{y} \bmod \Lambda_0 = (\mathbf{n}_{\mathsf{eff}} + a_{\mathsf{sum}}\mathbf{d}) \bmod \Lambda_0. \tag{27}$$

When the nested lattice code is asymptotically good (in the sense of [18]), we have $\mathbf{n}_{\mathsf{eff}} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_{n\times n})$, where $\sigma^2 = P\|\alpha\mathbf{h} - \mathbf{a}\|^2 + N_0|\alpha|^2$. Hence, by Lemma 1, $\alpha\mathbf{y} \bmod \Lambda_0$ tends to be uniform over $\mathcal{V}_{\Lambda_0}(\mathbf{0})$ as $\sigma$ becomes larger (or equivalently, as the scalar $\alpha$ becomes worse). Interestingly, this property still (approximately) holds for many commonly-used nested lattice codes, as suggested by our extensive numerical studies.

To illustrate this, we use a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ obtained via a $[1000, 500]$ binary LDPC code through lifted Construction A [8]. In particular, we have $\Lambda' = 2\mathbb{Z}[i]^{1000} \subset \Lambda \subset \mathbb{Z}[i]^{1000}$, $\mathbf{d} = \frac{1}{2}(1+i, \ldots, 1+i)$, and we choose $\Lambda_0 = \mathbb{Z}[i]^{1000}$. Fig. 6(a), 6(b), and 6(c) provide scatter-plots for $\alpha\mathbf{y} \bmod \Lambda_0$ with a bad scalar and two good scalars, respectively. Clearly, $\alpha\mathbf{y} \bmod \Lambda_0$ is close to uniform for a bad scalar, and is highly non-uniform for good scalars. In particular, $\alpha\mathbf{y} \bmod \Lambda_0$ distributes around four corners for a good scalar with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{d}$ (see Fig. 6(b)), and is centered at




(a) (b) (c)

Fig. 6. Scatter-plots for $\alpha\mathbf{y} \bmod \Lambda_0$ with $h_1 = -1.17 + 1.40i$, $h_2 = -0.01 - 0.71i$, and $\mathsf{SNR} = 16$dB: (a) a bad scalar $\alpha = 1.98 + 1.01i$; (b) a good scalar $\alpha = -0.12 + 1.52i$ with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{d}$; (c) a good scalar $\alpha = 1.21 - 0.24i$ with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{0}$.

the origin for a good scalar with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{0}$ (see Fig. 6(c)). It is easy to check that $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0$ takes values in $\{\mathbf{0}, \mathbf{d}\}$. So, we have two cases here in total.

Let $\mathbf{u}_{\Lambda_0}$ be a random vector uniform over the Voronoi region $\mathcal{V}_{\Lambda_0}(\mathbf{0})$. Clearly, $\mathbf{u}_{\Lambda_0}$ consists of i.i.d. random variables with variance $1/6$. On the other hand, the sample variances of $\alpha\mathbf{y} \bmod \Lambda_0$ in Fig. 6(a), 6(b), and 6(c) are $0.165$, $0.322$, and $0.051$, respectively. As expected, the sample variance of $\alpha\mathbf{y} \bmod \Lambda_0$ is close to $1/6 \approx 0.167$ for some bad scalars, and is away from $1/6$ for good scalars. This example confirms our previous observations. More importantly, it suggests a quick detection algorithm to identify some bad scalars. For ease of presentation, Algorithm 3 assumes that $T = \mathbb{Z}[i]$ and $\Lambda_0 = \mathbb{Z}[i]^n$ (which can be extended as we will soon see).

**Algorithm 3** *Quick detection for bad scalars when* $T = \mathbb{Z}[i]$ *and* $\Lambda_0 = \mathbb{Z}[i]^n$

*Input:* a point $\alpha$, a threshold $\delta$.
*Output:* $\alpha$ is bad, or a good $\alpha$ with its associated $\hat{\mathbf{u}}$.

```
Compute v = αy mod Λ0.
Compute the sample mean v̄ = (1/n) Σ_i v_i.
Compute the sample variance s^2 = (1/(n−1)) Σ_i |v_i − v̄|^2.
if s^2 ∈ [1/6 − δ, 1/6 + δ] then
    Declare α is bad.
else
    Perform the basic probing operation.
end if
```

Note that when $\Lambda_0 = \mathbb{Z}[i]^n$, the cost of computing $\alpha\mathbf{y} \bmod \Lambda_0$ is very low. In this case, $\alpha\mathbf{y} \bmod \Lambda_0 = \alpha\mathbf{y} - \text{round}(\alpha\mathbf{y})$, where $\text{round}(\cdot)$ is the standard rounding operation. Note also that the complexity of Algorithm 3 can be further reduced by operating on a subset of $\alpha\mathbf{y}$. For instance, in our previous numerical example, if we only operate on the first 100 elements of $\alpha\mathbf{y}$, then the new sample variances are $0.166$, $0.328$, and $0.054$, respectively, which are quite close to the original sample variances. Hence, it suffices to consider only a subset of $\alpha\mathbf{y}$ in practice. Therefore, the complexity of Algorithm 3 can be made very low. Also, note that Algorithm 3 can be easily extended to other cases, such as $T = \mathbb{Z}[\omega]$ and $\Lambda_0 = \mathbb{Z}[\omega]^n$. Clearly, its complexity remains to be low as long as $\Lambda_0$ is simple enough.

### C. Fast probing operation

Next, we present a fast probing method that reduces the use of $\mathcal{Q}_\Lambda^{\text{NN}}(\cdot)$ in the basic probing operation. Our method requires the underlying nested lattice code to satisfy some mild conditions. For ease of presentation, we focus on a case study in which the conditions are $T = \mathbb{Z}[i]$ and $(1+i)\mathbf{d} \in \Lambda$.

**Algorithm 4** *Fast probing when* $T = \mathbb{Z}[i]$ *and* $(1+i)\mathbf{d} \in \Lambda$

*Input:* a point $\alpha$.
*Output:* $\alpha$ is bad, or a good $\alpha$ with its associated $\hat{\mathbf{u}}$.

```
Compute û1 = φ(Q_Λ^NN(αy)).
for each t ∈ {b(1+i)d mod Λ' : b ∈ Z[i]} do
    if û1 − φ(t) is a non-zero codeword in C then
        Declare α is good, output û = û1 − φ(t), and then stop.
    end if
end for
Compute û2 = φ(Q_Λ^NN(αy − d)).
for each t ∈ {b(1+i)d mod Λ' : b ∈ Z[i]} do
    if û2 − φ(t) is a non-zero codeword in C then
        Declare α is good, output û = û2 − φ(t), and then stop.
    end if
end for
Declare α is bad.
```

Our fast probing operation presented in Algorithm 4 requires at most two uses of $\mathcal{Q}_\Lambda^{\text{NN}}(\cdot)$, while still achieving the same performance as the basic probing operation, as shown in the following theorem.

*Theorem 1:* When $T = \mathbb{Z}[i]$ and $(1+i)\mathbf{d} \in \Lambda$, Algorithm 4 is equivalent to Algorithm 1.

*Proof:* We only need to show that for each computation $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_\Lambda^{\text{NN}}(\alpha\mathbf{y} - \mathbf{t})\right)$ in Algorithm 1, there is a corresponding computation in Algorithm 4. Suppose that $\mathbf{t} = a\mathbf{d} \bmod \Lambda'$ for some $a \in \mathbb{Z}[i]$. Then $\mathbf{t} = a\mathbf{d} - \boldsymbol{\lambda}'$ for some $\boldsymbol{\lambda}' \in \Lambda'$. Note that every $a \in \mathbb{Z}[i]$ can be expressed as $a = b(1+i) + c$, where $b \in \mathbb{Z}[i]$ and $c \in \{0, 1\}$. (This is a natural generalization of the binary expansion.) Hence, we have $\mathbf{t} = b(1+i)\mathbf{d} + c\mathbf{d} - \boldsymbol{\lambda}'$ for some $b \in \mathbb{Z}[i]$ and $c \in \{0,1\}$. Therefore,

$$\begin{aligned}
\hat{\mathbf{u}} &= \varphi\left(\mathcal{Q}_\Lambda^{\text{NN}}(\alpha\mathbf{y} - c\mathbf{d} - b(1+i)\mathbf{d} + \boldsymbol{\lambda}')\right) \\
&= \varphi\left(\mathcal{Q}_\Lambda^{\text{NN}}(\alpha\mathbf{y} - c\mathbf{d}) - b(1+i)\mathbf{d} + \boldsymbol{\lambda}'\right) \\
&= \varphi\left(\mathcal{Q}_\Lambda^{\text{NN}}(\alpha\mathbf{y} - c\mathbf{d})\right) - \varphi(b(1+i)\mathbf{d}) \\
&= \varphi\left(\mathcal{Q}_\Lambda^{\text{NN}}(\alpha\mathbf{y} - c\mathbf{d})\right) - \varphi\left(b(1+i)\mathbf{d} \bmod \Lambda'\right),
\end{aligned}$$

which is indeed a computation in Algorithm 4. □

We note that $(1+i)\mathbf{d} \in \Lambda$ is a mild constraint for certain nested lattice codes. For example, for a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ obtained via an $[n,k]$ binary code $\mathcal{C}_{n,k}$ through lifted Construction A, $(1+i)\mathbf{d} \in \Lambda$ simply means that the binary code $\mathcal{C}_{n,k}$ contains the all-ones codeword, which implies that each parity-check equation for $\mathcal{C}_{n,k}$ involves an even number of bits. When $\mathcal{C}_{n,k}$ is an LDPC code, this means that all the check degrees are even, which can be easily satisfied in practice.

### D. Combining our strategies together

Now, we are ready to combine the quick detection strategy and fast probing strategy together. We note that such a combination has several unique advantages for nested lattice codes obtained via $[n,k]$ binary codes through lifted Construction A. For this family of nested lattice codes, $\alpha\mathbf{y} \bmod \mathbb{Z}[i]^n$ reveals whether $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n = \mathbf{0}$ or $\mathbf{d}$ for good scalars, as discussed in Sec. V-B. Note that $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n = \mathbf{0}$ means that $a_{\mathsf{sum}} = b(1+i)$ for some $b \in \mathbb{Z}[i]$. In this case, it suffices to compute $\hat{\mathbf{u}}_1 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y})\right)$. Similarly, $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n = \mathbf{d}$ means that $a_{\mathsf{sum}} = b(1+i)+1$ for some $b \in \mathbb{Z}[i]$, and it suffices to compute $\hat{\mathbf{u}}_2 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$. Hence, only one use of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$ is needed with the help of $\alpha\mathbf{y} \bmod \mathbb{Z}[i]^n$. This leads to a faster probing method presented in Algorithm 5. Moreover, Algorithm 5 has another advantage: Type-II errors can be completely eliminated if the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit.

**Algorithm 5** *Faster probing when $T = \mathbb{Z}[i]$, $(1+i)\mathbf{d} \in \Lambda$, and $\Lambda_0 = \mathbb{Z}[i]^n$*

*Input:* a point $\alpha$, a threshold $\delta$.
*Output:* $\alpha$ is bad, or a good $\alpha$ with its associated $\hat{\mathbf{u}}$.

Compute $\mathbf{v} = \alpha\mathbf{y} \bmod \mathbb{Z}[i]^n$.
Compute the sample mean $\bar{v} = \frac{1}{n}\sum_i v_i$.
Compute the sample variance $s^2 = \frac{1}{n-1}\sum_i |v_i - \bar{v}|^2$.
**if** $s^2 \in [1/6 - \delta,\ 1/6 + \delta]$ **then**
  Declare $\alpha$ is bad and stop.
**else if** $s^2 < 1/6 - \delta$ **then**
  Compute $\hat{\mathbf{u}}_0 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y})\right)$.
**else if** $s^2 > 1/6 + \delta$ **then**
  Compute $\hat{\mathbf{u}}_0 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$.
**end if**
**for** each $\mathbf{t} \in \{b(1+i)\mathbf{d} \bmod \Lambda' : b \in \mathbb{Z}[i]\}$ **do**
  **if** $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t})$ is a non-zero codeword in $\mathcal{C}$ **then**
    Declare $\alpha$ is good, output $\hat{\mathbf{u}} = \hat{\mathbf{u}}_0 - \varphi(\mathbf{t})$, and then stop.
  **end if**
**end for**
Declare $\alpha$ is bad.

*Theorem 2:* Suppose that $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n$ is revealed correctly. Then Type-II errors cannot occur as long as the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit.

*Proof:* We assume, without loss of generality, that $a_{\mathsf{sum}} = b(1+i)+1$ for some $b \in \mathbb{Z}[i]$. In this case, we have $\hat{\mathbf{u}}_0 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$, since $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n$ is revealed correctly. To prove that there is no Type-II error, we only need to show that there is a unique $\mathbf{t} \in \{b(1+i)\mathbf{d} \bmod \Lambda' : b \in \mathbb{Z}[i]\}$ such that $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t})$ is a non-zero codeword in $\mathcal{C}$. Suppose that there exist $\mathbf{t}_1$ and $\mathbf{t}_2$ such that both $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t}_1)$ and $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t}_2)$ are non-zero codewords in $\mathcal{C}$. Then, $\varphi(\mathbf{t}_1 - \mathbf{t}_2) = \varphi(\mathbf{t}_1) - \varphi(\mathbf{t}_2)$ must be a codeword in $\mathcal{C}$. In particular, it means that the syndrome of $\varphi(\mathbf{t}_1 - \mathbf{t}_2)$ is the zero vector. Recall that the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit. Hence, the syndrome of $\varphi(\mathbf{t}_1 - \mathbf{t}_2)$ is the zero vector if and only if $\mathbf{t}_1 = \mathbf{t}_2$. This proves the uniqueness. □

It is very easy to ensure that the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit. This is because even if the error-detecting code $\mathcal{C}$ does not satisfy this condition automatically, one can always add an extra parity-check in $\mathcal{C}$ to enforce this condition.

# VI. Simulation Results

In this section, we illustrate the feasibility and scalability of our blind C&F schemes through simulations. The nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ is constructed from a $[1000, 500]$ binary LDPC code with column weight 3 and row weight 6 following lifted Construction A in [8]. In particular, we have $\Lambda' = 2\mathbb{Z}[i]^{1000} \subset \Lambda \subset \mathbb{Z}[i]^{1000}$ (and the message space is over $\mathbb{Z}[i]/\langle 2\rangle$). The linear error-detecting code $\mathcal{C}$ is based on the standard CRC-32 code (lifted from $\mathbb{Z}_2$ to $\mathbb{Z}[i]/\langle 2\rangle$). Hence, the overhead of our blind schemes is is $32/500 = 6.4\%$. The region $\mathcal{R}$ is set to $[0, \log_{10}(\mathsf{SNR})/R] \times [0, \log_{10}(\mathsf{SNR})/R]$, and the lattice-partition chain is $\mathcal{L}_j = \frac{1}{16}\log_{10}(\mathsf{SNR})(1+i)^j\mathbb{Z}[i]$ ($j = 0, \ldots, 8$) as suggested in Sec. V-A. The threshold is set to $\delta = 0.0175$.

## A. Feasibility of Blind C&F

To demonstrate the feasibility, we consider a simple two-transmitter, single receiver configuration. Communication occurs in rounds. In each round, the channel gains are assumed to follow independent Rayleigh fading. A round is said to be successful if the receiver correctly recovers a linear combination. The throughput is defined as the fraction of successful rounds in the simulation (i.e., the throughput equals to one minus the outage probability).

We have evaluated four blind C&F schemes through simulation by carrying out $10,000$ rounds. These four schemes apply the list-building algorithm in Sec. V-A and the probing strategies presented in Algorithms 1, 3, 4, and 5, respectively. Note that the throughputs of these schemes are the same, since they use the same probing list. Table I compares the throughput of these blind schemes with that of coherent C&F under various SNRs. It is observed that these schemes are able to approach the throughput of coherent C&F. In addition, we also have evaluated the performance of estimation-based C&F schemes with $\mathsf{SNR}_{\mathsf{est}}$ set to $\mathsf{SNR}$, $\mathsf{SNR}+\ 3$ dB, and $\mathsf{SNR}+\ 6$ dB. We note that even if we set $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+\ 6$ dB, our blind schemes still outperform the estimation-based scheme.

TABLE I
Throughput (%) of various C&F schemes.

| $\mathsf{SNR}$ | 6 dB | 8 dB | 10 dB | 12 dB | 14 dB |
|---|---|---|---|---|---|
| Coherent | 78.35 | 87.03 | 93.17 | 97.16 | 98.87 |
| Blind | 73.23 | 84.35 | 92.21 | 96.91 | 98.63 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}$ | 37.46 | 48.86 | 61.96 | 74.03 | 82.41 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+\ 3$ dB | 48.99 | 61.62 | 75.24 | 84.31 | 91.58 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+\ 6$ dB | 58.30 | 72.99 | 83.83 | 91.01 | 95.13 |

We next examine the complexity of our blind C&F schemes under various SNRs. Recall that the complexity of our blind schemes is dominated by the use of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$. As such, the complexity is measured by the number of uses of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$. Table II compares the complexity of these blind schemes under various SNRs. It is observed that our proposed strategies, especially the quick detection strategy, are effective in controlling the complexity. Recall that, in order to identify a bad scalar, Algorithm 1 requires eight uses of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$, whereas Algorithm 4 only requires two uses of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$. Hence, the complexity of Algorithm 1 is roughly four times the complexity of Algorithm 4. Similarly, the complexity of Algorithm 3 is roughly eight times that of Algorithm 5.

TABLE II
COMPLEXITY OF FOUR BLIND C&F SCHEMES.

| SNR | 6 dB | 8 dB | 10 dB | 12 dB | 14 dB |
|---|---|---|---|---|---|
| Algorithm 1 | 557.61 | 348.95 | 202.44 | 113.57 | 67.68 |
| Algorithm 3 | 51.68 | 28.77 | 18.06 | 12.46 | 9.33 |
| Algorithm 4 | 139.92 | 87.75 | 51.09 | 28.86 | 17.40 |
| Algorithm 5 | 6.83 | 4.01 | 2.69 | 2.01 | 1.63 |

To summarize, for the two-transmitter, single receiver setup, our blind C&F scheme with Algorithm 5 is able to approach the throughput of coherent C&F with just twice the complexity in the high-throughput region. Interestingly, this conclusion still holds even if we increase the number of transmitters, as we will see shortly.

### B. Scalability of blind C&F

To demonstrate the scalability, we increase the number of transmitters in our simulation, while keeping all other setups exactly the same as before. In particular, this means that the overhead of our blind C&F schemes is independent of the number of transmitters. By contrast, for any given $\mathsf{SNR}_{\mathsf{est}}$, the overhead of estimation-based schemes increases linearly with the number of transmitters.

TABLE III
THROUGHPUT (%) OF VARIOUS C&F SCHEMES WITH THREE USERS.

| SNR | 8 dB | 10 dB | 12 dB | 14 dB | 16 dB |
|---|---|---|---|---|---|
| Coherent | 71.89 | 84.80 | 93.57 | 97.77 | 99.32 |
| Blind | 69.11 | 82.49 | 92.27 | 97.26 | 99.24 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}$ | 28.89 | 39.08 | 48.94 | 59.91 | 68.95 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+$ 3 dB | 40.99 | 53.43 | 65.37 | 74.96 | 84.19 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+$ 6 dB | 52.15 | 64.80 | 77.63 | 86.61 | 92.80 |

TABLE IV
COMPLEXITY OF FOUR BLIND C&F SCHEMES WITH THREE USERS.

| SNR | 8 dB | 10 dB | 12 dB | 14 dB | 16 dB |
|---|---|---|---|---|---|
| Algorithm 1 | 752.71 | 512.60 | 319.52 | 196.83 | 128.24 |
| Algorithm 3 | 24.39 | 16.87 | 12.39 | 9.60 | 7.98 |
| Algorithm 4 | 188.59 | 128.60 | 80.35 | 49.68 | 32.51 |
| Algorithm 5 | 3.39 | 2.51 | 1.99 | 1.66 | 1.46 |

Tables III and V compare the throughput of our blind schemes with that of coherent C&F as well as estimation-based C&F schemes. It is observed that our blind schemes consistently approach the throughput of coherent C&F, especially in the high-throughput region. However, the throughput of estimation-based schemes degrades significantly as the number of transmitters increases. For example, in the case of four transmitters, our blind scheme achieves 25% higher throughout than the estimation-based scheme with $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+$ 6 dB.

Tables IV and VI compare the complexity of our blind schemes under various SNRs for the cases of three and four transmitters, respectively. Once again, the complexity of our blind scheme with Algorithm 5 is just twice of coherent C&F in the high-throughput region. This demonstrates the scalability of our blind schemes.

TABLE V
THROUGHPUT (%) OF VARIOUS C&F SCHEMES WITH FOUR USERS.

| SNR | 10 dB | 12 dB | 14 dB | 16 dB | 18 dB |
|---|---|---|---|---|---|
| Coherent | 56.20 | 72.28 | 86.44 | 95.57 | 96.51 |
| Blind | 52.48 | 67.66 | 83.44 | 93.78 | 95.42 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}$ | 21.06 | 27.95 | 35.33 | 43.75 | 44.37 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+$ 3 dB | 29.43 | 39.35 | 49.77 | 59.79 | 65.85 |
| $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}+$ 6 dB | 38.18 | 50.78 | 61.97 | 73.98 | 76.53 |

TABLE VI
COMPLEXITY OF FOUR BLIND C&F SCHEMES WITH FOUR USERS.

| SNR | 10 dB | 12 dB | 14 dB | 16 dB | 18 dB |
|---|---|---|---|---|---|
| Algorithm 1 | 1091.83 | 843.79 | 563.40 | 356.82 | 325.68 |
| Algorithm 3 | 28.50 | 21.49 | 15.66 | 11.77 | 10.74 |
| Algorithm 4 | 273.27 | 211.30 | 141.27 | 89.65 | 81.96 |
| Algorithm 5 | 3.82 | 3.01 | 2.35 | 1.91 | 1.73 |

To sum up, our blind C&F schemes scale well with the number of transmitters with respect to overhead, throughput, and complexity, whereas estimation-based schemes suffer from degraded performance in terms of overhead and throughput. This trend is already clear when the number of transmitters is as small as four.

### C. Performance of blind C&F in multi-source multi-relay networks

Finally, we evaluate the performance of our blind C&F schemes in a two-source two-relay network as shown in Fig. 1, with a comparison to the coherent scheme described in Sec. II-C as well as the estimation-based scheme described in Sec. III. This helps us to understand the loss due to the lack of knowledge of CSI at the relays.

At first, it looks like that—without CSI—the $m$th relay using our blind C&F schemes cannot acquire the coefficient vector $\mathbf{a}_m$ even if it decodes a linear combination correctly. For example, suppose that the message space $W$ is over $\mathbb{Z}_5$, and the messages at Source 1 and Source 2 are

$$\mathbf{w}_1 = (2,1,4,3,0,1) \text{ and } \mathbf{w}_2 = (3,2,2,4,1,2),$$

respectively. Now, suppose that Relay 1 is able to recover

$$\mathbf{u}_1 = \mathbf{w}_1 + 2\mathbf{w}_2 = (3,0,3,1,2,0)$$

via our blind decoding schemes. Although Relay 1 knows $\mathbf{u}_1$, it cannot infer the coefficient vector $\mathbf{a}_1 = (1,2)$.

To solve this problem, one can apply the header technique developed in [28], which is widely used in practical (digital) network-coding schemes. Specifically, the first two symbols of the messages are used as headers as follows:

$$\mathbf{w}_1 = (1,0,4,3,0,1) \text{ and } \mathbf{w}_2 = (0,1,2,4,1,2).$$

In this case, the linear combination $\mathbf{u}_1$ becomes

$$\mathbf{u}_1 = \mathbf{w}_1 + 2\mathbf{w}_2 = (1,2,3,1,2,0),$$

from which Relay 1 can infer that $\mathbf{a}_1 = (1,2)$ by checking the first two symbols of $\mathbf{u}_1$. In general, when there are $L$ sources, the header for the $\ell$th message has the following format:

$$(\underbrace{0,\ldots,0}_{\ell-1},1,\underbrace{0,\ldots,0}_{L-\ell}),$$

which allows a relay to infer a coefficient vector by checking the first $L$ symbols of its recovered linear combination.

This header technique incurs an overhead of $L$ symbols, which is negligible when the message length (i.e., the number of symbols in a message) is much larger than $L$ [28]. This header technique can be adapted to any type of nested lattice codes, as explained in [29], [30]. In addition, this header technique supports non-symmetric rates. This is because the use of non-symmetric rates leads to zero-padded messages of equal length [1], for which the first $L$ symbols can still be used as headers.

Now, we are ready to present two methods of applying our blind C&F schemes to a two-source two-relay network.

1) each relay tries to decode a linear combination using our blind C&F schemes and then forwards its linear combination to the destination.
2) each relay tries to decode two (linearly independent) linear combinations and forwards their coefficient vectors to the destination, which then decides which linear combinations to request from the relays.

Clearly, Method 1 has lower complexity than Method 2. On the other hand, Method 2 enjoys higher throughput than Method 1, as illustrated in the following example.

Suppose that the message space is over $\mathbb{Z}_5$. Suppose that, with blind C&F decoding, Relay 1 can recover two linear combinations $\mathbf{u}_1 = \mathbf{w}_1 + 2\mathbf{w}_2$ and $\mathbf{u}_1' = 2\mathbf{w}_1 + \mathbf{w}_2$, and Relay 2 can recover only one linear combination $\mathbf{u}_2 = 3\mathbf{w}_1 + \mathbf{w}_2$. If Method 1 is employed, as soon as Relay 1 recovers $\mathbf{u}_1$, it forwards $\mathbf{u}_1$ to the destination (and doesn't bother to recover $\mathbf{u}_1'$). Similarly, once Relay 2 recovers $\mathbf{u}_2$, it forwards $\mathbf{u}_2$ to the destination. In this case, the destination obtains $\mathbf{u}_1$ and $\mathbf{u}_2$, which are, however, linearly dependent, since $\mathbf{u}_2 = 3\mathbf{u}_1$ (over $\mathbb{Z}_5$). As such, the destination declares an outage event. On the other hand, if Method 2 is employed, Relay 1 recovers both $\mathbf{u}_1$ and $\mathbf{u}_1'$ and forwards their coefficient vectors $\mathbf{a}_1 = (1, 2)$ and $\mathbf{a}_1' = (2, 1)$ to the destination. Similarly, Relay 2 recovers $\mathbf{u}_2$ and forwards its coefficient vector $\mathbf{a}_2 = (3, 1)$. The destination receives three coefficient vectors $\mathbf{a}_1$, $\mathbf{a}_1'$ and $\mathbf{a}_2$, and will decide to request $\mathbf{u}_1'$ and $\mathbf{u}_2$ from Relay 1 and Relay 2, respectively (because $\mathbf{a}_1'$ and $\mathbf{a}_2$ are linearly independent). In this case, the destination can reconstruct the original messages $\mathbf{w}_1$ and $\mathbf{w}_2$ by solving a system of linear equations.

In fact, the destination in Method 2 can recover the original messages if one of the following two conditions is met:

1) One relay successfully decodes two (linearly independent) linear combinations, and the other relay successfully decodes at least one linear combination.
2) Each relay successfully decodes exactly one linear combination, and these two combinations are linearly independent.

Otherwise, the destination declares an outage event.

Table VII compares the throughput of our two methods with that of coherent C&F and estimation-based C&F. As we can see, Method 2 clearly outperforms estimation-based schemes, and is able to approach the throughput of coherent C&F. This is quite interesting, because Method 2 does not require any CSI, whereas coherent C&F requires either full CSI at the destination or additional signaling overhead among the relays.

TABLE VII
THROUGHPUT (%) OF VARIOUS C&F SCHEMES FOR A TWO-SOURCE TWO-RELAY NETWORK.

| SNR | 6 dB | 8 dB | 10 dB | 12 dB |
|---|---|---|---|---|
| Coherent | 61.24 | 72.63 | 85.67 | 92.52 |
| $\mathsf{SNR}_{\text{est}} = \mathsf{SNR} + 6$ dB | 39.32 | 53.35 | 67.92 | 79.91 |
| Method 1 | 30.93 | 38.14 | 40.36 | 46.19 |
| Method 2 | 56.58 | 71.03 | 85.21 | 92.43 |

On the other hand, the complexity of Method 2 is roughly twice of that of Method 1, as shown in Table VIII. This is expected, because each relay in Method 2 decodes two linear combinations instead of one. This leads to an interesting performance/complexity tradeoff.

Finally, for a general multi-source multi-relay network with $L$ sources and $L$ relays, we can define $L$ methods based on the number of linear combinations needed to decode at each relay. On one extreme, each relay attempts to decode only one linear combination; at the other extreme, each relay tries to decode $L$ linear combinations. This enables us to achieve a performance/complexity tradeoff for a more general setup, which we leave for our future work.

TABLE VIII
COMPLEXITY OF BLIND C&F IN A TWO-SOURCE TWO-RELAY NETWORK.

| SNR | 6 dB | 8 dB | 10 dB | 12 dB |
|---|---|---|---|---|
| Method 1 | 6.83 | 4.01 | 2.69 | 2.01 |
| Method 2 | 11.47 | 7.13 | 4.47 | 3.90 |

## VII. CONCLUSION

In this paper, the problem of designing blind C&F schemes has been considered. A framework based on error-detection and the Smoothing Lemma has been proposed, which eliminates the need for CSI in C&F. In particular, a generic blind C&F scheme has been developed, and several strategies have been suggested to make it computationally efficient. The effectiveness of our approach has been demonstrated through simulations. The simulation results show that our proposed blind C&F schemes can approach the throughput of coherent C&F with a modest increase in computational complexity.

We believe that there is still much work to be done in this direction, including investigating the effect of the threshold $\delta$ as well as devising more efficient probing lists based on the properties of good scalars.

## ACKNOWLEDGMENT

The authors would like to thank Bobak Nazer and Michael Gastpar for suggesting the blind compute-and-forward problem.

## REFERENCES

[1] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.
[2] U. Niesen and P. Whiting, “The degrees of freedom of compute-and-forward,” *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5214–5232, Aug. 2012.

[3] U. Niesen, B. Nazer, and P. Whiting, "Computation alignment: Capacity approximation without noise accumulation," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3811–3832, Jun. 2013.

[4] J. Zhan, B. Nazer, M. Gastpar, and U. Erez, "MIMO compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Seoul, South Korea, Jun. 28– Jul. 3, 2009, pp. 2848–2852.

[5] S.-N. Hong and G. Caire, "Compute-and-forward strategy for cooperative distributed antenna systems," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 5227–5243, Sep. 2013.

[6] O. Ordentlich, U. Erez, and B. Nazer, "The approximate sum capacity of the symmetric Gaussian $k$-user interference channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3450–3482, Jun. 2014.

[7] J. Zhu and M. Gastpar, "Lattice codes for many-to-one interference channels with and without cognitive messages," *IEEE Trans. Inf. Theory*, vol. 61, no. 3, pp. 1309–1324, Mar. 2015.

[8] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.

[9] B. Hern and K. Narayanan, "Multilevel coding schemes for compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Saint Petersburg, Russia, Jul. 31– Aug. 5, 2011, pp. 1713–1717.

[10] O. Ordentlich, J. Zhan, U. Erez, M. Gastpar, and B. Nazer, "Practical code design for compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Saint Petersburg, Russia, Jul. 31– Aug. 5, 2011, pp. 1876–1880.

[11] J.-C. Belfiore, "Lattice codes for the compute-and-forward protocol: The flatness factor," in *IEEE Inf. Workshop*, Paraty, Brazil, Oct. 16–20, 2011, pp. 1876–1880.

[12] N. E. Tunali, K. R. Narayanan, J. J. Boutros, and Y.-C. Huang, "Lattices over Eisenstein integers for compute-and-forward," in *Proc. 2012 Allerton Conf. Commun., Control, and Comput.*, Monticello, IL, Oct. 2012, pp. 33–40.

[13] Q. Sun and J. Yuan, "Lattice network codes based on Eisenstein integers," in *Proc. 20012 IEEE Int. Conf. on Wireless and Mobile Comput.*, Barcelona, Spain, Oct. 2012, pp. 225–231.

[14] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proc. IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011.

[15] K. N. Pappi, G. K. Karagiannidis, and R. Schober, "How sensitive is compute-and-forward to channel estimation errors," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Istanbul, Turkey, Jul. 7–12, 2013, pp. 3110–3114.

[16] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.

[17] C. Feng, D. Silva, and F. R. Kschischang, "Blind compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Cambridge, MA, Jul. 1–6, 2012, pp. 408–412.

[18] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[19] A. Mejri and G. R.-B. Othman, "Practical physical layer network coding in multi-sources relay channels via the compute-and-forward," in *Proc. of WCNC*, Shanghai, China, Apr. 7–10, 2013, pp. 166–171.

[20] M. E. Soussi, A. Zaidi, and L. Vandendorpe, "Compute-and-forward on a multiaccess relay channel: Coding and symmetric-rate optimization," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 1932–1947, Apr. 2014.

[21] L. Wei and W. Chen, "Compute-and-forward network coding design over multi-source multi-relay channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3348–3357, Sep. 2012.

[22] Z. Chen, P. Fan, and K. B. Letaief, "Compute-and-forward: Optimization over multi-source-multi-relay networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1806–1818, May 2015.

[23] M. E. Soussi, A. Zaidi, and L. Vandendorpe, "Compute-and-forward on a multi-user multi-relay channel," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 589–592, Dec. 2014.

[24] K. N. Pappi, P. D. Diamantoulakis, H. Otrok, and G. K. Karagiannidis, "Cloud compute-and-forward with relay cooperation," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3415–3428, Jun. 2015.

[25] G. D. Forney, Jr., "Coset codes—part II: Binary lattices and related codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1152–1187, Sep. 1988.

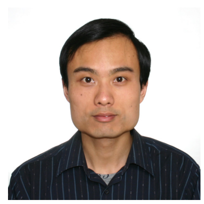
[26] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM J. on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[27] C. Ling, L. Luzzi, J.-C. Belfiore, and D. Stehle, "Semantically secure lattice codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6399–6416, Oct. 2014.

[28] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003, pp. 40–49.

[29] C. Feng, R. W. Nóbrega, F. R. Kschischang, and D. Silva, "Communication over finite-chain-ring matrix channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5899–5917, Oct. 2014.

[30] R. W. Nóbrega, C. Feng, D. Silva, and B. F. Uchôa-Filho, "On multiplicative matrix channels over finite chain rings," in *Proc. of IEEE Int. Symp. on Network Coding*, Calgary, Canada, Jun. 7–9, 2013, (see also Arxiv: http://arxiv.org/abs/1311.4861).


**Chen Feng** (S'08–M'14) received the B.Eng. degree from the Department of Electronic and Communications Engineering, Shanghai Jiao Tong University, China, in 2006, and the M.A.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Toronto, Canada, in 2009 and 2014, respectively.

From 2014 to 2015, he was a Postdoctoral Fellow with Boston University, USA, and the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. He joined the School of Engineering, University of British Columbia, Kelowna, Canada, in July 2015, where he is currently an Assistant Professor. His research interests include data networks, coding theory, information theory, and network coding.

Dr. Feng was a recipient of the prestigious NSERC Postdoctoral Fellowship in 2014. He was recognized by the IEEE TRANSACTIONS ON COMMUNICATIONS (TCOM) as an Exemplary Reviewer in 2015. He is a member of ACM and IEEE.



**Danilo Silva** (S'06–M'09) received the B.Sc. degree from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 2002, the M.Sc. degree from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil, in 2005, and the Ph.D. degree from the University of Toronto, Toronto, Canada, in 2009, all in electrical engineering.

From 2009 to 2010, he was a Postdoctoral Fellow at the University of Toronto, at the École Polytechnique Fédérale de Lausanne (EPFL), and at the State University of Campinas (UNICAMP). In 2010, he joined the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Brazil, where he is currently an Assistant Professor. His research interests include wireless communications, channel coding, information theory, and network coding.

Dr. Silva was a recipient of a CAPES Ph.D. Scholarship in 2005, the Shahid U. H. Qureshi Memorial Scholarship in 2009, and a FAPESP Postdoctoral Scholarship in 2010.

**Frank R. Kschischang** (S'83–M'91–SM'00–F'06) received the B.A.Sc. degree (with honors) from the University of British Columbia, Vancouver, BC, Canada, in 1985 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1988 and 1991, respectively, all in electrical engineering. He is a Professor of Electrical and Computer Engineering at the University of Toronto, where he has been a faculty member since 1991. During 1997-98, he was a visiting scientist at MIT, Cambridge, MA; in 2005 he was a visiting professor at the ETH, Zurich, and in 2011 and again in 2012-13 he was a visiting Hans Fischer Senior Fellow at the Institute for Advanced Study at the Technische Universität München.

His research interests are focused primarily on the area of channel coding techniques, applied to wireline, wireless and optical communication systems and networks. In 1999 he was a recipient of the Ontario Premiers Excellence Research Award and in 2001 (renewed in 2008) he was awarded the Tier I Canada Research Chair in Communication Algorithms at the University of Toronto. In 2010 he was awarded the Killam Research Fellowship by the Canada Council for the Arts. Jointly with Ralf Kötter he received the 2010 Communications Society and Information Theory Society Joint Paper Award. He is a recipient of the 2012 Canadian Award in Telecommunications Research. He is a Fellow of IEEE, of the Engineering Institute of Canada, and of the Royal Society of Canada.

During 1997-2000, he served as an Associate Editor for Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY, and since January 2014, he serves as this journals Editor-in-Chief. He also served as technical program co-chair for the 2004 IEEE International Symposium on Information Theory (ISIT), Chicago, and as general co-chair for ISIT 2008, Toronto. He served as the 2010 President of the IEEE Information Theory Society.