Serial Quantization for Sparse Time Sequences

Alejandro Cohen, *Member, IEEE*, Nir Shlezinger, *Member, IEEE*, Salman Salamatian, Yonina C. Eldar, *Fellow, IEEE*, and Muriel Médard, *Fellow, IEEE*

***Abstract*—Sparse signals are encountered in a broad range of applications. In order to process these signals using digital hardware, they must first be sampled and quantized using an analog-to-digital convertor (ADC), which typically operates in a serial scalar manner. In this work, we propose an serial quantization of sparse time sequences (SQuaTS) method inspired by group testing theory. This method is designed to reliably and accurately quantize sparse signals acquired in a sequential manner using the serial scalar ADCs. Unlike previously proposed approaches that combine quantization and compressed sensing (CS), our SQuaTS scheme updates its representation on each incoming analog sample and does not require the complete signal to be observed or stored in analog prior to quantization. We characterize the asymptotic tradeoff between the accuracy and quantization rate of SQuaTS as well as its computational burden. We also propose a variation of SQuaTS that trades the quantization rate for computational efficiency. Next, we show how SQuaTS can be naturally extended to distributed quantization scenarios, where a set of jointly sparse time sequences are acquired individually and processed jointly. Our numerical results demonstrate that SQuaTS is capable of achieving substantially improved representation accuracy over previous CS-based schemes without requiring the complete set of analog signal samples to be observed prior to signal quantization, making this method an attractive approach for acquiring sparse time sequences.**

***Index Terms*—Quantization, distributed quantization, sparsity, group testing.**

I. Introduction

QUANTIZATION allows for continuous-amplitude physical signals to be represented using discrete values and processed in digital hardware. Such continuous-to-discrete conversions play an important role in digital signal processing systems [1]. In theory, jointly mapping a set of samples via vector quantization yields the most accurate digital representation [2, Ch. 10]. However, as such joint mappings are difficult to implement, quantization is most commonly performed using ADCs, which operate in a serial and scalar manner; specifically, the analog signal is sampled and each incoming sample is sequentially mapped into a discrete representation using the same mapping [3]. Since ADCs operating at high frequencies are costly in terms of memory and power usage, it is often desirable to utilize low quantization rates, i.e., assign a limited number of bits per each input sample, inducing an additional quantization error which degrades the digital representation accuracy [4, Ch. 23].

The quantization error encountered under bit budget constraints can be mitigated by accounting for the underlying structure or the system task. Such quantization systems are the focus of several recent works. For example, scalar quantization mappings designed to maximize the mutual information and Fisher information with respect to a statistically related quantity were studied in [5] and [6], respectively. The [7] study showed that a quantization system using a uniform ADCs can approach an achievable performance using vector quantizers when the system task is not to recover the analog signal but rather to estimate some lower-dimensional information embedded into the signal. This approach was extended to massive multiple-input multiple-output (MIMO) channel estimation with quantized outputs in [8], as well as to the recovery of quadratic functions in [9], while a data-driven implementation was proposed in [10]. The systems proposed in [7]–[10] used hybrid architectures, allowing some constrained processing to be performed in analog prior to quantization in order to mitigate the error induced by bit-limited serial scalar ADCs.

Conventional quantization theory considers the acquisition of a discrete-time analog source into a digital form [1]. In some practical applications, such as sensor networks, multiple signals are acquired in distinct physical locations, while their digital representation is utilized in some central processing device, resulting in a distributed quantization setup. The recovery of a single parameter from the acquired signals was considered in [11], [12], and its extension to the recovery of a common source, known as the CEO problem, was studied in [13], [14] (see also [15, Ch. 12]). Joint recovery of sources acquired in a distributed manner was studied in [16], which focused on sampling, while [17], [18] proposed non-uniform quantization mappings for the representation of multiple sources. Multivariate (vector) quantizers for arbitrary networks were then considered in [19].

A common structure exhibited by physical signals is sparsity. Sparse signals are frequently encountered in various applications, ranging from biomedical and optical imaging [20], [21] to radar [22] and communications [23], [24]. An important property of sparse signals is the fact that they can be perfectly reconstructed from a lower-dimensional projection without knowledge of the sparsity pattern. This property was studied within the framework of CS [25], [26], which considered the recovery of sparse signals from their lower dimensional projections.

Manuscript received January 31, 2020; revised November 20, 2020 and May 14, 2021; accepted May 20, 2021. Date of publication May 27, 2021; date of current version June 15, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weiyu Xu. This work was supported in part by the Benoziyo Endowment Fund for the Advancement of Science, the Estate of Olga Klein - Astrachan, the European Unions Horizon 2020 research and innovation Program under Grant 646804-ERC-COG-BNYQ, and in part by the Israel Science Foundation under Grant 0100101. Parts of this work were presented at the Allerton Annual Conference on Communications, Control, and Computing, July 2019. *(Corresponding author: Alejandro Cohen.)*

Alejandro Cohen, Salman Salamatian, and Muriel Médard are with the Research Laboratory of Electronics, MIT, Cambridge, MA 02139 USA (e-mail: cohenale@mit.edu; salmansa@mit.edu; medard@mit.edu).

Nir Shlezinger is with the School of Electrical Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel (e-mail: nirshl@bgu.ac.il).

Yonina C. Eldar is with the Math, and CS, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

Digital Object Identifier 10.1109/TSP.2021.3083985

Recovery of sparse signals from quantized measurements is the focus of a large body of work [27]–[33]. The most common approach studied in the literature is to first project the signal in the analog domain and then quantize the compressed measurements through a one-bit representation [27], [28], uniform quantization [29], sigma-delta quantization [30], [33], or vector source coding [31]. A detailed survey and analysis of methods combining quantization and CS can be found in [32]. In the context of distributed systems, CS for multiple signals acquired separately was studied in [34]–[38], while [39], [40] proposed vector quantization schemes for bit-constrained distributed CS. Despite their similarities, the distributed quantization of sparse signals and distributed CS with quantized observations have a fundamental difference: in the quantization framework, the measurements are the sparse signals, while in CS, the observations are a linear projection of the signals. Consequently, in order to utilize the CS methods, one must first have access to the complete signal in order to project it and then quantize, imposing a major drawback when acquiring time sequences. On the other hand, a sequential approach allows for quantizing without requiring that the entirety of the signal be accessible, which is particularly relevant not only in the distributed scenarios but also for signals sparse in time, since storing the entire time-signal in analog form is expensive. Furthermore, while CS algorithms have been proven to achieve asymptotic recovery guarantees, their performance may be degraded in finite signal sizes. These drawbacks give rise to the need for a reliable and sequential method for quantizing and recovering sparse signals, which is the focus of this work.

Here, we propose SQuaTS, which is a method for quantizing and recovering discrete-time sparse time sequences that utilize standard serial scalar ADC quantizers.[1] Our scheme is inspired by recent developments in group testing theory and leverages coding principles initially designed for secure group testing [41] in order to facilitate recovery of the time sequence. In particular, SQuaTS first quantizes each sample using a scalar ADC, and uses the ADC output to update a single binary value, which in turn is used as a codeword from which the sequence can be accurately recovered with a high probability. The resulting coding scheme, which quantizes the sparse signal directly and operates over the binary field, allows improved reconstruction compared to CS-based methods, which process a quantized linear projection of the real-valued observations while also avoiding the need to store samples in analog by sequentially updating a single register.

We characterize the achievable accuracy of SQuaTS in the asymptotically large signal size regime, showing that any fixed desirable distortion level can be achieved with an overall number of bits that grows logarithmically with the signal dimensionality and linearly with the support size. We then characterize the computational complexity of SQuaTS and propose a reduced complexity scheme for implementing SQuaTS at the cost of a degraded representation accuracy.

Next, we discuss how SQuaTS can be naturally applied for distributed quantization of a set of temporally jointly sparse time sequences. We begin with the case where each acquired signal is conveyed to the central unit through a direct link, representing, e.g., single-hop networks. Then we show how the technique can be extended to multi-hop networks, in which the quantized data must travel over multiple intermediate links to reach the central server, and formulate simplified network policies, dictating the behavior of each intermediate node. We characterize the achievable distortion of SQuaTS when applied in a distributed setup, and derive conditions on the system parameters under which it can achieve the same distortion as when applied in a non-distributed case, assuming that at least a single path to the central unit exists.

[1]While ADC traditionally refers to hardware that samples and quantizes an analog signal, here we denote ADC as only the quantization end of this hardware, i.e., we assume throughout the paper a discrete time input signal.

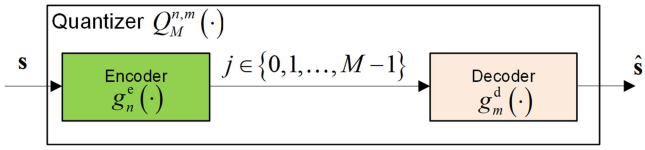


Fig. 1. Quantization system.

Our numerical results demonstrate that both SQuaTS and its reduced complexity variation substantially outperform the conventional approach of combining CS and quantization when applied to the digital representation of a single sparse time-sequence, as well as in distributed acquisition scenarios with jointly sparse signals. This demonstrates the potential of SQuaTS for feasible and reliable quantization of sequentially acquired sparse time sequences.

The remainder of this paper is organized as follows: In Section II we review some preliminaries in quantization theory and present the system model. Section III proposes SQuaTS along with a discussion and an asymptotic performance analysis. Section IV presents a reduced complexity variation of SQuaTS. In Section V we apply SQuaTS for distributed quantization. Section VI details the simulation study, and Section VII provides the concluding remarks. Proofs of the results stated in the paper are detailed in the appendix.

Throughout the paper, we use boldface lower-case letters for vectors, e.g., $\boldsymbol{x}$. Matrices are denoted with boldface upper-case letters, e.g., $\boldsymbol{M}$. Sets are expressed with calligraphic letters, e.g., $\mathcal{X}$, and $\mathcal{X}^n$ is the nth order Cartesian power of $\mathcal{X}$. The stochastic expectation is denoted by $\mathbb{E}\{\cdot\}$, $\bigvee$ is the Boolean OR operation, and $\mathcal{R}$ is the set of real numbers. All logarithms are taken to base-2.

II. Preliminaries and System Model

A. Preliminaries in Quantization Theory

To formulate the quantization of sparse signals setup, we first briefly review standard quantization notions. We begin with the definition of a quantizer:

Definition 1 (Quantizer): A quantizer $Q_M^{n,m}(\cdot)$ with $\log M$ bits, an input size of n, an input alphabet of $\mathcal{S}$, an output size of m, and an output alphabet of $\hat{\mathcal{S}}$ consists of the following: *1)* An encoding function $g_n^{\mathrm{e}} : \mathcal{S}^n \mapsto \{0, 1, \ldots, M-1\} \triangleq \mathcal{M}$ that maps the input from $\mathcal{S}^n$ into a discrete index $j \in \mathcal{M}$. *2)* A decoding function $g_m^{\mathrm{d}} : \mathcal{M} \mapsto \hat{\mathcal{S}}^m$ that maps each index $j \in \mathcal{M}$ into a codeword $\boldsymbol{q}_j \in \hat{\mathcal{S}}^m$.

The quantizer output for an input $\boldsymbol{s} \in \mathcal{S}^n$ is $\hat{\boldsymbol{s}} = g_m^{\mathrm{d}}(g_n^{\mathrm{e}}(\boldsymbol{s})) \triangleq Q_M^{n,m}(\boldsymbol{s})$. An illustration is depicted in Fig. 1. *Scalar quantizers* have scalar input and output, i.e., $n = m = 1$ and $\mathcal{S}$ is a scalar space, while *vector quantizers* operate a multivariate input. When the input size and output size are equal, namely, $n = m$,

we write $Q_M^n(\cdot) \triangleq Q_M^{n,n}(\cdot)$, while for scalar quantizers we use the notation $Q_M(\cdot) \triangleq Q_M^1(\cdot)$.

In the conventional quantization problem, a $Q_M^n(\cdot)$ quantizer is designed to minimize some of the distortion measure $d_n : \mathcal{S}^n \times \hat{\mathcal{S}}^n \mapsto \mathcal{R}^+$ between its input and its output. The performance of a quantizer is therefore characterized using two measures: the quantization rate, defined as $R \triangleq \frac{1}{n}\log M$, and the expected distortion $\mathbb{E}\{d_n(\boldsymbol{s}, \hat{\boldsymbol{s}})\}$. For a fixed input size n and codebook size M, the optimal quantizer is given by

$$Q_M^{n,\text{opt}}(\cdot) = \min_{Q_M^n(\cdot)} \mathbb{E}\left\{d_n\left(\boldsymbol{s}, Q_M^n(\boldsymbol{s})\right)\right\}. \tag{1}$$

In the following, the distortion between a source realization $\boldsymbol{s}$ and a reconstruction sequence $\hat{\boldsymbol{s}}$ is defined as the mean-squared error (MSE) of their difference, which is given by

$$d_n(\boldsymbol{s}, \hat{\boldsymbol{s}}) \triangleq \frac{1}{n}||\boldsymbol{s} - \hat{\boldsymbol{s}}||^2. \tag{2}$$

Characterizing the optimal quantizer via (1) and the distortion via (2), as well as the optimal tradeoff between the distortion and quantization rate, is in general a difficult task. Consequently, optimal quantizers are typically studied assuming either a high quantization rate, i.e., $R \to \infty$, see, e.g., [42], or an asymptotically large input size, namely, $n \to \infty$, typically with stationary inputs via rate-distortion theory [2, Ch. 10].

Comparing a high rate analysis for scalar quantizers and rate-distortion theory for vector quantizers demonstrates the suboptimality of serial scalar quantization. For example, for quantizing a large-scale real-valued Gaussian random vector with i.i.d. entries and a sufficiently large quantization rate R, where intuitively there is little benefit in quantizing the entries jointly over quantizing each entry independently, vector quantization notably outperforms serial scalar quantization [4, Ch. 23.2]. Nonetheless, vector quantizers are significantly more complex compared to scalar quantizers. One of the main sources for this increased complexity stems from the fact that vector quantizers operate on a set of analog samples. As a result, a digital signal processor (DSP) utilizing vector quantizers to acquire an analog time sequence must store n samples in the analog domain before it produces a digital representation, which may be difficult to implement, especially for a large n. Scalar quantizers, commonly used in ADC devices [43], do not store data in analog, since each incoming sample is immediately converted into a digital representation, which is the focus here.

B. System Model

We consider the acquisition of a sampled time sequence $s[i]$ observed over the period $i \in \{1, \ldots, T\} \triangleq \mathcal{T}$ into a digital representation $\hat{s}[i]$ using up to b bits, i.e., $M = 2^b$ codewords. The signal $s[i]$ is temporally sparse with a support size $k < T$, where k is a-priori known.[2] We propose a quantization system that is specifically designed to exploit this sparsity in order to improve the recovery accuracy. In particular, we propose an encoder-decoder pair that utilizes tools from group testing theory to exploit the underlying sparsity of the continuous amplitude signal.

[2]While we conduct our derivations and analysis assuming k is known, we only require an upper bound on it. In fact, the SQuaTS method supports operations with erroneous knowledge of k, as discussed in Section III-E.

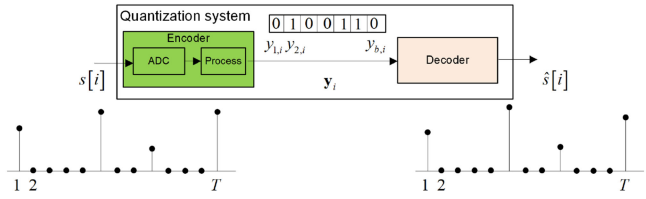


Fig. 2. Serial quantization system for sparse time sequences.

In order to avoid the need to store samples in analog, the system operates at each sample $s[i]$ independently. In particular, on each time instance $i \in \mathcal{T}$, the encoder updates a register of b bits, whose value upon the encoding of $s[i]$ is denoted by $\boldsymbol{y}_i$. Once the complete time sequence is observed, i.e., $i = T$, the decoder uses the digital codeword $\boldsymbol{y}_T$ to produce an estimate of the sequence denoted $\{\hat{s}[i]\}_{i\in\mathcal{T}}$. An illustration of the system is depicted in Fig. 2. Since the decoder processes the discrete codeword $\boldsymbol{y}_T$, while each $\boldsymbol{y}_i$, $i < T$ is stored only during the ith acquisition step, the system uses b bits for the digital representation.

III. The SQuaTS System

We next detail the proposed SQuaTS system. The main rationale of SQuaTS is to facilitate quantization of sparse signals using conventional low-complexity serial scalar quantizers by utilizing group testing theory tools. Broadly speaking, SQuaTS quantizes each incoming sample using a scalar ADC. However, instead of storing this quantized value, it is used to update a b bits codeword, which is decoded into a digital representation of the sparse signal. This approach allows each incoming sample to be quantized with a relatively high resolution, while using a single b bits register from which the digital representation of the complete signal is obtained.

To properly formulate SQuaTS, we first present the codebook generation in Subsection III-A. Then we elaborate on the SQuaTS encoder and decoder structures in Subsection III-B and III-C, respectively. In Subsection III-D we characterize the achievable distortion of SQuaTS in the large signal size regime. Finally, in Subsection III-E we discuss the pros and cons of SQuaTS compared to previously proposed approaches for quantizing sparse signals.

A. Codebook Generation

The SQuaTS system maintains a codebook used by its encoder and decoder. In particular, for a time sequence of T samples, SQuaTS uses a codebook of $l \cdot T + 1$ codewords, each consisting of b bits, where l is a fixed integer. We discuss the effect of l on the MSE as well as the complexity of SQuaTS in Subsection III-D, and propose guidelines for setting its value to optimize the tradeoff between these performance measures.

Our codebook design is based on the wireless sensor coding scheme of [44], which is inspired by recent advances in group testing theory [45], and particularly the code proposed in [41] for secure group testing. The objective of group testing is to identify a subset of defective items in a larger set using as few measurements as possible. This objective can be recast as a codebook generation problem, such that for each outcome vector, i.e., a set of measurements, it should be possible to identify the non-zero inputs [45]. While this setup bears much similarity to our quantization of sparse sources problem, in group testing the

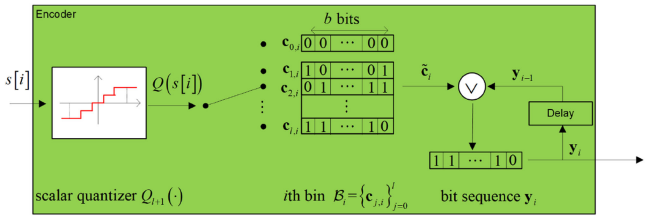


Fig. 3. Encoding process of the SQuaTS system.

inputs are represented over a binary field, while in our setting the inputs can be any real value. Consequently, the codebook here needs to be able to not only detect the indexes of the non-zero inputs, as in conventional group testing, but also recover their value. To facilitate our design, we henceforth assume that the inputs are discretized to a set of $l+1$ different values and show how this is incorporated into the overall scheme in the following subsections. We refer to Section VI for a discussion on how the parameter l relates to the overall quantization rate.

In particular, to formulate the codebook we generate $T \cdot l \cdot b$ independent realizations from a Bernoulli distribution with a mean value $\frac{\ln(2)}{k}$. These realizations form $l \cdot T$ mutually independent codewords. The codewords are then divided into T bins, denoted $\mathcal{B}_i \triangleq \{\boldsymbol{c}_{j,i}\}_{j=1}^{l}$, $i \in \mathcal{T}$, and we add the all-zero codeword to each bin, denoted as $\boldsymbol{c}_0$. Since $\boldsymbol{c}_0$ is common to all the bins, the total number of codewords is $l \cdot T + 1$. The benefits of this codebook design are discussed in Subsection III-E.

B. Encoder Structure

Having generated T bins of l codewords, $\{\mathcal{B}_i\}_{i=1}^{T}$, we now discuss the encoding process. To that end, we fix some scalar quantization mapping $Q_{l+1}(\cdot)$ over $\mathcal{R}$ with a resolution of $l+1$, denoted henceforth as $Q(\cdot)$ for simplicity, and let $\{q_j\}_{j=0}^{l}$ be the set of its possible outputs. The specific selection of the quantization mapping represents the acquisition hardware. For example, when using the common flash ADC architecture, $Q(\cdot)$ represents a uniform quantization mapping with $l+1$ uniformly spaced decision regions [3]. Without a loss of generality, we assume that the scalar quantizer maps the input value 0 into the discrete value q_0, namely, $Q(0) = q_0$.

The encoding process consists of the following three stages, illustrated in Fig. 3:

- *E1* Each incoming sample $s[i]$ is then quantized into the discrete scalar value $Q(s[i])$. Since this same identical mapping is applied to each incoming sample in a serial manner, it can be implemented using conventional ADCs.
- *E2* The encoder uses the index of the discrete value $Q(s[i])$ to select a codeword from the ith bin as follows: If $q_j = Q(s[i])$, then the selected codeword is $\tilde{\boldsymbol{c}}_i = \boldsymbol{c}_{j,i} \in \mathcal{B}_i$.
- *E3* The encoder output $\boldsymbol{y}_i$, which is initialized such that $\boldsymbol{y}_0$ is the all-zero vector, is updated by taking its Boolean OR with the selected codeword $\tilde{\boldsymbol{c}}_i$, i.e.,

$$\boldsymbol{y}_i = \boldsymbol{y}_{i-1} \bigvee \tilde{\boldsymbol{c}}_i. \tag{3}$$

Consequently, the encoder output $\boldsymbol{y}_T$ is given by

$$\boldsymbol{y}_T = \bigvee_{i=1}^{T} \tilde{\boldsymbol{c}}_i. \tag{4}$$

Note that only the discrete index of the quantized $Q(s[i])$, not its actual value, affects the selection of the encoder output $\boldsymbol{y}_T$. Nonetheless, in Subsection III-D we show that the selection of the output of $Q(\cdot)$, i.e., the values of $\{q_j\}$, not only its partition of $\mathcal{R}$ into decision regions, affect the overall MSE of SQuaTS. Additionally, the formulation of $\boldsymbol{y}_T$ via (3) implies that it can be represented using a single register of b bits, which is updated using logical operations on each incoming sample. Consequently, while the encoder assigns a b bits codeword to each incoming sample, the overall output size is b and not $T \cdot b$, so the quantization rate is $R = \frac{b}{T}$.

C. Decoder Structure

The recovery of the digital representation $\{\hat{s}[i]\}_{i\in\mathcal{T}}$ from the output of the encoder $\boldsymbol{y}_n \in \{0,1\}^b$ is based on maximum likelihood (ML) decoding. In this decoding scheme, the most likely set of k codewords are selected, from which the digital representation is obtained. To formulate the decoding process, recall that the set $\mathcal{T}$ has exactly $\binom{T}{k}$ possible subsets of size k, thus representing the possible sets of non-zero entries of $\boldsymbol{s}$. We use $\{\mathcal{X}_w\}_{w\in\{1,\ldots,\binom{T}{k}\}}$ to denote these subsets. The SQuaTS decoder implements the following steps:

- *D1* For a given encoder output $\boldsymbol{y}_T$, the decoder recovers a collection of k codewords $\hat{\boldsymbol{C}}_{\mathcal{X}_w} = \{\boldsymbol{c}_{j_i,i}\}_{i\in\mathcal{X}_w}$, *each one taken from a separate bin*, for which $\boldsymbol{y}_T$ is most likely, namely,

$$\Pr\left(\boldsymbol{y}_T \middle| \hat{\boldsymbol{C}}_{\mathcal{X}_w}\right) \geq \Pr\left(\boldsymbol{y}_T \middle| \hat{\boldsymbol{C}}_{\mathcal{X}_{\tilde{w}}}\right), \quad \forall \tilde{w} \neq w. \tag{5}$$

  The decoder looks for both the set of k bins $\mathcal{X}_w$ as well as the selection of the codeword for each bin, i.e., the selection of codeword index j_i within the ith bin, $i \in \mathcal{X}_w$, which maximizes the conditional probability (5).
- *D2* The decoder recovers $\{\hat{s}[i]\}$ from $\hat{\boldsymbol{C}}_{\mathcal{X}_w} = \{\boldsymbol{c}_{j_i,i}\}_{i\in\mathcal{X}_w}$ by setting its ith entry, denoted as $\hat{s}[i]$, to be $\hat{s}[i] = q_{j_i}$ for each $i \in \mathcal{X}_w$ and $\hat{s}[i] = q_0$ for $i \notin \mathcal{X}_w$.

The ML decoder scans $\binom{T}{k}(l)^k$, the possible subsets of codewords in the codebook, i.e., the $\binom{T}{k}$ possible bins corresponding to indexes which may contain non-zero values, and the l codewords in each such bin. For every scanned subset of codewords, the decoder compares the Boolean OR of each subset which contains k codewords to the quantized register $\boldsymbol{y}_T$. Since the length of each codeword is b, the computational complexity is on the order of $\mathcal{O}(\binom{T}{k} l^k k b)$ operations.

While the decoding process described above may be computationally complex, it essentially implements a one-to-one mapping from $\boldsymbol{y}_T$ to $\{\hat{s}[i]\}$, and can thus be implemented using a standard look-up table. In Section IV, we present a sub-optimal low-complexity SQuaTS decoder.

In the following subsection we study the achievable performance, in terms of the tradeoff between the quantization rate and distortion, of the proposed SQuaTS system.

D. Achievable Performance

In order to study the achievable performance, we first note that the SQuaTS encoder and decoder are designed to recover the output of the scalar quantizer $Q(\cdot) = Q_{l+1}(\cdot)$. Therefore, when the SQuaTS decoder detects the correct set of codewords, the

distortion is determined by the scalar quantizer and its resolution, which is dictated by the parameter l.

To formulate this distortion, define the overall average MSE of the scalar quantizer via

$$D_T(l) \triangleq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[|s[i]-Q_{l+1}(s[i])|^2\right]. \tag{6}$$

The average MSE (6) is then determined by the scalar quantization mapping $Q(\cdot)$ and the distribution of the time sequence $\{s[i]\}$. This represents the accuracy of applying $Q(\cdot)$ directly to the sequence $\{s[i]\}$ without using any additional processing, thus operating at a quantization rate of $\log(l+1)$ bits per input sample. SQuaTS with rate R, which as we show next can be much smaller than $\log(l+1)$, is capable of achieving this average MSE when its decoder successfully recovers the correct set of codewords. A sufficient condition for successful recovery in the limit of asymptotically large inputs, and thus for (6) to be achievable, is stated in the following theorem:

Theorem 1: The SQuaTS system applied to a sparse signal $\{s[i]\}_{i\in\mathcal{T}}$ with support size $k=\mathcal{O}(1)$ achieves the average MSE $D_T(l)$ given in (6) in the limit $T\to\infty$ when for some $\varepsilon>0$, the quantization rate R satisfies:

$$R \geq R_\varepsilon(l) \triangleq \max_{1\leq i\leq k}\frac{(1+\varepsilon)k}{i\cdot T}\log\left(\binom{T-k}{i}\cdot l^i\right). \tag{7}$$

Proof: The proof is given in the appendix.

Theorem 1 implies that, as T increases, if the number of bits is $b=R\cdot T$ where R satisfies (7), then the average error probability in detecting the SQuaTS codewords approaches zero (decaying exponentially with T) and thus the SQuaTS system achieves the average MSE $D_T(l)$ given in (6).

Note that the average MSE $D_T(l)$ and the corresponding quantization rate $R_\varepsilon(l)$ both depend on the auxiliary parameter l. The dependence of $D_T(l)$ on l is obtained from the quantization mapping used, as well as the distribution of the input $\{s[i]\}$. For example, when the samples of $\{s[i]\}$ are identically distributed with probability density function (PDF) $f_s(\cdot)$, then, using the Panter-Dite approximation [46], the optimal (non-uniform) scalar quantizer in the fine quantization regime achieves the following average MSE:

$$D_T(l) \approx \frac{1}{12}2^{-2\log(l+1)}\left(\int_{\alpha\in\mathcal{R}} f_s^{1/3}(\alpha)d\alpha\right)^3. \tag{8}$$

The average MSE in (8) implies that the achievable distortion using scalar quantizers, including such conventional architectures as uniform quantization mappings, can be made arbitrarily small by increasing the resolution $\log(l+1)$.

While the average MSE directly depends on the quantization mapping, the rate $R_\varepsilon(l)$ is invariant to the setting of $Q(\cdot)$, and is obtained as the maximal value of the right hand side of (7). To avoid the need to search for the maximal value in (7), we state an upper bound on $R_\varepsilon(l)$ in the following corollary:

Corollary 1: The quantization rate $R_\varepsilon(l)$ in (7) satisfies

$$R_\varepsilon(l) \leq (1+\varepsilon)\frac{k}{T}\log\left(T\cdot l\right). \tag{9}$$

Proof: The corollary is obtained by substituting in (7) the upper bound $\log\binom{T-k}{i}\leq i\log T$, which follows from Stirling's approximation [2], [47]. ■

We note that when $k=\mathcal{O}(1)$, the upper bound (9) tends to zero as T grows for any fixed l. Consequently, for a large T, SQuaTS requires significantly smaller quantization rates to achieve $D_T(l)$ compared to directly applying $Q(\cdot)$ to $\{s[i]\}$, which requires a rate of $\log(l+1)$ in order to achieve the same average MSE. This gain, which demonstrates the ability of SQuaTS to exploit the underlying sparsity of $\{s[i]\}$, is also observed in the simulations study presented in Section VI.

Corollary 1 can be used to determine the quantization rate for achieving a desirable MSE for a given family of scalar quantization mappings: The parameter l is then set to the minimal value, for which $D_T(l)$ is not larger than the desirable distortion. Next, using the resulting l, the quantization rate can be obtained using the right-hand side of (9). Theorem 1 guarantees that, for a large input size T, the desirable distortion is achievable when using SQuaTS with the selected quantization rate. In fact, in the numerical study presented in Section VI we demonstrate that by properly tuning l, the proposed system can achieve substantial MSE gains over previously proposed approaches for quantizing sparse time sequences.

The bound on the quantization rate required to approach $D_T(l)$ given in Corollary 1 can also be used to characterize the asymptotic growth rate of the number of quantization bits used by the SQuaTS system, b, as stated in the following corollary:

Corollary 2: The MSE $D_T(l)$ can be approached as T increases when the number of quantization bits b grows as

$$b=\mathcal{O}\left(k\log T+k\log l\right). \tag{10}$$

Corollary 2 implies that, besides the obvious linear dependence in $k\log l$, the required number of bits grows proportionally to a logarithmic factor of T, which depends on the sparsity pattern size k. A similar asymptotic growth in the number of bits, i.e., proportional to $k\log T$, was also shown to be sufficient to achieve a given distortion when using the CS-based methods in [27, Thm. 2]. However, our numerical study presented in Section VI demonstrates that, despite the similarity in the asymptotic growth, when b is fixed SQuaTS achieves an improved reconstruction accuracy compared to CS-based techniques.

Substituting (10) in the ML decoding complexity in Subsection III-C allows us to characterize the computational burden of the ML decoder, as stated in the following corollary:

Corollary 3: SQuaTS with the ML decoder detailed in Subsection III-C is capable of achieving the MSE $D_T(l)$ (6) in the limit of $T\to\infty$ with a computational complexity on the order of $\mathcal{O}(\binom{T}{k}l^k k^2\log T+\binom{T}{k}l^k k^2\log l)$ operations.

The complexity of the SQuaTS decoder is significantly more affected by the size of the sparsity pattern k than by the signal size T and the resolution of the scalar quantizer l. While this implies that the SQuaTS system is the most computationally efficient for highly sparse inputs, the proposed mechanism is applicable for any size of the sparsity pattern.

We note that SQuaTS is geared towards low rate and low resolution scenarios, where one typically has much to gain by incorporating coding schemes in quantization over merely utilizing serial scalar ADCs. As shown in (10), the codeword length scales logarithmically with l and T. If the codewords are too long due to design issues, e.g., high values of l yielding fine resolution quantization are desired, fragmentation as proposed in [44] can be used. That is, in the SQuaTS system, fragmentation is performed by dividing the set of input bits T or setting the levels l into two or more groups. This operation

reduces the memory size and the decoding complexity as needed. Furthermore, when the sparsity level is approximately identical among the groups, as is the case for large groups with i.i.d. inputs or in the presence of prior knowledge of structured sparsity, fragmentation does not compromise the performance of the proposed SQuaTS system, as we numerically demonstrate in Fig. 10 and Fig. 14 in Section VI.

E. Discussion

We next discuss the practical aspects of this method and its rationale. In particular, we first detail the benefits which stem from the SQuaTS architecture and compare it to related schemes for quantizing sparse signals, such as direct application of scalar quantizers as well as compress-and-quantize [27]–[30], [33]. Then we elaborate on the relationship between SQuaTS and the group testing theory.

1) Practical Benefits and Comparison With Related Schemes: SQuaTS is specifically designed to utilize scalar ADCs in a serial manner. The resulting structure can therefore be naturally implemented using practical ADC architectures [3]. Moreover, SQuaTS is tailored to exploit an underlying sparsity of the input signal. A straight-forward application of a serial scalar ADC requires $T \cdot \log(l+1)$ bits to achieve the distortion $D_T(l)$ in (6). Our proposed SQuaTS, which exploits the sparsity of the input by further encoding the ADC output in a serial manner, requires $b = \mathcal{O}(k \log(Tl))$ bits to achieve the same MSE, as follows from Corollary 2. This implies that for highly sparse signals, i.e., when $k \ll T$, SQuaTS significantly reduces the number of bits while utilizing the scalar ADCs for acquisition by introducing an additional encoding applied in a serial manner at its output. The resulting approach thus bears some similarity to previously proposed universal quantization methods that are based on applying entropy coding to the output of a quantizer. See [48] for scalar quantizers and [49] for vector quantizers. Indeed, since the codewords representing the quantized value are generated according to a Bernoulli distribution with mean value of $\frac{\ln(2)}{k}$ and the outcome $\boldsymbol{y}_T$ is the Boolean OR of k inputs, it can be shown that its entries approach being independent and equally distributed on the set $\{0,1\}$ for large values of k, namely, the optimal lossless encoded representation, as achieved using entropy coding [2, Ch. 5]. Nonetheless, to apply conventional entropy coding, one must first quantize all the entries of the input (or at least a large block of input entries) before applying the encoding process, requiring a large number of bits to store and represent this quantized block. SQuaTS, which is specifically designed to operate in a serial manner, updates the same b-bits register on each incoming sample, thus avoiding the need to store the output of the serial scalar ADC $Q(\cdot)$ prior to its encoding.

Arguably the most common approach considered in the literature for quantization of sparse signals is based on CS techniques. In these methods, a sensing matrix is used to linearly combine the sparse signal into a lower-dimensional vector, which is then quantized, either using optimal vector quantization, as in [31], or more commonly, via some scalar continuous-to-discrete mapping, as in [27]–[30]. When the input signal is a sequentially acquired time sequence, as considered here, such CS-based techniques need to store the incoming samples in the analog domain prior to their combining using the sensing matrix.[3] This requirement, which does not exist for our proposed SQuaTS, limits the applicability of these proposed schemes, especially for long time sequences, i.e., in the regime typically considered in the literature. It should be stressed that CS-based methods assume a *simple* acquisition, i.e. conventionally linear, at the expense of a more complex decoding process. On the other hand, SQuaTS is a more involved encoding scheme that is tailored to the task of serial acquisition of sparse signals.

An additional benefit of SQuaTS compared to CS-based methods stems from its usage of binary codebooks for compression. SQuaTS operates directly on $\{s[i]\}$, and not on its lower dimensional projections as in CS, which assigns to binary codewords that originate from group testing theory based on the value of $\{s[i]\}$, and more precisely, on $\{Q(s[i])\}$. By doing so, SQuaTS achieves improved immunity to measurement errors compared to operating over fields of higher cardinality. This benefit is then translated to more accurate digital representations, as numerically demonstrated in Section VI.

Our analysis of SQuaTS is conducted assuming that the value of k used in the design of the quantization is the true sparsity level, as detailed in Section II-B. However, SQuaTS is applicable and its performance guarantees holding also when only an upper bound on k is known, and the actual sparsity pattern is smaller. This is a common assumption in the group testing literature [50], and it was shown that k can be estimated in real-time with $O(\log T)$ bits [51], [52]. In general, the sparsity assumption, i.e., $k \ll T$, allows the incorporation of group testing tools to yield accurate and computationally feasible serial quantization. When the number of non-zero samples is higher than the value of k used to design the code, the error probability increases. However, if the quantization rate of the defined code is higher than the sufficiency rate (given in Theorem 1), the decoder will not fail drastically. Only some degradation in the MSE results are obtained, as we numerically demonstrate in Section VI-A.

The codebook used by the proposed SQuaTS system, as defined in Section III-A, is generated randomly. When b is large, the probability of repetition is small [53]. In the case that b is small, one can choose "typical codewords" in the codebook generation stage, namely, only codewords without repetition to avoid errors in the recovery at the decoding process. We note that the numbers of non-zero bits in the codewords is dependent on b rather than on the sparsity level k. On average, the nTl required codewords in SQuaTS system have $p \cdot b \approx \frac{\ln 2}{k} \cdot (1+\varepsilon) k \log_2 nTl \triangleq |c(1)|$ non-zero bits. Hence, the number of possible codewords is given by, $\binom{b}{|c(1)|} \geq (\frac{b}{|c(1)|})^{|c(1)|}$, and thus to be able to generate sufficient codewords without repetitions in the codebook, it is required that

$$nTl \leq \left(\frac{b}{|c(1)|}\right)^{|c(1)|},$$

for any k, n, l, T and some $\varepsilon > 0$. Rearranging terms in the inequality results in

$$\frac{1}{\ln 2 \log_2(\frac{k}{\ln 2})} - 1 \leq \varepsilon', \tag{11}$$

where $\varepsilon = \varepsilon' + \varepsilon''$ and $0 \leq \varepsilon' < \varepsilon$. In Fig. 4 it is numerically demonstrated that for any $k \neq 1$ there are sufficient possible

[3] One may also store only the lower-dimension compressed vector and update its entries on each incoming input sample. However, this approach still requires the storage of a large amount of samples in analog, since quantization can only be conducted once the complete signal is compressed.

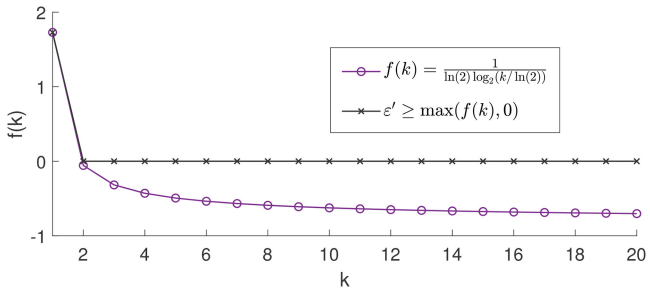


Fig. 4. Setting of ε' in (11) required to obtain sufficient possible codewords without repetitions.

codewords for any n, l, T with $\varepsilon' = 0$. For $k = 1$ increasing the size of b by ε' is required to obtain sufficient possible codewords without repetitions.

A possible drawback of SQuaTS compared to CS-schemes stems from the fact that SQuaTS is designed assuming that the signal $\{s[i]\}$ is sparse, i.e., that at most k of its entries are non-zero. CS methods are commonly capable of reliably recovering signals that are sparse in an alternative domain, namely, when a non-singular matrix $\boldsymbol{P}$ exists such that $\boldsymbol{Ps}$ is sparse, where $\boldsymbol{s}$ is the $T \times 1$ vector representation of $\{s[i]\}$. Notably, SQuaTS can still be applied to such signals by first projecting the signal using the matrix $\boldsymbol{P}$, resulting in a sparse signal that can be represented using SQuaTS. However, such application requires the entire signal to be first acquired, as is the case with conventional CS methods.

Finally, we note that while CS-based techniques typically require the ratio between the sparsity pattern size k and the input dimensionality T to be upper bounded, our proposed SQuaTS can be applied for any ratio between k and T.

2) Relationship to Group Testing Theory: As mentioned in Subsection III-A, the SQuaTS code construction is inspired by codebooks designed for group testing. Group testing first originated from the need to identify a small subset k of infected draftees with syphilis from a large set of a population with size T, using as few pool measurements b as possible. Thus, group testing measurements, i.e., codewords, are designed such that given an outcome vector of size b, one should be able to identify the defective items, namely the non-zero inputs. As discussed in Subsection III-A, a fundamental difference between our setup and conventional group testing stems from the fact that in group testing, the inputs are represented over the binary field, while in our setting, the inputs are the quantized values $\{Q(s[i])\}_{i=1}^{T}$ whose alphabet size is $l + 1$. Our code construction overcomes this difference by exploiting recent code designs targeting extended group testing models, and in particular, those considered in [44] and in the secure group testing framework [41].

The resulting group testing-based code design leads to a compact and accurate digital representation. In particular, due to the binning structure of the code suggested, when k inputs are different from zero there are only $\binom{T}{k} l^k$ possible subsets of codewords from which the output of the encoder is selected. For comparison, in a naive codebook which assigns a different codeword to each quantized input value without binning, there are $\binom{Tl}{k}$ possible subsets. This significantly reduces the number of bits required in the outcome vector.

Finally, we note that the construction of the suggested code does not depend on the distribution of the input signal, which is similar to universal quantization methods [48]. In fact, the distribution of $\{s[i]\}$ only affects the MSE induced by the serial scalar quantizer $Q(\cdot)$. The codebook presented in Subsection III-A is designed to allow for reliable reconstruction under the worst case scenario, i.e., the setting in which $\{Q(s[i])\}_{i=1}^{T}$ are i.i.d. uniformly distributed. Intuitively, the quantization rate required to achieve the MSE $D_T(l)$ can be further reduced by exploiting a-priori information on the input distribution. This approach was considered for the original group testing problem with, e.g., Poisson priors in [54]. We leave an investigation of this approach for a future study.

IV. Efficient Decoding Algorithm

The SQuaTS system detailed in the previous section and its performance analysis rely on ML decoding. In particular, the decoder uses an ML approach to identify the sub-set of k samples of the sparse signal which may be non-zero, along with their corresponding quantized values. Such a decoding algorithm suffers from a high computational complexity, as noted in Corollary 3. To overcome this drawback of SQuaTS, in this section we propose a decoding algorithm based on the Column Matching (CoMa) method used in [47] for group testing setups. The proposed algorithm is presented in Subsection IV-A. In Subsection IV-B, we analyze the algorithm performance and discuss its benefits.

A. CoMa Decoder

As mentioned above, our proposed decoding algorithm is based on the CoMa method [47]. Broadly speaking, unlike the ML decoder, which looks for the set of k codewords from different code bins that are most likely to correspond to the binary vector $\boldsymbol{y}_T$, the CoMa decoder attempts to separately match a codeword from each bin to $\boldsymbol{y}_T$. Replacing the joint search for a set of codewords with a separate examination of each codeword significantly reduces the computational burden at the cost of a degraded decoding accuracy, as we show in Subsection IV-B. The resulting decoder operates with the same code construction and encoder mapping as described for SQuaTS in Subsection III-A and III-B, respectively, thus maintaining the sequential operation and natural implementation with a practical ADCs of SQuaTS.

In particular, given an encoder output $\boldsymbol{y}_T$, the CoMa decoder consists of two stages: First, it scans the codebook $\{\mathcal{B}_i\} = \{\boldsymbol{c}_{j,i}\}$, removing all codewords that could not have resulted in $\boldsymbol{y}_T$. Since the encoding procedure, and specifically, step E3 detailed in Subsection III-B, is based on a logical OR operation between the selected codewords from each bin, any codeword that has a non-zero entry in an index of zero entries of $\boldsymbol{y}_T$ could not have been used in the encoding of $\boldsymbol{y}_T$. Once this elimination stage is concluded, the resulting set of possible codewords, which we denote by $\mathcal{C}$, is used to generate the digital representation $\{\hat{s}[i]\}$. Specifically, for every remaining codeword $\boldsymbol{c}_{j,i}$ in $\mathcal{C}$, the decoder sets $\hat{s}[i]$ to be the quantized value assigned to $\boldsymbol{c}_{j,i}$, i.e., $\hat{s}[i] = q_j$. The time instances $i \in \mathcal{T}$ for which there is no codeword in $\mathcal{C}$ are assumed to have originated from the zero codeword $\boldsymbol{c}_0$, and are thus set to q_0. If the remaining set of codewords $\mathcal{C}$ contains several codewords from the same bin, i.e., $\exists j_1 \neq j_2$ such that $\boldsymbol{c}_{j_1,i} \in \mathcal{C}$ and $\boldsymbol{c}_{j_2,i} \in \mathcal{C}$, then one of these codewords is randomly selected to generate the corresponding

Algorithm 1: CoMa Decoding.

Input : $\boldsymbol{y}_n = (y_{1,n}, \ldots, y_{b,T})$, codebook $\{\boldsymbol{c}_{j,i}\}$.
Init: $\mathcal{C} \leftarrow \{(j,i) : j \in \{1,\ldots,l\}, i \in \mathcal{T}\}$.
for $i_b = 1$ *to* b **do**
  if $\boldsymbol{y}_{i_b,T} = 0$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \backslash \{(j,i) : (\boldsymbol{c}_{j,i})_{i_b} = 1\}$;
  end
end
for $i = 1$ *to* T **do**
  if $\exists j_i$ *such that* $(j_i, i) \in \mathcal{C}$ **then**
    $\hat{s}[i] \leftarrow q_{j_i}$;
  else
    $\hat{s}[i] \leftarrow q_0$;
  end
end
Output: Recovered time sequence $\{\hat{s}[i]\}$.

recovered sample $\hat{s}[i]$. The decoding method is summarized as Algorithm 1.

B. Analysis and Discussion

The CoMa decoder is based on examining each codeword one-by one, which is less computationally complex compared to the straight forward ML evaluation, at the cost of a reduced performance. As we show next, Algorithm 1 requires a larger quantization rate R to guarantee that the MSE $D_T(l)$ is achievable for any fixed l than that required by the ML decoder detailed in Subsection III-C, thus trading computational burden for the quantization rate. The performance of the SQuaTS system using the CoMa decoder is stated in the following proposition:

Proposition 1: The MSE $D_T(l)$ is achievable via SQuaTS with the CoMa decoder in the limit $T \to \infty$ when for some $\varepsilon > 0$ the quantization rate R satisfies:

$$R \geq \bar{R}_\varepsilon(l) \triangleq \frac{(1+\varepsilon)e}{T} k \log\left(T \cdot l\right), \tag{12}$$

where e is the base of the natural logarithm. For a finite and large T, the probability of the MSE being larger than $D_T(l)$ is at most $T^{-\varepsilon}$.

Proof: The proof directly follows using similar arguments as in [47], where instead of T possible codewords, in the SQuaTS system there are $T \cdot l$ possible codewords. ■

Comparing (12) and Corollary 1 (which states an upper bound on the corresponding achievable quantization rate when using the ML decoder $R_\varepsilon(l)$) indicates that $\bar{R}_\varepsilon(l) > R_\varepsilon(l)$, i.e., the CoMa decoder requires larger quantization rates to achieve the MSE $D_T(l)$ than the ML decoder. However, Proposition 1 implies that, as the sequence length T increases, the asymptotic growth in the number of bits, $b = RT$, is $b = \mathcal{O}(k \log T + k \log l)$, i.e., the same as the growth rate characterized in Corollary 2 for SQuaTS with the ML decoder. Furthermore, Proposition 1 holds for any sparsity pattern size k, while the corresponding analysis of the ML decoder requires k not to grow with the sequence length, i.e., $k = \mathcal{O}(1)$.

Based on the characterization of the asymptotic growth of the number of bits b, we obtain the complexity of Algorithm 1. The computational burden under which SQuaTS is capable of achieving the MSE $D_T(l)$ when using the CoMa decoder is stated in the following Corollary:

Corollary 4: SQuaTS with the CoMa decoder achieves the MSE $D_T(l)$ in the limit of $T \to \infty$ with complexity on the order of $\mathcal{O}(T \cdot l \cdot k \log(T \cdot l))$ operations.

Proof: Algorithm 1 essentially scans over all the $T \cdot l$ codewords, comparing each to the b-bits binary $\boldsymbol{y}_T$. Consequently, its number of operations is on the order of $\mathcal{O}(Tlb)$. Combining this with the observation that for $b = \mathcal{O}(k \log T + k \log l)$, SQuaTS with the CoMa decoder achieves the MSE $D_T(l)$ in the limit $T \to \infty$ proves the corollary. ■

Comparing the complexity of the CoMa method in Corollary 4 to that of the ML decoder in Corollary 3 reveals the computational gains of Algorithm 1. Focusing on highly sparse setups where $T \gg k$, and recalling that $\binom{T}{kk} > \frac{T^k}{k^k}$, Corollary 3 indicates that the complexity of the ML decoder is larger than a term which is dominated by $T^k \log T$. Consequently, the ML decoder becomes infeasible as k grows. The corresponding computational complexity of the CoMa decoder in Corollary 4 is dominated by the term $T \log T$, implying that it can be implemented for practically any sparsity pattern satisfying $T \gg k$. In fact, unlike the ML decoder, Algorithm 1 is invariant to the value of k, which is required only to set the distribution of the codewords, as explained in Subsection III-A, and to determine the quantization rate under which a desired MSE level is achievable with a sufficiently high probability.

Finally, we note that the CoMa method detailed here is only one example of an efficient algorithm given in the literature that can be leveraged by the SQuaTS system decoder. In fact, it is possible to use several proposed efficient algorithms that were initially designed for the purpose of traditional group testing, see e.g., [55], [56], or even to modify the coding scheme to be based on systematic codes, as studied in [57]. An additional decoding scheme that can be combined with SQuaTS is the recently proposed two-stage multi-level decoding method recently proposed in [58] for COVID-19 pooled testing. We leave the analysis of SQuaTS with these alternative decoding methods for future investigation.

V. SQUATS for Distributed Quantization

In distributed quantization, a set of signals are acquired individually and then jointly recovered. Such setups correspond to, e.g., sensor arrays, where each bit-constrained sensor observes a different time sequence, and all the measured sequences should be recovered by some centralized server. Each sensor may have a direct link to the server, resulting in a single hop structure, or must convey its quantized measurements over a route with intermediate nodes, representing a multi-hop topology. In this section we show how SQuaTS can be naturally applied to distributed quantization setups. We first formulate the system model for distributed quantization in Subsection V-A. Then in Subsection V-B we adapt SQuaTS to distributed quantization over single-hop networks, and discuss how to it can applied to multi-hop networks in Subsection V-C.

A. Distributed Quantization System Model

We consider distributed acquisition and centralized reconstruction of n analog time sequences. The sequences, denoted by $\{s_m[i]\}_{m=1}^n$, are separately observed over the period $i \in \mathcal{T}$, representing, e.g., sources measured at distinct physical locations. The signals are jointly sparse with joint support size k [35]. We then focus on two models for the joint sparsity of $\{s_m[i]\}$:

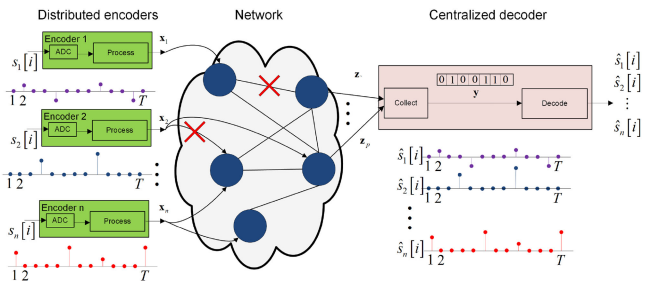


Fig. 5. Distributed quantization system illustration.

Overall sparsity: In the first model, the ensemble of all n signals over the observed duration is k-sparse, namely, the set $\{s_m[i]\}_{m\in\mathcal{N},i\in\mathcal{T}}$, with $\mathcal{N} \triangleq \{1,\ldots,n\}$, contains at most k non-zero entries. This model, in which no structure is assumed on the sparsity pattern of each signal, coincides with the general joint-sparse model of [35] without a shared component.

Structured sparsity: In the second model, the signals are sparse in both time and space. Specifically, for each $m \in \mathcal{N}$, the signal $\{s_m[i]\}_{i\in\mathcal{T}}$ is k_t-sparse, while for any $i \in \mathcal{T}$, the set $\{s_m[i]\}_{m\in\mathcal{N}}$ is k_s-sparse. This setup is a special case of the overall sparsity with $k = k_s k_t$ with an additional structure that can facilitate recovery.

Each time sequence $\{s_m[i]\}_{i\in\mathcal{T}}$ is encoded into a b-bits vector denoted as $\boldsymbol{x}_m \in \{0,1\}^b$. The encoding stage is conducted in a distributed manner, namely, each $\boldsymbol{x}_m$ is only determined by its corresponding time sequence $\{s_m[i]\}_{i\in\mathcal{T}}$ and is not affected by the remaining sequences. The binary vectors $\{\boldsymbol{x}_m\}$ are then conveyed to a single centralized decoder over a network, possibly undergoing several links over multi-hop routes. We consider a binary network model, such that each link can be either broken or error-free. The centralized decoder maintains links with P nodes. The P network outputs, denoted by $\{\boldsymbol{z}_m\}_{m=1}^P$, are collected by the decoder into a b-bits vector $\boldsymbol{y} \in \{0,1\}^b$, which is decoded into a digital representation of $\{s_m[i]\}$, denoted $\{\hat{s}_m[i]\}$, as illustrated in Fig. 5.

The accuracy is measured by the MSE $\sum_i \sum_m \mathbb{E}[(s_m[i]-\hat{s}_m[i])^2]$, and the quantization rate here is $R = \frac{b}{nT}$.

B. Single-Hop Networks

We next show how SQuaTS, proposed in Section III for the quantization of a single time sequence, can be adapted to distributed quantization setups. We begin here with single-hop networks, where each encoder has a direct error-free link to the centralized decoder. In particular, the applicability of SQuaTS to distributed quantization stems from the fact that its encoding procedure, and specifically step E3, is based on applying a logical OR operation to a set of codewords, selected according to the quantized values of each observed sample. The associative property of the logical OR operation implies that an SQuaTS encoder can be applied to an ensemble of n sequences by separately encoding each sequence with an SQuaTS encoder using a different codebook.

In particular, in order to apply SQuaTS in a distributed quantization method, one must simply generate a codebook for the ensemble of sequences, i.e., a total of nT codewords, via the generation procedure detailed in Subsection III-A. Then, the generated codewords are distributed among the n encoders, and each encoder of index $m \in \{1,\ldots,n\}$ uses these codewords to apply the SQuaTS encoding method detailed in Subsection III-B to its corresponding sequence $\{s_m[i]\}$. The output of the m-th encoder at time instance T, i.e., after its sequence is acquired, is used as the binary vector $\boldsymbol{x}_m$ conveyed to the centralized receiver. Each node in the network performs Boolean OR operations for all incoming input vectors $\boldsymbol{x}_{m'}, m' \in \{1,\ldots,P'\}$. The output of each node given P' inputs vectors is thus $\vee_{m=1}^{P'} \mathbf{x}_m$. We denote the output vector of the last nodes before the centralized decoder by $\mathbf{z}_m$. For a single-hop network, the bit vectors received by the decoder, $\{\boldsymbol{z}_m\}_{m=1}^P$, are given by $\boldsymbol{z}_m = \boldsymbol{x}_m$ and $P = n$. Consequently, by the associativity of the logical operator, the centralized decoder can recover the output of applying an SQuaTS encoder to *the ensemble of sequences*, denoted as $\boldsymbol{y}_T$, from the outputs of the separate encoders, via

$$\boldsymbol{y}_T = \bigvee_{m=1}^{P} \boldsymbol{z}_m. \tag{13}$$

Using $\boldsymbol{y}_T$, the centralized decoder can recover the estimate of the ensemble of time sequences $\{\hat{s}_m[i]\}$ via conventional SQuaTS decoding, i.e., the ML decoding detailed in Subsection III-C or the CoMa method presented in Section IV.

The fact that the proposed distributed adaptation of SQuaTS effectively implements the application of SQuaTS to the ensemble of sequences implies that the achievable performance guarantees of SQuaTS, derived in Subsection III-D and IV-B for the ML and CoMa decoders, respectively, also hold in the distributed setup. For example, by letting $D_{T,n}(l)$ be the MSE achieved when $\hat{s}_m[i] = Q(s_m[i])$, namely, a desirable MSE determined only by the ADC resolution, it follows from Theorem 1 that $D_{T,n}(l)$ is achievable by the distributed quantization scheme when its rate satisfies the condition stated in the following proposition:

Proposition 2: SQuaTS adapted to distributed quantization using the ML decoder achieves the MSE $D_{T,n}(l)$ in the limit $nT \to \infty$ with $k = \mathcal{O}(1)$ when the quantization rate R satisfies the following inequality for some $\varepsilon > 0$:

$$R \geq \max_{u\in\mathcal{I}(k)} \frac{(1+\varepsilon)k}{u \cdot nT} \log\left(\vartheta \cdot l^u\right). \tag{14}$$

The parameter ϑ and the set $\mathcal{I}(k)$ depend on the type of joint sparsity: for overall sparsity, $\vartheta = \binom{nT}{k}$ and $\mathcal{I}(k) = \{1,\ldots,k\}$; for structured sparsity, $\vartheta = \binom{n}{k_t}\binom{T}{k_s}$, and $\mathcal{I}(k = k_s k_t) = \{u_t u_s : 1 \leq u_t \leq k_t, 1 \leq u_s \leq k_s\}$.

Proof: The proposition follows by repeating the proof of Theorem 1 given in the Appendix while setting the length of the sequence to be nT, i.e., the length of the ensemble of signals, instead of T, and noting that the joint sparsity affects the number of possible codeword combinations. In particular, ϑ is the number of possible sets of non-zero entries in the ensemble of signals over which the ML decoder searches, which is used in Lemma 1 in the Appendix. ■

Proposition 2 indicates that, as expected, SQuaTS can exploit structures in the joint sparse nature of the observed signals to improve performance, namely, to utilize less bits while guaranteeing that a desired MSE $D_{T,n}(l)$ is achievable.

C. Multi-Hop Networks

We now generalize our scheme to a multi-hop network, in which multiple directed links relate the distributed encoders and

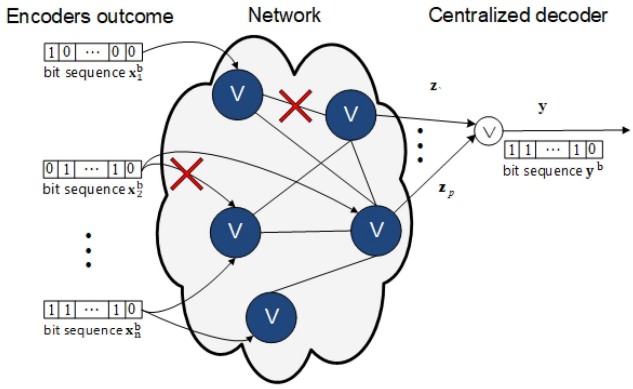


Fig. 6. Acquisition process in multi-hop Networks.

the centralized decoder. The intermediate nodes in the networks, which act as helpers or relays, can perform basic operations on their input from incoming links. For the sake of space and exposition, we consider a simplified model for this communication network, in which links are assumed to either support b-bits of information without errors or result in a complete erasure. We also assume that the transmission is synchronized, i.e., the encoders and intermediate nodes all transmit in sync across their outgoing links, and that the network is acyclic. Note that despite its simplicity, this model is reminiscent of several network models used in the literature, e.g., [15].

The operation of the encoders and the decoder in the multi-hop setup is identical to that discussed for single hop networks in Subsection V-B. The only addition is in the network policy, as depicted in Fig. 6: At each intermediate node, we perform a Boolean OR operation of all the incoming input vectors (which is the same mathematical operation performed by the encoder and decoders in Subsection V-B), and transmit the resulting length b-vector on all outgoing links. The network outputs are then collected in $\boldsymbol{y}_T$ via (13).

Clearly the resulting bit sequence at the decoder $\boldsymbol{y}_T$ is identical to the one in Subsection V-B, as long as at least one path exists in the network from each encoder to the centralized decoder. Note that this is in contrast with the previous literature on distributed CS over networks, where it is typical to impose conditions on the network topology that guarantee a successful description [38]. Consequently, the structures of the encoders and the decoder are invariant to whether the encoders communicate with the decoder directly or over multi-hop networks, and the achievable performance of SQuaTS stated in, e.g., Proposition 2, holds in such multi-hop networks. Additionally, the scheme we propose is robust to link failures: as long as at least one path exists from all encoders to the decoder, any number of link failures in the network still leads to the same received vector at the decoder, i.e., the coding scheme can achieve the min-cut max-flow bound of the network [15], [59].

It follows from the above discussion that the presence of a multi-hop network does not affect the operation of the distributed adaptation of SQuaTS or its achievable performance, and only requires a simplified network policy to be conducted by the intermediate network nodes. While our analysis assumes that each encoder has at least a single path to the decoder, it can be shown that the presence of missing paths for some encoders does not affect the recovery of the remaining signals. In particular, by treating the output of a broken link as the zero vector, if the mth encoder has no path to the decoder, then the recovery of $\{s_j[i]\}_{j\neq m}$ remains intact, while $\hat{s}_m[i]$ is estimated as being all zeros.

VI. Numerical Evaluations

In this section, we evaluate the performance of the proposed SQuaTS scheme using various simulations for a fixed and finite signal size T. We first numerically evaluate SQuaTS for the quantization of a single sparse signal in Subsection VI-A. Then in Subsection VI-B we study the resiliency of SQuaTS to noisy digital representations. Finally, in Subsection VI-C we evaluate its extension to distributed quantization setups, as discussed in Section V.

A. Single Sparse Signal

We begin by numerically evaluating SQuaTS used for quantizing a single sparse signal $\{s[i]\}, i \in \mathcal{T}$. To this end, we consider two sparse sources with sizes $T \in \{100, 50\}$ and support sizes $k \in \{3, 2\}$, respectively. To generate each signal, we randomly select k indexes, denoted as $\{i_j\}_{j=1}^k$, and then choose the values of $\{s[i_j]\}$ to be i.i.d. zero-mean unit-variance Gaussian random variables (RVs), while the remaining entries are set to zero.

Each of the generated signals is quantized and represented in digital form using each of the following methods:

- SQuaTS system with $l = \max(\lfloor \frac{1}{T} 2^{\frac{TR}{k(1+\epsilon)}} \rfloor, 2)$ following (9), where ϵ is selected empirically for each point in the range $\epsilon \in [0.8, 1.3]$ to maximize the performance of the ML decoder, where we select the value which achieves the minimal MSE among four different values of ϵ uniformly placed in this region. Here the continuous-to-discrete mapping $Q(\cdot)$ implements uniform quantization over the region $[-2, 2]$. We consider three different SQuaTS decoders: the ML decoder detailed in Subsection III-C; the CoMa-based reduced complexity decoder proposed in Section IV, which is tuned with the same value of ϵ as that used by the ML decoder; and the CoMa decoder with ϵ optimized separately by a fine search for each quantization rate.
- A uniform scalar quantizer with support $[-2, 2]$ is applied separately to each sample of $\{s[i]\}$, mapping every decision region to its centroid. This system, which models the direct application of a serial scalar ADC to the sparse signal $\{s[i]\}$, can be utilized only when the quantization rate satisfies $R \geq 1$, as the quantizers require at least one bit.
- A compress-and-quantize system, which first compresses $\boldsymbol{s} = [s[1], \ldots, s[T]]^T$ into $\mathcal{R}^m$, where m is selected in the range $[6k, 20k]$ to minimize the MSE. The compression is conducted using a sensing matrix $\boldsymbol{A} \in \mathcal{R}^{m \times T}$ whose entries are i.i.d. zero-mean unit variance Gaussian RVs. The compressed signal $\boldsymbol{As}$ is quantized using a uniform scalar quantizer with support $[-2, 2]$. The digital representation $\hat{\boldsymbol{s}} = [\hat{s}[1], \ldots, \hat{s}[T]]^T$ is then recovered using the quantized iterative hard thresholding (QIHT) method [60] as well as fast iterative soft thresholding algorithm (FISTA) [61].

All of the above schemes are compared with the same number of bits $b = R \cdot T$, and the MSE is computed by averaging the squared error over 100 Monte Carlo simulations.

The empirically evaluated MSEs of the considered quantization systems versus the quantization rate R are depicted in Figs. 7–8 for the setups with $(T, k) = (100, 3)$ and $(T, k) =$

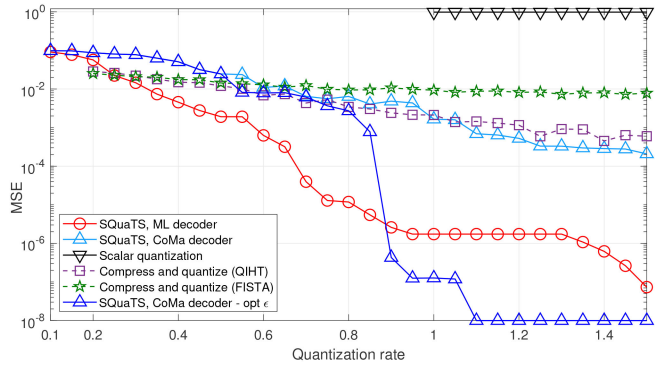


Fig. 7. Quantization systems comparison, $T = 100$, $k = 3$.

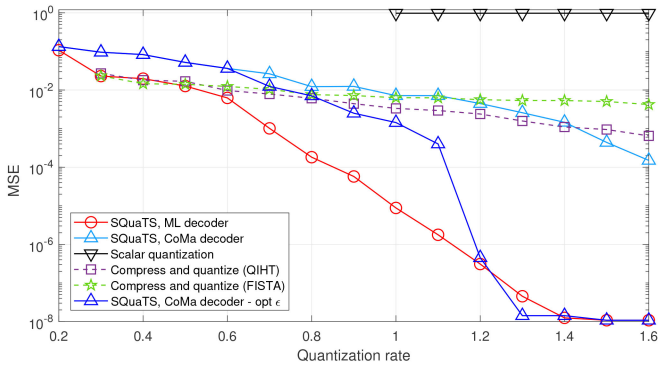


Fig. 8. Quantization systems comparison, $T = 50$, $k = 2$.

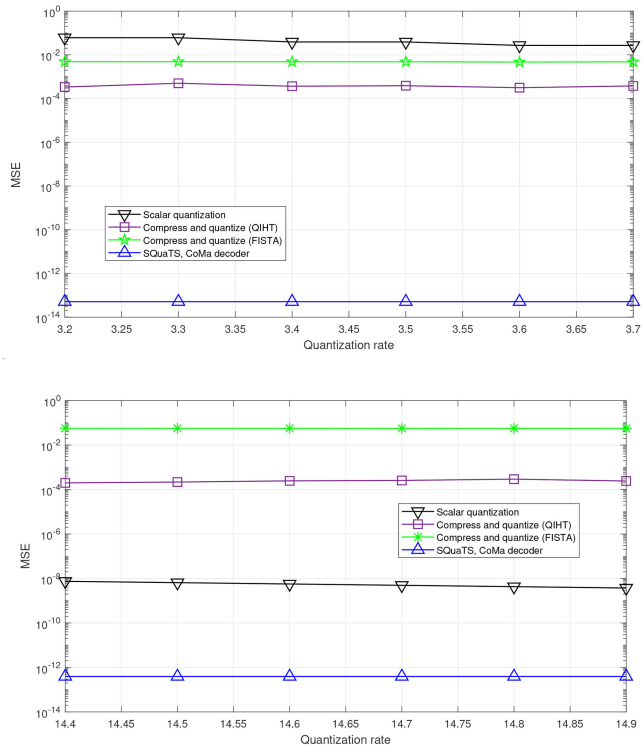


Fig. 9. Quantization systems comparison with CoMa decoder for $T = 50$. In the top panel $k = 2$ and in the bottom panel $k = 10$.

$(50, 2)$, respectively. Observing Figs. 7–8, it is noted that the proposed SQuaTS system achieves a superior representation accuracy and that its resulting MSE is not larger than 10^{-4} for quantization rates $R \geq 0.8$. For comparison, directly applying a scalar quantizer to the sparse signal is only feasible for $R \geq 1$, and its achievable MSE is only slightly less than 1. This degraded performance of directly applying scalar quantizers stems from the fact that for the considered rates $R \in [1, 2)$, this quantization mapping implements a one-bit sign quantization of $\{s[i]\}$. Since most of the samples of $\{s[i]\}$ are zero, this quantization rule induces a substantial distortion.

The MSE performance of the CS-based quantization scheme improves much less dramatically with the quantization rate R compared to SQuaTS. For example, for the scenario depicted in Fig. 7, the SQuaTS system achieves an MSE of $2 \cdot 10^{-3}$ for $R = 0.5$, while the CS-based systems achieve an MSEs of $1.2 \cdot 10^{-2}$ and $1.4 \cdot 10^{-2}$ for the QIHT and FISTA decoders, respectively, i.e., a gap of approximately 7 dB. However, for a quantization rate of $R = 1$, the corresponding MSE values are $1.7 \cdot 10^{-6}$, $2 \cdot 10^{-3}$, and $9 \cdot 10^{-3}$, for the SQuaTS system, CS with QIHT recovery, and CS with FISTA recovery, respectively; namely, performance gaps of $30 - 37$ dB in MSE. For all considered scenarios, the QIHT recovery scheme, which is specifically designed for reconstructing sparse signals from compressed and quantized measurements, outperforms the FISTA method, which considers general sparse recovery.

Furthermore, it is observed in Figs. 7–8 that the SQuaTS encoder combined with the reduced complexity CoMa detector proposed in Section IV outperforms CS-based quantizers as the quantization rate R increases when using both a fixed ϵ as well as an optimized setting of this parameter. In particular, for the scenario whose results are depicted in Fig. 7, SQuaTS with the CoMa decoder without ϵ optimization outperforms compress-and-quantize with QIHT and FISTA recovery for rates $R > 0.6$ and $R > 1$, respectively. The corresponding rate thresholds for the scenario in Fig. 8 are $R > 1.1$ and $R > 1.4$, respectively. However, when optimizing ϵ specifically for the CoMa decoder in a fine manner, which can be conducted without substantial overhead due to its reduced computational complexity, its performance is notably improved. In fact, in high quantization rates we observe that the optimization of ϵ allows the CoMa decoder to outperform the ML decoder, for which ϵ is merely selected out of four possible candidates.

The ability of SQuaTS with the CoMa decoder to notably outperform CS-based approaches is also demonstrated for higher resolution quantization regimes in Fig. 9. This is achieved by increasing the quantization rate to satisfy the sufficiency rate in Proposition 1 and selecting an appropriate l. Fig. 10 numerically evaluates SQuaTS combined with fragmentation under the scenario presented in the bottom panel of Fig. 9. In particular, fragmentation is performed here to reduce the computational burden by dividing the signal into ten groups, as discussed in Section III-D, resulting in $T = 5$ and $k = 1$. We observe that CoMa also achieves an improved performance for various quantization rates by properly setting l, as demonstrated in Fig. 14. These results indicate that SQuaTS, which acquires the sparse signal in a serial manner, is capable of achieving a superior performance, even when using sub-optimal reduced complexity decoders, compared to CS-based methods that must observe the complete signal before it is quantized.

Fig 11 numerically evaluates the performance obtained for the scenario discussed in Section III-E, in which the value of k used

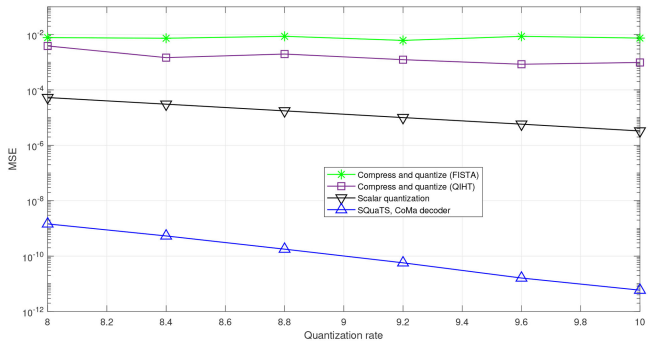


Fig. 10. Quantization systems comparison for fragmentation of $T = 50$ and $k = 10$, as given in Fig. 9, to ten groups such that $T = 5$ and $k = 1$.

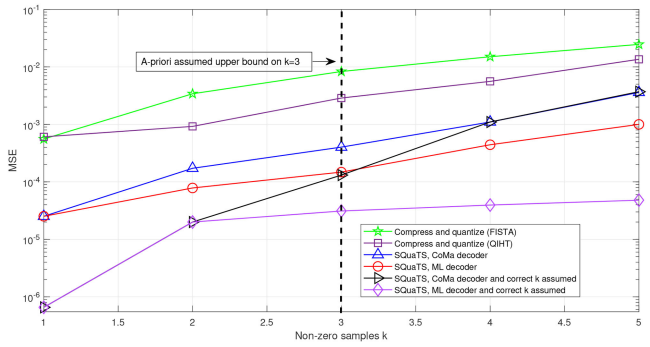


Fig. 11. Quantization systems comparison for different numbers of non-zero inputs, $T = 100$, quantization rate of $R = 0.83$ and SQuaTS codebook designed for upper bound of $k = 3$.

in designing the SQuaTS system differs from the true sparsity level. Here we designed the system for $k = 3$, while evaluating its performance when the actual number of the non-zero samples varies in the proximity of $k = 3$. For lower non-zero samples case, we observe in Fig. 11 that SQuaTS still exploits the reduced sparsity level, allowing it to be translated into an improved performance, despite the fact that it was designed with higher values of k. When the true value of k is larger than that used in the design, we observe some graceful degradation in the MSE accuracy of SQuaTS, indicating its ability to maintain a reliable operation in such scenarios. In summary, if the number of non-zero input samples is higher or lower than the a priori selected k, the codebook designed for k can still be applied. In such cases, when the mismatch is not too large, only a minor degradation in the quantization distortion is observed. However, when this number of non-zero inputs samples is substantially different than the a-priori assumption, one will need different codebook designs, and a look up table if this is used to reduce the decoding complexity with an ML algorithm. For comparison, the evaluating results for the case of correct a-priori assumed values of k are presented in Fig. 11 using the SQuaTS system with the corresponding values of l for the ML and CoMa decoders.

Finally, Figs. 12 and 13 show the trade-off between the quantization rate required for achieving a desired distortion level $D_T(l)$, computed via (7), with the parameters k and l, for $T = 50$ and $T = 100$, respectively. The numerical results in Figs. 12 and 13 demonstrate how the quantization rate scales with l and k. These observed curves settle with the characterization

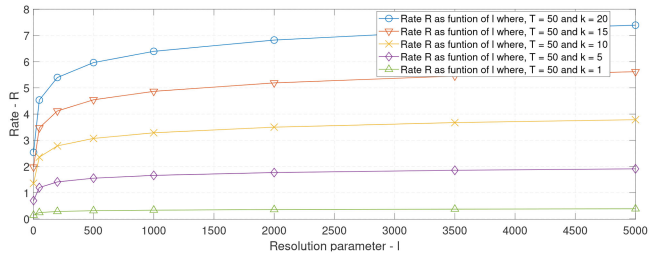


Fig. 12. Numerical results of SQuaTS system, for $T = 50$ and different values of l and k.

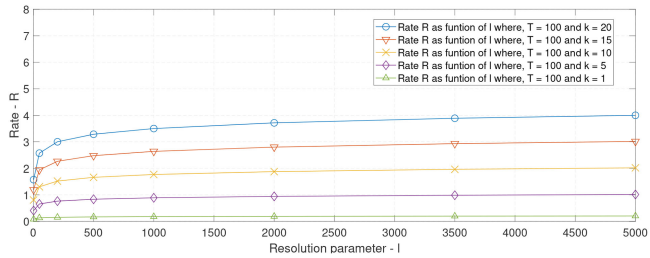


Fig. 13. Numerical results of SQuaTS system, for $T = 100$ and different values of l and k.

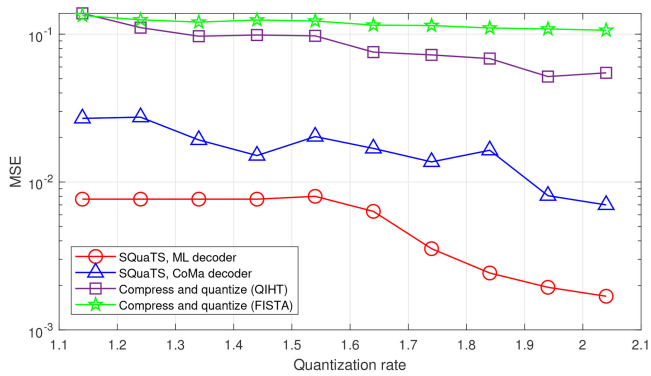


Fig. 14. Quantization systems comparison, $T = 50$ and $k = 5$. This scenario is equivalent to the case of $T = 100$ and $k = 10$, in which we are performing fragmentation to two groups. In this case, the quantization rate is divided by the number of groups.

in Corollary 1, which noted that the rate scales logarithmically with l and linearly with k. The results depicted in Figs. 12 and 13 demonstrate the ability of SQuaTS to exploit the signal sparsity and to benefit from small values of the support k, as the required quantization rate is reduced substantially as k decreases. To demonstrate that the performance gains of SQuaTS persist in such regimes, we numerically evaluate its performance for a scenario with $T = 50$ and $k = 5$. This scenario, for which the empirical performance is depicted in Fig. 14, is equivalent to the scenario of $T = 100$ and $k = 10$, in which fragmentation into two groups is used to reduce the decoding complexity. as suggested in Section III-D. Note that by performing fragmentation, the quantization rate is divided by the number of groups. Fig. 14 also presents the performance with CoMa decoding for a low quantization rate when ϵ is optimized for this decoder. In practice, for a high k using an ML decoder, fragmentation is needed due to the computational complexity. In the regime

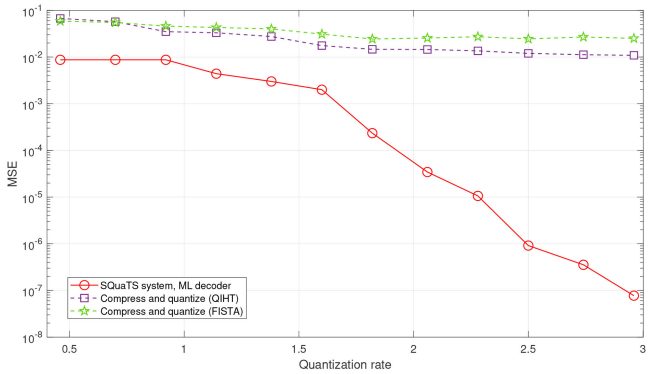


Fig. 15. Quantization systems comparison, $T = 50$, $k = 2$, $q = 0.1$, $u = 0.1$.

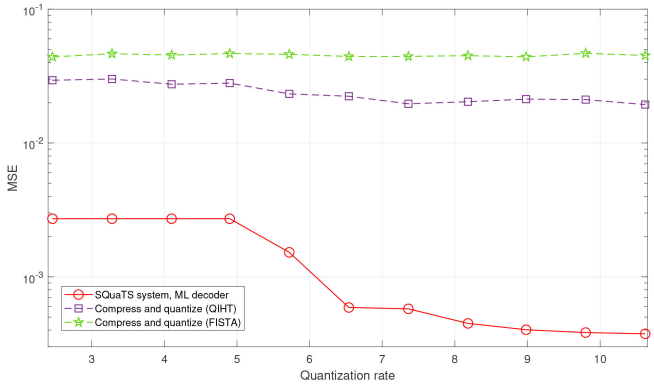


Fig. 16. Quantization systems comparison, $T = 50$, $k = 2$, $q = 0.4$, $u = 0.1$.

we simulated using CoMA fragmentation is not needed. We observe in Fig. 14 that the expected gains of SQuaTS are indeed maintained here.

B. Noisy Digital Representation

Next, we conduct a set of experiments whose goal is to demonstrate how SQuaTS can be adapted to deal with noisy digital representations. To that end, we consider the setup in which the value of the bits register $\boldsymbol{y}_T$ experience-independent random bit-flips, i.e., each bit $y_{i,T}$ is flipped from 0 to 1 with a probability $0 \leq q < 1/2$ (positive flip) and flipped from 1 to 0 with a probability $0 \leq u < 1$ (negative flip). This noisy model represents corruption of the codeword in the digital domain.

By making use of existing results in noisy group testing theory, we propose increasing the length of the codewords by some factor that depends on u and q, while keeping the encoding identical. A trivial adaptation of the results in [53] to our setup, reveals that increasing the length of the codeword b by a factor of $\frac{1}{1-q} \cdot \frac{1}{(1-u)^2}$ is sufficient.[4] On the side of the decoder, the ML scheme operates identically. However, the efficient CoMa method must be modified to deal with this noise in the system. While a precise discussion of this is outside of the scope of this paper, we refer the interested reader to existing efficient algorithms for noisy group testing, such as Noisy-CoMa [63], as possible ways to tweak the CoMa decoding scheme of Subsection IV-A to account for the presence of digital noise.

Figs. 15 and 16 show the empirical MSE of the adapted scheme under the same signal settings considered in the previous subsection in the presence of digital noise with parameters $(q = 0.1, u = 0.1)$ and $(q = 0.4, u = 0.1)$, respectively. We observe in Figs. 15–16 that the quantization rate R required to achieve a given MSE level is increased compared to the noiseless case in Figs. 7–8 – quantifying the additional bits that enable SQuaTS to be robust to digital noise. More precisely, to achieve an average MSE of 10^{-3} with the proposed scheme, the quantization rate must be increased from. 7, as required in the absence of digital noise, to approximately 1.6 and 6, when $q = 0.1$ and $q = 0.4$, respectively. As observed, this loss in performance is much more dramatic when q grows, revealing that positive flips are more costly than negative flips. In either cases, the performance of the adapted SQuaTS scheme significantly outperforms the CS-based approaches, which fail to breach under the MSE of 10^{-2}, even for large quantization rates.

[4]While this factor is relatively loose, it is preferred here over the complex yet more precise expression that can be found in [62] due to its simple formulation.

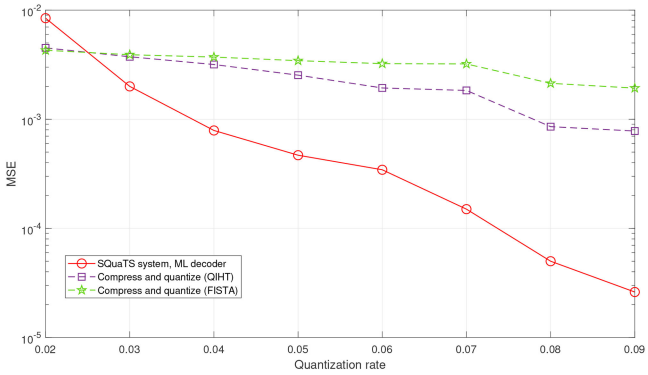


Fig. 17. Distributed quantization systems comparison, $T = 100$, $n = 10$, $k = 3$.

C. Distributed Quantization

We end this section with a numerical study of the extension of SQuaTS to distributed quantization setups, as detailed in Section V. To that end, we numerically compute the achievable MSE of SQuaTS applied to $n = 10$ jointly sparse signals of size $T = 100$ with a joint support size of $k = 3$ in a single hop network. In Fig. 17, we compare the MSE of distributed SQuaTS to that of distributed CS [35], in which the quantized values of compressed projections are aggregated and recovered by the central decoder. We consider the cases where the decoder recovers the set of signals using the QIHT method [60] as well as FISTA [61]. While more advanced schemes combining distributed CS and vector quantization were proposed in [40], their complexity grows rapidly when $n > 2$, and thus we focus on conventional distributed CS with scalar quantization.

Observing Fig. 17, we note that the proposed distributed quantization scheme notably outperforms techniques based on a distributed CS. In particular, our method is shown to substantially improve the accuracy of the overall digital representation as the quantization rate increases, while distributed quantized CS is demonstrated to meet an error floor of approximately $4 \cdot 10^{-2}$ for FISTA and $9 \cdot 10^{-3}$ for QIHT.

Finally, we demonstrate how the minimal quantization rate grows with the resolution l. To that end, in Fig. 18 we compute the minimal rate versus l. The setup evaluated here consists of

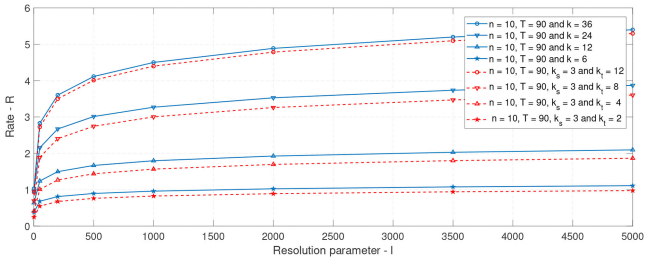


Fig. 18. Quantization rate threshold versus the resolution l.

$n = 10$ sequences of $T = 90$ samples each, for both the overall sparsity with $k \in \{6, 12, 24, 36\}$ as well as structured sparsity with the same overall sparsity level and $k_s = 3$. Observing Fig. 18, we note that structured sparsity allows to use lower quantization rates, i.e., fewer bits, to achieve the same level of distortion due to the additional structure. We also note that the quantization rate grows slowly with l, indicating that a minor increase in the quantization rate can allow the scheme to utilize the ADCs of a much higher resolution while maintaining the guaranteed performance. Observing Fig. 18, we note that structured sparsity allows for using lower quantization rates, i.e., fewer bits, to achieve the same level of distortion.

The results presented in this section demonstrate the potential of SQuaTS as a quantization scheme for sparse signals. This method is both accurate and suitable for implementations with a conventional serial scalar ADCs. Our results also demonstrate the ability of SQuaTS to implement distributed quantization and the robustness of SQuaTS to digital noise.

VII. Conclusion

In this paper we proposed SQuaTS, a quantization system designed for representing sparse signals acquired in a sequential manner. SQuaTS combines code structures from group testing theory with the limitations and characteristics of conventional ADCs. We derived the achievable MSE of the proposed scheme in the asymptotic signal size regime and characterized its complexity. We proposed a reduced complexity decoding method for SQuaTS, which trades performance for computational burden while maintaining the sequential acquisition property of SQuaTS, and showed how SQuaTS can be extended to distributed setups. Our simulation study demonstrates the substantial performance gain of SQuaTS compared to directly applying a serial scalar ADC, as well as to CS-based methods.

Appendix

To prove the Theorem 1, we first provide a reliability bound that guarantees accurate reconstruction of the quantized representation $\{Q(s[i])\}_{i=1}^{T}$ from $\boldsymbol{y}_T$. Then, we show that this bound results in the condition on the quantization rate stated in Theorem 1. An achievability bound on the required number of bits is stated in the following lemma:

Lemma 1: If for some $\varepsilon > 0$ independent of T and k, the number of bits used for digital representation satisfies

$$b \geq \max_{1 \leq i \leq k} \frac{(1+\varepsilon)k}{i} \log \binom{T-k}{i} l^i, \tag{15}$$

then, under the code construction of Section III, as $T \to \infty$ the average error probability to recover $\{Q(s[i])\}_{i=1}^{T}$, given by $\frac{1}{T}\sum_{i=1}^{T} \Pr(\hat{s}_i \neq Q(s[i]))$, approaches zero exponentially.

Proof: The Lemma follows from [44, Lemma 1], whose proof is based on [41, Lemma 2]. ■

We note that the corresponding bound in [53, Theorem III.1], in which group testing is conducted over a binary field, can be considered a special case of Lemma 1 with $l = 1$, i.e., using one-bit quantizers. In particular, since we consider quantizers with an arbitrary resolution, the bound in Lemma 1 must account for the fact that the codewords have to be selected from different bins, since l can be larger than one.

Now, Lemma 1 yields a sufficient condition for the digital representation $\{\hat{s}[i]\}$ to approach the directly quantized $\{s[i]\}$, for which the MSE is $D_T(l)$ given in (6). Since the quantization rate is given by $R = \frac{b}{T}$, the condition (15) becomes

$$R \geq \max_{1 \leq i \leq k} \frac{(1+\varepsilon)k}{i \cdot T} \log \left(\binom{T-k}{i} \cdot l^i \right),$$

proving the theorem. ■

References

[1] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[3] S. Kosonocky and P. Xiao, "Analog-to-digital conversion architectures," *Digital Signal Processing Handbook*, Boca Raton, FL, USA: CRC Press LLC, 1999, pp. 106–119.

[4] Y. Polyanskiy and Y. Wu, "Lecture notes on information theory," *Lecture Notes ECE563 (UIUC)*, vol. 6, pp. 2012–2016, 2014.

[5] A. Bhatt, B. Nazer, O. Ordentlich, and Y. Polyanskiy, "Information-distilling quantizers," *IEEE Trans. Inf. Theory*, vol. 67, no. 4, pp. 2472–2487, Apr. 2021.

[6] L. P. Barnes, Y. Han, and A. Ozgur, "Lower bounds for learning distributions under communication constraints via fisher information," *J. Mach. Learn. Res.*, vol. 21, no. 236, pp. 1–30, 2020.

[7] N. Shlezinger, Y. C. Eldar, and M. Rodrigues, "Hardware-limited task-based quantization," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5223–5238, Oct. 2019.

[8] N. Shlezinger, Y. C. Eldar, and M. R. Rodrigues, "Asymptotic task-based quantization with application to massive MIMO," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 3995–4012, Aug. 2019.

[9] S. Salamatian, N. Shlezinger, Y. C. Eldar, and M. Médard, "Task-based quantization for recovering quadratic functions using principal inertia components," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 390–394.

[10] N. Shlezinger and Y. C. Eldar, "Deep task-based quantization," *Entropy*, vol. 23, no. 1, pp. 104–122, 2021.

[11] J. A. Gubner, "Distributed estimation and quantization," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1456–1459, Jul. 1993.

[12] W.-M. Lam and A. R. Reibman, "Design of quantizers for decentralized estimation systems," *IEEE Trans. Commun.*, vol. 41, no. 11, pp. 1602–1605, Nov. 1993.

[13] T. Berger, Z. Zhang, and H. Viswanathan, "The CEO problem [multiterminal source coding]," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 887–902, May 1996.

[14] Y. Oohama, "The rate-distortion function for the quadratic Gaussian ceo problem," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1057–1070, May 1998.

[15] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[16] N. Shlezinger, S. Salamatian, Y. C. Eldar, and M. Médard, "Joint sampling and recovery of correlated sources," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 385–389.

[17] A. Saxena, J. Nayak, and K. Rose, "On efficient quantizer design for robust distributed source coding," in *Proc. IEEE Data Compression Conf.*, 2006, pp. 63–72.

[18] N. Wernersson, J. Karlsson, and M. Skoglund, "Distributed quantization over noisy channels," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1693–1700, Jun. 2009.
[19] M. Fleming, Q. Zhao, and M. Effros, "Network vector quantization," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1584–1604, Aug. 2004.
[20] N. Wagner, Y. C. Eldar, and Z. Friedman, "Compressed beamforming in ultrasound imaging," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4643–4657, Sep. 2012.
[21] Y. Shechtman, A. Beck, and Y. C. Eldar, "GESPAR: Efficient phase retrieval of sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 928–938, Feb. 2014.
[22] M. Rossi, A. M. Haimovich, and Y. C. Eldar, "Spatial compressive sensing for MIMO radar," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 419–430, Jan. 2014.
[23] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, Nov. 2010.
[24] S. Feizi and M. Médard, "A power efficient sensing/communication scheme: Joint source-channel-network coding by using compressive sensing," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 1048–1054.
[25] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[26] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.
[27] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable Embeddings of sparse vectors," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2082–2102, Apr. 2013.
[28] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Proc. IEEE 42nd Annu. Conf. Inf. Sci. Syst.*, 2008, pp. 16–21.
[29] L. Jacques, D. K. Hammond, and J. M. Fadili, "Dequantizing compressed sensing: When oversampling and non-gaussian constraints combine," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 559–571, Jan. 2011.
[30] C. S. Güntürk, M. Lammers, A. Powell, R. Saab, and Ö. Yilmaz, "Sigma delta quantization for compressed sensing," in *Proc. IEEE 44th Annu. Conf. Inf. Sci. Syst.*, 2010, pp. 1–6.
[31] A. Kipnis, G. Reeves, and Y. C. Eldar, "Single letter formulas for quantized compressed sensing with gaussian codebooks," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 71–75.
[32] P. T. Boufounos, L. Jacques, F. Krahmer, and R. Saab, "Quantization and compressive sensing," in *Compressed Sensing and Its Applications*. Berlin, Germany: Springer, 2015, pp. 193–237.
[33] R. Saab, R. Wang, and Ö. Yılmaz, "Quantization of compressive samples with stable and robust recovery," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 1, pp. 123–143, 2018.
[34] S. Sarvotham, D. Baron, M. Wakin, M. F. Duarte, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2005, pp. 1537–1541.
[35] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," 2009, *arXiv:0901.3403*.
[36] T. T. Do, Y. Chen, D. T. Nguyen, N. Nguyen, L. Gan, and T. D. Tran, "Distributed compressed video sensing," in *Proc. IEEE Int. Conf. Image Process.*, 2009, pp. 1393–1396.
[37] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 4931–4946, Oct. 2014.
[38] S. Feizi, M. Médard, and M. Effros, "Compressive sensing over networks," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2010, pp. 1129–1136.
[39] A. Shirazinia, S. Chatterjee, and M. Skoglund, "Distributed quantization for measurement of correlated sparse sources over noisy channels," 2014, *arXiv:1404.7640*.
[40] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed distortion-rate optimized compressed sensing in wireless sensor networks," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1609–1623, Apr. 2018.
[41] A. Cohen, A. Cohen, and O. Gurewitz, "Secure group testing," *IEEE Trans. Inf. Forensics Security*, to be published, doi: 10.1109/TIFS.2020.3029877.
[42] J. Li, N. Chaddha, and R. M. Gray, "Asymptotic performance of vector quantizers with a perceptual distortion measure," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1082–1091, May 1999.
[43] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.
[44] A. Cohen, A. Cohen, and O. Gurewitz, "Efficient data collection over multiple access wireless sensors network," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 491–504, Apr. 2020.
[45] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Statist.*, vol. 14, no. 4, pp. 436–440, 1943.
[46] P. Panter and W. Dite, "Quantization distortion in pulse-count modulation with nonuniform spacing of levels," *Proc. IRE*, vol. 39, no. 1, pp. 44–48, 1951.
[47] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 3019–3035, May 2014.
[48] J. Ziv, "On universal quantization," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 3, pp. 344–347, May 1985.
[49] R. Zamir and M. Feder, "On universal quantization by randomized uniform/lattice quantizers," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 428–436, Mar. 1992.
[50] A. J. Macula, "Probabilistic nonadaptive group testing in the presence of errors and DNA library screening," *Ann. Combinatorics*, vol. 3, no. 1, pp. 61–69, 1999.
[51] P. Damaschke and A. Muhammad, "Bounds for nonadaptive group tests to estimate the amount of defectives," *Proc. Combinatorial Optim. Appl.*, 2010, pp. 117–130.
[52] P. Damaschke and A. S. Muhammad, "Competitive group testing and learning hidden vertex covers with minimum adaptivity," *Discrete Math., Algorithms Appl.*, vol. 2, no. 3, pp. 291–311, 2010.
[53] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1880–1901, Mar. 2012.
[54] A. Emad and O. Milenkovic, "Poisson group testing: A probabilistic model for nonadaptive streaming boolean compressed sensing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 3335–3339.
[55] M. Aldridge, L. Baldassini, and O. Johnson, "Group testing algorithms: Bounds and simulations," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3671–3687, Jun. 2014.
[56] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Information-theoretic and algorithmic thresholds for group testing," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7911–7928, Dec. 2020.
[57] T. V. Bui, M. Kuribayashi, T. Kojima, R. Haghvirdinezhad, and I. Echizen, "Efficient (nonrandom) construction and decoding for non-adaptive group testing," *J. Inf. Process.*, vol. 27, pp. 245–256, 2019.
[58] A. Cohen, N. Shlezinger, A. Solomon, Y. C. Eldar, and M. Médard, "Multi-level group testing with application to one-shot pooled COVID-19 tests," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 1030–1034.
[59] G. Dantzig and D. R. Fulkerson, "On the max flow min cut theorem of networks," *Linear Inequalities Related Syst.*, vol. 38, pp. 225–231, 2003.
[60] L. Jacques, K. Degraux, and C. De Vleeschouwer, "Quantized iterative hard thresholding: Bridging 1-bit and high-resolution quantized compressed sensing," 2013, *arXiv:1305.1786*.
[61] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[62] D. Sejdinovic and O. Johnson, "Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2010, pp. 998–1003.
[63] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama, "Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 1832–1839.

Alejandro Cohen (Member, IEEE) received the B.Sc. degree from the Department of Electrical Engineering, SCE College of Engineering, Israel, in 2010, and the M.Sc. and Ph.D. degrees from the Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beersheba, Israel, in 2013 and 2018, respectively. He is currently a Senior Postdoctoral Associate with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. From 2007 to 2014, he was with DSP Group, where he worked on voice enhancement and signal processing. From 2014 to 2019, he was with Intel, where he was a Research Scientist with Innovation Group at Mobile and Wireless. His research interests include information theory, signal processing, and networks. In particular, he is interested in wireless communication, security, network information theory and network coding, anomaly detection, coding, computation in networks, and speech enhancement.

Nir Shlezinger (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering from Ben-Gurion University, Israel, in 2011, 2013, and 2017, respectively. From 2017 to 2019, he was a Postdoctoral Researcher with Technion, and from 2019 to 2020, he was a Postdoctoral Researcher with the Weizmann Institute of Science, Rehovot, Israel, where he was awarded the FGS prize for outstanding research achievements. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Ben-Gurion University, Israel. His research interests include communications, information theory, signal processing, and machine learning.

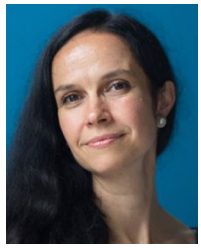
Salman Salamatian received the B.Sc. and M.S. degrees in communication systems from École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2009 and 2012, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2020. He is currently a quantitative Researcher with The D.E. Shaw Group. His research interests include the intersection of information theory, privacy and security, statistical learning, and more recently finance theory.

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics and the B.Sc. degree in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2002.

She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. She was previously a Professor with the Department of Electrical Engineering, Technion, where she held the Edwards Chair in Engineering. She is also a Visiting Professor with MIT, a Visiting Scientist with Broad Institute, and an Adjunct Professor with Duke University, Durham, NC, USA, and was a Visiting Professor with Stanford. She is the author of the book *Sampling Theory: Beyond Bandlimited Systems* and coauthor of four other books published by Cambridge University Press. Her research interests include the broad areas of statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology, medical imaging, and optics.

She is a Member of the Israel Academy of Sciences and Humanities (elected 2017) and a EURASIP Fellow. She was the recipeint of many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award (2013), the IEEE/AESS Fred Nathanson Memorial Radar Award (2014), and the IEEE Kiyo Tomiyasu Award (2016). She was a Horev Fellow of the Leaders in Science and Technology program at the Technion and an Alon Fellow. She was the recipient of the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel & David Jacknow Award for Excellence in Teaching, and the Technion's Award for Excellence in Teaching (two times). She was also the recipient of several best paper awards and best demo awards together with her research students and colleagues, including the SIAM Outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award and the IET Circuits, Devices and Systems Premium Award, was selected as one of the 50 most influential women in Israel and in Asia, and is a highly cited Researcher.

She was a Member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is the Editor in Chief of the *Foundations and Trends in Signal Processing*, a Member of the IEEE Sensor Array and Multichannel Technical Committee, and serves on several other IEEE committees. In the past, she was a Signal Processing Society Distinguished Lecturer, a Member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees, and was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal of Signal Processing*, the *SIAM Journal on Matrix Analysis and Applications*, and the*SIAM Journal on Imaging Sciences*. She was the Co-Chair and Technical Co-Chair of several international conferences and workshops.



Muriel Médard (Fellow, IEEE) received three Bachelors degrees in EECS, mathematics, and humanities in 1988 and 1991, respectively. She received her M.S. degree in 1991 and Sc.D in 1995, all from the MIT. She is the Cecil H. Green Professor with Electrical Engineering and Computer Science (EECS) Department, Massachusetts Institute of Technology, Cambridge, MA, USA, and leads the Network Coding and Reliable Communications Group, Research Laboratory for Electronics, Massachusetts Institute of Technology. She was the Editor for many publications of the Institute of Electrical and Electronics Engineers (IEEE), of which she was Elected Fellow, and she was an Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She was the President of the IEEE Information Theory Society in 2012, and was on its board of governors for eleven years. She was a technical program committee co-chair of many of the main conferences in information theory, communications, and networking. She was the recipient of the 2019 Best Paper Award for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, 2009 IEEE Communication Society and Information Theory Society Joint Paper Award, the 2009 William R. Bennett Prize in the Field of Communications Networking, the 2002 IEEE Leon K. Kirchmayer Prize Paper Award, the 2018 ACM SIGCOMM Test of Time Paper Award and several conference paper awards. She was co-winner of the MIT 2004 Harold E. Egerton Faculty Achievement Award was the recipient of the 2013 EECS Graduate Student Association Mentor Award, and served as undergraduate Faculty in Residence for seven years. In 2007, she was named a Gilbreth Lecturer by the U.S. National Academy of Engineering. She was the recipient of the 2016 IEEE Vehicular Technology James Evans Avant Garde Award, the 2017 Aaron Wyner Distinguished Service Award from the IEEE INFORMATION THEORY SOCIETY, and the 2017 IEEE Communications Society Edwin Howard Armstrong Achievement Award. She is a Member of the National Academy of Inventors. She was Elected Member of the National Academy of Engineering for her contributions to the theory and practice of network coding in 2020. She has been elected to American Academy of Arts and Sciences for 2021. For technology transfer, she has co-founded two companies, CodeOn, for which she consults, and Steinwurf, for which she is Chief Scientist.