entropy

MDPI

Article

On Compressed Sensing of Binary Signals for the Unsourced Random Access Channel

Elad Romanov and Or Ordentlich *

The Rachel and Selim Benin School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem 919050, Israel; elad.romanov@mail.huji.ac.il
* Correspondence: or.ordentlich@mail.huji.ac.il

Abstract: Motivated by applications in unsourced random access, this paper develops a novel scheme for the problem of compressed sensing of binary signals. In this problem, the goal is to design a sensing matrix A and a recovery algorithm, such that the sparse binary vector $\mathbf{x}$ can be recovered reliably from the measurements $\mathbf{y} = A\mathbf{x} + \sigma\mathbf{z}$, where $\mathbf{z}$ is additive white Gaussian noise. We propose to design A as a parity check matrix of a low-density parity-check code (LDPC) and to recover $\mathbf{x}$ from the measurements $\mathbf{y}$ using a Markov chain Monte Carlo algorithm, which runs relatively fast due to the sparse structure of A. The performance of our scheme is comparable to state-of-the-art schemes, which use dense sensing matrices, while enjoying the advantages of using a sparse sensing matrix.

Keywords: unsourced random access; compressed sensing; low-density parity-check codes; glauber dynamics

check for updates

Citation: Romanov, E.; Ordentlich, O. On Compressed Sensing of Binary Signals for the Unsourced Random Access Channel. *Entropy* **2021**, *23*, 605. https://doi.org/10.3390/e23050605

Academic Editors: Benjamin M. Zaidel and Ori Shental

Received: 31 March 2021
Accepted: 11 May 2021
Published: 14 May 2021

1. Introduction

The emergence of the Internet of Things (IoT) has motivated much research interest in designing communication protocols for massive machine-to-machine type communication. This type of communication setup is characterized by a large number of users that transmit simultaneously to the same receiver, while each of these users has a very short message to send. In addition, since IoT sensors are often required to be extremely cheap, the transmission scheme must be as simple as possible, and the design objective is to minimize the energy-per-bit, E_b/N_0, under a reliability constraint.

In [1], Polyanskiy defined a communication model capturing the challenges in massive machine-to-machine type communication. In this model, there is an unbounded number of potential users, among which only k are active at each frame. Each active user has a message of B bits to transmit, and transmission takes place over a multiple access channel (MAC). Since the number of users is unbounded, the receiver cannot recover the identities of the active users (as this information has unbounded entropy, assuming all potential users are equally likely to transmit, and the channel has bounded capacity). Thus, the receiver's goal is to recover a list of k messages that contains "most" of the transmitted messages, without identifying the sender of each message. This setup is therefore called *the unsourced random access channel* [2]. The performance of a communication scheme over the unsourced random access channel is assessed by the tradeoff it achieves between energy-per-bit and the per-user probability of error (PUPE), which is the probability that the message transmitted by an active user did not enter the list of messages the receiver outputs.

Over the last few years, there has been great interest in developing efficient low-complexity schemes for the unsourced random access channel [2–15]. A natural approach for this setup is for all users to transmit codewords from the same codebook. It can be easily seen that if $A \in \mathbb{R}^{n\times 2^B}$ is a matrix whose columns are the codewords of this codebook, and $\mathbf{x} \in \{0,1\}^{2^B}$ is a vector whose ith entry equals 1 if one of the active users chose message i and 0 otherwise, the channel output is $\mathbf{y} = A\mathbf{x} + \sigma\mathbf{z}$, where $\mathbf{z}$ is white

Gaussian noise (we assume here for simplicity that no message was chosen by more than one user). Since the number of active users k is typically of the order of tens to hundreds, and is much smaller than 2^B, whereas the blocklength n is typically on the order of 10^4 to 10^5, the problem of designing efficient codebooks and decoding algorithm for the unsourced random access channel corresponds to designing the sensing matrix A and a recovery algorithm for a compressed sensing problem [1]. However, this compressed sensing problem has two non-standard features: (i) the dimensions of the problem are huge (recall that $B = 100$ is a typical number); and (ii) the sparse vector is binary, in contrast to the standard compressed sensing setup where the nonzero entries can take values in an interval within the real line.

To address the dimensions of the compressed sensing problem, Amalladinne et al. [7] introduced the coded compressed sensing framework, where the B message bits are divided to smaller chunks and are encoded on different sub-blocks. This idea breaks the original compressed sensing problem into a sequence of compressed sensing problems with manageable dimensions, which can be handled via existing tools from the compressed sensing literature (we remark that this is somewhat related to ideas that have previously appeared in the compressed sensing and group testing literature, wherein one constructs the measurements matrix by combining an "outer" and "inner" code—see, e.g., [16–18]). A difficulty that arises under this framework is that the sub-messages eventually have to be stitched to one long message, and a tree code was developed by Amalladinne et al. [7] for this purpose. While there has been many important advances in the field since the first appearance of the coded compressed sensing framework [19], the idea of first solving small compressed sensing problems and then leveraging the solutions to obtain estimates of the entire message still appears in one way or another in practically all schemes achieving state of the art performance.

Motivated by the above, the focus of this paper is the design of sensing matrices and efficient decoding algorithms for (small dimensions) compressed sensing of binary signals. Originally, Amalladinne et al. [7] treated this challenge by designing the sensing matrix based on BCH codes and using off-the-shelf recovery algorithms, such as LASSO or non-negative least squares (NNLS), for decoding. The main weakness of this approach is that it fails to exploit the fact that the entries are binary. Later, Fengler, Jung and Caire [9] suggested using Sparse regression codes with approximate message passing (AMP) decoding. The main benefit of the AMP decoder is that it allows incorporating any prior one has on the signal $\mathbf{x}$, and not just sparsity. Consequently, it achieves excellent performance when $\mathbf{x}$ is a binary sparse vector and the sensing matrix A is i.i.d. Gaussian. This framework has more benefits, for example it allows one to efficiently jointly decode all compressed sensing problems corresponding to the different sub-blocks and one can even iterate between the AMP decoder and the tree decoder [11].

In this paper, we propose an alternative design for a sensing matrix A and a decoding algorithm. Our sensing matrix A is taken as the parity check matrix of a low-density parity check (LDPC) code, thought of as a matrix over the reals. The decoder is based on the Markov Chain Monte Carlo (MCMC) method, more specifically, Glauber dynamics. This method performs a random walk over a Markov chain whose state space consists of all possible values of $\boldsymbol{x}$ and whose stationary distribution is the conditional probability of $\boldsymbol{x}$ given the measurement $\boldsymbol{y}$. Due to the sparse structure of the matrix A, each step in the random walk can be simulated with a low computational cost.

For the compressed sensing problem with binary signals problem, our proposed framework achieves comparable performance to that of AMP with a Gaussian sensing matrix. However, in contrast to the AMP framework, which is based on sensing matrices that are Gaussian i.i.d., or "Gaussian i.i.d.-like", our sensing matrix is sparse. The sparsity of the sensing matrix A in compressed sensing of binary signals has several benefits that go beyond the unsourced random access application:

- **Storage.** Storing a sparse matrix requires fewer memory resources than storing a dense unstructured matrix, such as a matrix sampled from the i.i.d. Gaussian ensemble.

We remark, however, that the AMP algorithm often works very well for compressed sensing of binary signals even when the Gaussian i.i.d. matrix A is replaced with a sensing matrix that is dense yet easy to store. For example, Amalladinne et al. [20] suggested taking A as a sub-sampled Hadamard matrix.

- **Joint source-channel coding with local updates.** Consider the problem of storing a sparse binary vector $\mathbf{x} \in \{0,1\}^M$ with Hamming weight at most k, in an array of n noisy memory cells. By noisy memory cells, we mean that the value read from memory cell i is modelled as $s_i + z_i$, where s_i is the stored value and z_i is additive noise, e.g. Gaussian. This is a reasonable model for magnetic recording (ignoring intersymbol interference) [21] and for flash memories (ignoring further impairments like cross talk) [22]. Note that this is actually a joint-source channel coding problem where the source is $\mathbf{x} \in \{0,1\}^M$, the channel is Gaussian and can be used n times, and the distortion measure is Hamming distortion. It is often desirable to use update efficient schemes. In such schemes changing one bit in the input vector $\mathbf{x}$, should correspond to changing the content of a small number of memory cells (see, e.g., [23]). When the encoding scheme is $\mathbf{s} = A\mathbf{x}$, an update in one coordinate of $\mathbf{x}$, say x_i, corresponds to adding (removing) the ith column of A to (from) $\mathbf{s}$. If each column has a small number of nonzero entries, the update involves changing the stored value in a small number of cells. Thus, using a matrix A with sparse columns is highly desirable.
- **Group testing.** In group testing, the goal is to detect a set of at most k defective items from M possible items. To this end, we designate by $\mathbf{x} \in \{0,1\}^M$ the vector whose nonzero entries are defective. We have n measurements of $\mathbf{x}$, each corresponding to a different "pool". Each pool is a subset of $[n]$, and the corresponding measurement is obtained by passing the number of defective items in the pool, denoted by ℓ, through some noisy channel $P_{\mathbf{Y}|L}(\mathbf{y}|\ell)$ (see, Definitions 3.1 and 3.3 in [24]). The typical case is that the channel depends on the number of defective items ℓ only through the indicator on the event $\{\ell > 0\}$, but the general model allows the measurement to be distributed as $\ell + \sigma z$, where $z \sim \mathcal{N}(0,1)$. Thus, with this model the design of the group testing scheme corresponds to designing a binary sensing matrix $A \in \{0,1\}^{n\times M}$, and the measurements are $\mathbf{y} = A\mathbf{x} + \sigma\mathbf{z}$. Using pools, corresponding to the rows of A, with small Hamming weight, results in simpler tests. For example, the original application for which the group testing framework was developed was detection of syphilis among a large group of patients, using a small number of tests. Using pools with small Hamming weight means that we need to mix samples from fewer patients in each pool, which results in less work for the lab technician.

In Section 2, we formalize the problem of compresses sensing of binary signals and present our suggested construction for the sparse sensing matrix and our MCMC-based recovery algorithm. Some theoretical analysis and justification for our suggested method is also given. In Section 3, we evaluate the performance of the proposed scheme numerically and compare it to other state-of-the art schemes for the compressed sensing of binary signals problem. We also evaluate the performance of an end-to-end communication scheme for the unsourced random access channel with a small amount of feedback, which uses the proposed compressed sensing of binary signals scheme as an important ingredient. Section 4 is devoted to conclusion and additional discussion.

2. Compressed Sensing of Binary Signals

We now define a formal mathematical model for the problem studied in this paper. Consider a linear inverse problem of the form

$$\boldsymbol{y} = A\boldsymbol{x} + \sigma\boldsymbol{z}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^M$ is an unknown signal, to be recovered; $A \in \mathbb{R}^{n\times M}$ is a (known) linear measurement matrix; and $\boldsymbol{z} \in \mathbb{R}^n$ is i.i.d. Gaussian noise: $z_1, \ldots, z_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$. This problem becomes especially interesting in the *under-determined regime*, where the number of samples

n is smaller than the signal dimension M—here, clearly, one cannot recover $\boldsymbol{x}$ generically, and it is necessary to make additional structural assumptions on $\boldsymbol{x}$. In compressed sensing, one assumes that $\boldsymbol{x}$ is a sparse vector, where the number of non-zero entries k is very small compared to M. Perhaps the most fundamental result in sparse recovery states that, in order to recover exactly any k-sparse $\boldsymbol{x}$ from noiseless measurements $\boldsymbol{y} = A\boldsymbol{x}$, one in fact needs only $n = O(k \log(M/k))$ linear measurements, where the sensing matrix A is taken to be an i.i.d. Gaussian random matrix (see, e.g., [25], Chapter 9). The recovery procedure itself, while not linear, can be formulated as a convex program which is computationally easy to solve. In recent years, a vast literature on compressed sensing has formed, spanning new theory, low-complexity algorithms and new constructions of good sensing matrices, beyond the i.i.d. Gaussian setup. We make no pretense to give a literature review on this topic; for a starting point, we refer primarily to surveys [25–34].

We consider a setting where $\boldsymbol{x}$ is constrained to be in a discrete set, on top of being sparse. Specifically, we assume it is binary: $\boldsymbol{x} \in \{0,1\}^M$. As described above, this problem is closely related to communication over the unsourced random access channel, but problems of this form have received some attention in the past (see, e.g., [35–38]).

Throughout, we assume a sparse binary prior for $\boldsymbol{x}$. Specifically, let k be the expected sparsity, and denote $\rho = k/M$. The coordinates of $\boldsymbol{x}$ are assumed i.i.d. Bernoulli random variables:

$$x_1, \ldots, x_M \overset{i.i.d.}{\sim} \text{Bernoulli}(\rho), \tag{2}$$

that is, $\Pr(x_i = 1) = \rho$ and otherwise $x_i = 0$. Clearly, the expected number of non-zero entries is just $\mathbb{E}\|\boldsymbol{x}\|_0 = k$. A recovery algorithm for $\mathbf{x}$ from $\mathbf{y}$ is a mapping $\hat{\mathbf{x}} : \mathbb{R}^n \to \{0,1\}^M$. The performance of a recovery algorithm is measured in terms of the bit error rate (BER) it attains

$$\text{BER}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{k} \sum_{i=1}^{M} \Pr(x_i \neq \hat{x}_i(\boldsymbol{y})), \tag{3}$$

where the probability is taken with respect to both the additive noise, as described in (1), and the signal prior (2). Note that the normalization in (3) is by the expected sparsity k, rather than by the length M of the vector $\mathbf{x}$. Since typically $k \ll M$ in compressed sensing, normalizing by M would yield a very small BER for any reasonable estimator, and normalizing by k therefore makes more sense.

Given a signal dimension M and budget of measurements n, one would typically like to: (i) construct "good" sensing matrices A that allow for noise-robust recovery of signals with as little sparsity (large k) as possible; and (ii) come up with low-complexity recovery algorithms for recovering $\boldsymbol{x}$ from $\boldsymbol{y}$. As for Point (ii), note that one would like to go beyond off-the-shelf compressed sensing algorithms, such as the LASSO [39–41] or Non-Negative Least Squares (NNLS) [7,26,42–46], which are designed with any real or positive signal in mind, and find algorithms that explicitly leverage the binary structure of the signal, to attain an advantage in terms of recovery performance. In this paper, we address these two points: for the sensing matrix, we propose to use sparse matrices based on LDPC codes; as for the recovery algorithm, we propose to use an MCMC sampling method that approximates the optimal (in terms of bit error probability) MAP estimator.

2.1. Sensing Matrices from LDPC Codes

We consider sensing matrices based on Gallager's ensemble of LDPC codes [47]. Denote by $\mathbf{LDPC}(\nu, s; M, n)$ the following ensemble of random bipartite and biregular graphs, described below:

- One side of the graph has M vertices, which we call "variables" (the left side), and the other has n vertices, called "factors" (the right side).

- For simplicity, assume $\nu M = sn$. Each variable has degree ν, meaning it is connected to exactly ν factors; each factor has degree s. Thus, there are exactly $\nu M = sn$ edges in the graph.
- The edges of $\mathcal{G} \sim \mathbf{LDPC}(\nu, s; M, n)$ are sampled according to the following procedure. The procedure runs in ν rounds, so that in every round one introduces M/s new factors (we assume M/s is integer for simplicity), by randomly partitioning the variables $[n]$ into M/s parts of size s each, namely,

  $$[n] = \bigcup_{i=1}^{M/s} S_i\,, \quad S_i \cap S_j = \emptyset, \quad |S_i| = s \quad \text{for all } 1 \le i, j \le n\,, i \ne j\,.$$

  For every new factor $1 \le i \le M/s$ introduced in this round, one adds an edge between i and all the variables in the corresponding S_i.

The sensing matrix $A \in \{0,1\}^{n \times M}$ is taken to be the adjacency matrix of a randomly sampled graph $\mathcal{G} \sim \mathbf{LDPC}(\nu, s; M, n)$, that is,

$$A_{i,j} = A(\mathcal{G})_{i,j} = \begin{cases} 1 & \text{there is an edge in } \mathcal{G} \text{ between factor } i \text{ and variable } j \\ 0 & \text{otherwise} \end{cases}.$$

The idea of constructing sensing matrices from bipartite graphs is not new. It is known that when $\mathcal{G}$ is a sufficiently good expander, the corresponding adjacency matrix A is a good sensing matrix (see, e.g., [48–51], ([25], Chapter 13) and the references therein). Specifically, ensembles of LDPC codes have also been considered previously for compressed sensing [52–54].

It is worthwhile to recall at this point that the recovery problem we consider here is more structured than the "standard" compressed sensing setup: on top of being sparse, we assume the unknown signal is *binary*, and in particular *non-negative*. Past results show that the non-negativity assumption may give a considerable advantage in terms of the required number of measurements, as well as robustness to noise (see, e.g., [26,42–46,55]).

We especially mention the results of Khajehnejad et al. [55]. We say that a bipartite graph with left degree ν is an (r, ε)-expander if for every set $|S| \le r$ of left vertices, one has $|N(S)| \ge (1-\varepsilon)\nu|S|$, $N(S)$ being the neighbors of vertices in S. The results of Khajehnejad et al. [55] state that a bipartite left-regular $(r, 1-1/\nu)$-expander yields, after applying a very small perturbation to the entries of the adjacency matrix, a sensing matrix where all **non-negative** $\lceil r/\nu - 1 \rceil$ sparse vectors $\boldsymbol{x}$ can be recovered from $\boldsymbol{y} = A\boldsymbol{x}$ (noiseless measurements). This guarantee, for non-negative signals, is *considerably* better than what one has without the non-negativity constraint—to get recoverability guarantees for "general" compressed sensing, one needs considerably larger expansion (smaller ε) (see, e.g., [25], Chapter 13). There are well-known connections between the decodability of LDPC codes and their expansion properties [56]. For example, for a slightly different ensemble of LDPC codes (that contains Gallager's ensemble), one can show ([56], Theorem 8.7) (this result first appeared in [57]) that, with high probability, a random graph is an $(\alpha^* M, 1-1/\nu)$-expander, where α^* is the positive solution of

$$\frac{\nu-1}{\nu} h_2(\alpha) - \frac{1}{s} h_2\left(\alpha \frac{s}{\nu}\right) - \alpha \frac{s}{\nu} h_2\left(\frac{\nu}{s}\right) = 0\,,$$

and $h_2(p) = -p\log(p) - (1-p)\log(1-p)$ is the binary entropy function. While not precisely applicable for our setup (which uses Gallager's ensemble), the following calculation could nonetheless be thought of as a crude heuristic. For example, in the setup, we consider later on in the numerical experiments, corresponding to a typical use-case for detection in unsourced random access, $M = 2^{14}$, $n = 2^{11}$, $\nu = 16$, $s = 128$, one can solve the above equation numerically and get $\alpha^* \approx 0.993$. Together with Khajehnejad et al. [55], this hints that $k = \alpha^* M/'\nu \approx 101$ sparse, non-negative signals can be consistently recovered. In fact, the experiments indicate that, practically, binary signals with considerably more non-zeros can be recovered reliably in this setting (see Section 3.1).

2.2. MCMC Algorithm for Recovery

Recall that, for a given sensing matrix A, our goal is to construct an estimator $\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{x}}(\boldsymbol{y})$ such as to minimize the per-bit error rate (BER), as defined in (3). Clearly, the optimal estimator in the sense of minimizing the BER is simply the per-coordinate maximum a posteriori (MAP) estimator:

$$\widehat{x}_{\mathrm{BER},i} = \operatorname*{argmax}_{\widehat{x}\in\{0,1\}} \Pr\big(x_i = \widehat{x}\,\big|\,\boldsymbol{y}\big)\,, \quad \text{for all } 1 \le i \le M\,. \tag{4}$$

Computing the posterior $\Pr(x_i \,|\, \boldsymbol{y})$ is a formidable task: it requires one to marginalize over all other coordinates $\ell \neq i$. From a computational point of view, this is highly nontrivial, since the coupling between the coordinates of $\boldsymbol{x}$, as induced by A, creates a strong cross-coordinate dependency conditioned on $\boldsymbol{y}$.

We propose to mitigate this difficulty by **sampling**. Instead of marginalizing and and maximizing, we sample an $\widehat{\boldsymbol{x}} \in \{0,1\}^M$ from the full posterior, given by

$$\Pr\big(\boldsymbol{x} = \widehat{\boldsymbol{x}}\,\big|\,\boldsymbol{y}\big) = \frac{1}{Z}\exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\widehat{\boldsymbol{x}}\|^2 + \lambda\|\widehat{\boldsymbol{x}}\|_1\right\}, \tag{5}$$

where $\lambda = \log\frac{\rho}{1-\rho}$, $\rho = k/M$ and Z is the partition function (to see the correctness of $\lambda = \log\frac{\rho}{1-\rho}$ in the expression above, note that the prior is $\Pr(x_i = 1) = e^{\log\rho}$ and $\Pr(x_i = 0) = e^{\log(1-\rho)}$; in other words, $\Pr(x_i = \hat{x}_i) = e^{\hat{x}_i\log\rho + (1-\hat{x}_i)\log(1-\rho)} = e^{\lambda\hat{x}_i + \log(1-\rho)} \propto e^{\lambda\hat{x}_i}$). Taking the ith coordinate of $\widehat{\boldsymbol{x}}$, call it $\widehat{x}_i$, we obtain a sample from $\Pr(x_i = \cdot|\boldsymbol{y})$, the desired single-bit posterior distribution.

Intuition suggests that, when $\widehat{x}_{\mathrm{BER},i}$ has small error, the estimator obtained by sampling, call it $\widehat{x}_{\mathrm{SAMP},i}$, should have small error as well. This is because, if the optimal error is small, the posterior $\Pr(x_i|\boldsymbol{y})$ must put most of its mass on $\widehat{x}_{\mathrm{BER},i} = \widehat{x}_{\mathrm{BER},i}(\boldsymbol{y})$; this in turn means that, with high probability over the sampling procedure, one should in fact get $\widehat{x}_{\mathrm{SAMP},i} = \widehat{x}_{\mathrm{BER},i}$. This reasoning is formalized in the following Lemma:

Lemma 1. *Denote $\widehat{\boldsymbol{x}}_{\mathrm{BER}} = (\widehat{x}_{\mathrm{BER},1}, \ldots, \widehat{x}_{\mathrm{BER},M})$, with coordinates given by Equation (4). Let $\widehat{\boldsymbol{x}}_{\mathrm{SAMP}} = \widehat{\boldsymbol{x}}_{\mathrm{SAMP}}(\boldsymbol{y}) \sim \Pr(\cdot\,|\,\boldsymbol{y})$ be a random sample from the posterior (5). Then:*

$$\mathrm{BER}(\boldsymbol{x}, \widehat{\boldsymbol{x}}_{\mathrm{SAMP}}) \le 2 \cdot \mathrm{BER}(\boldsymbol{x}, \widehat{\boldsymbol{x}}_{\mathrm{BER}})$$

In other words, the bit error rate of $\widehat{\boldsymbol{x}}_{\mathrm{SAMP}}$ is bounded by twice the optimal bit error rate, over all estimators.

Note that, on the left-hand side, the probability is taken both over the randomness in $\boldsymbol{x}$ and the noise, as well as the sampling procedure used for constructing $\widehat{\boldsymbol{x}}_{\mathrm{SAMP}}$.

Several variants of Lemma 1 have been proved in the past (see, e.g., [58–60]). For completeness, we provide a short proof in Appendix A.1.

Thus, we are left with the problem of sampling from the posterior $\Pr(\boldsymbol{x}\,|\,\boldsymbol{y})$—doing so "directly" might seem, at first glance, essentially just as hard as maximizing the posterior (namely, we would need to go over all 2^M possible signal configurations). Markov-Chain Monte Carlo (MCMC) methods provide a strong toolbox for sampling, *approximately*, from high-dimensional distributions. The idea is to construct an ergodic Markov chain such that: (i) its stationary distribution is the desired (high-dimensional) distribution one would like to sample from, namely $\Pr(\boldsymbol{x}\,|\,\boldsymbol{y})$; and (ii) the chain is easy to propagate in time (e.g., its update rule is local). Having constructed such a chain, and assuming that it mixes sufficiently fast (which is often difficult to ensure), one can therefore efficiently sample from the desired distribution, up to high precision. For further background and discussion on MCMC, we refer to the work of Levin and Peres [61] (Chapter 3). The use of MCMC methods for solving inverse problems in signal processing and for decoding/detection in communication is by no means novel (see, e.g., [62–69]). While both the idea of using LDPC codes as sensing matrices and the idea of using MCMC methods for decoding are not new, our innovation here is in combining the two concepts for the compressed sensing of binary signals problem. As becomes evident below, the sparse structure of the sensing matrix constructed from an LDPC code significantly reduces the computational load from the MCMC decoder by reducing the computational cost of each iteration.

We propose using the well-known *Gibbs sampling* method, also known as *Glauber dynamics*, which is a general-purpose recipe for sampling from high-dimensional distributions. Let $\mathbb{Q}(\boldsymbol{x})$ be a distribution over $\{0,1\}^M$ from which one wants to sample; in our case, of course, $\mathbb{Q}(\boldsymbol{x}) = \Pr(\boldsymbol{x} \,|\, \boldsymbol{y})$. We construct a chain $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots \in \{0,1\}^M$ starting from some (arbitrary) initial state $\boldsymbol{x}^{(0)}$ according to the following transition rule. Suppose that the current state is $\boldsymbol{x}^{(t)}$; one samples a coordinate to update at random, $i_t \sim \mathrm{Uniform}(\{1,\ldots,M\})$, so that $x_j^{(t+1)} = x_j^{(t)}$ for all $j \neq i_t$. As for coordinate i_t, it is sampled according to the conditional distribution of x_{i_t}, with all other coordinates fixed and given by $\boldsymbol{x}^{(t)}$, that is: $x_{i_t}^{(t+1)} \sim \mathbb{Q}(x_{i_t} \,|\, \boldsymbol{x}_{\sim i_t} = \boldsymbol{x}_{\sim i_t}^{(t)})$ (we denote the vector of all coordinates, except for i_t, by $\boldsymbol{x}_{\sim i_t}$).

Applied to the posterior in (5), Glauber dynamics reads as follows:

It is easy to see that the process $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots \in \{0,1\}^M$ is an ergodic Markov chain, and therefore has a unique stationary distribution. Furthermore, it is easy to verify that $\mathbb{Q}(\boldsymbol{x})$ is a stationary distribution of this chain. Thus, for T sufficiently large, we have that indeed $\mathbf{x}^{(T)}$ is distributed as a random sample from $\mathbb{Q}(\boldsymbol{x})$. Note that, when A is a sparse LDPC matrix, each iteration of Algorithm 1 is computationally very cheap. One can easily keep track of $\boldsymbol{y}^{(t)} = A\boldsymbol{x}^{(t)}$ and $\|\boldsymbol{y} - \boldsymbol{y}^{(t)}\|^2$ across iterations, noting that an update to a coordinate of $\boldsymbol{x}^{(t)}$ requires updating only ν coordinates of $\boldsymbol{y}^{(t)}$, where ν is the degree of a variable in A. Thus, the computational complexity of Algorithm 1 is $O(T\nu)$, where a typical choice of T should be $T = O(M \log M)$ (see Lemma 2 below).

Algorithm 1: Glauber dynamics for binary compressed sensing.

```
Input: $T$ = number of steps to run; $\boldsymbol{x}^{(0)} \in \{0,1\}^M$ = initial state; $\boldsymbol{y} \in \mathbb{R}^n$ =
        measurements;
parameters $\sigma^2 > 0$, $\lambda \in \mathbb{R}$.
for $t = 1, \ldots, T$ do
    $i_t \sim \mathrm{Uniform}(\{1,\ldots,M\})$; new coordinate to update
    Let $q_0^{(t)} = \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\boldsymbol{x}_{i_t=0}^{(t)}\|^2\right\}$, $q_1^{(t)} = \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\boldsymbol{x}_{i_t=1}^{(t)}\|^2 + \lambda\right\}$
        Update:
        $x_{i_t}^{(t+1)} = 1$ w.p. $p_1^{(t)} = \frac{q_1^{(t)}}{q_0^{(t)} + q_1^{(t)}}$; otherwise $x_{i_t}^{(t+1)} = 0$
        $\boldsymbol{x}_{\sim i_t}^{(t+1)} = \boldsymbol{x}_{\sim i_t}^{(t)}$
    (Note: Above, $\boldsymbol{x}_{i_t=0}^{(t)}, \boldsymbol{x}_{i_t=1}^{(t)}$ stand for setting, in $\boldsymbol{x}^{(t)}$, the $i_t$th coordinate to $0,1$
    respectively).
end
return $\boldsymbol{x}^{(T)}$
```

We can give the following guarantee for the mixing time of Glauber dynamics:

Lemma 2 (Fast mixing for Glauber dynamics)**.** *Let* $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x}^{(0)} \in \{0,1\}^M$, $\sigma^2 > 0$ *and* $\lambda \in \mathbb{R}$ *be any parameters. Denote the following distribution* $\mathbb{Q}(\cdot)$ *on the cube* $\{0,1\}^M$ *by*

$$\mathbb{Q}^{(\infty)}(\boldsymbol{x}) = \mathbb{Q}^{(\infty)}_{\boldsymbol{y},\sigma^2,\lambda}(\boldsymbol{x}) = \frac{1}{Z}\exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_1\right\},$$

where Z is the partition function. Denote by $\mathbb{Q}^{(T)}(\cdot) = \mathbb{Q}^{(T)}_{\boldsymbol{y},\sigma^2,\lambda,\boldsymbol{x}^{(0)}}(\cdot)$ the distribution of $\boldsymbol{x}^{(T)}$, the state returned after running Algorithm 1 for T steps. Suppose that

$$4\sigma^2 > \nu(s-1)\,, \tag{6}$$

where ν and s are, respectively, the variable and factors degrees in A.

Let $\epsilon > 0$ let be a target precision. Then, for any $T \geq (\log(1/\varepsilon) + \log(M)) \cdot \frac{4\sigma^2 M}{4\sigma^2 - \nu(s-1)} = \Theta(M \log M)$ (here the $\Theta(\cdot)$ notation suppresses a dependence on $\varepsilon, \sigma^2, \nu, s$), one has

$$\mathbf{d_{TV}}\left(\mathbb{Q}^{(T)}, \mathbb{Q}^{(\infty)}\right) \leq \varepsilon\,,$$

where $\mathbf{d_{TV}}(\cdot,\cdot)$ stands for total variation (statistical) distance.

A proof is given in Appendix A.2.

Note that Lemma 2 applies for *any* $\boldsymbol{y} \in \mathbb{R}^n$ and σ^2, that do not necessary have anything to do with the model (1). However, when $\boldsymbol{y}, \sigma^2$ do correspond to measurements from (1), namely $\boldsymbol{y} = A\boldsymbol{x} + \sigma\boldsymbol{z}$, Lemma 2, combined with Lemma 1, allows us to bound the bit error rate of the estimator $\boldsymbol{x} = \boldsymbol{x}^{(T)}$ returned by running T iterations of Glauber dynamics. Assuming condition (6) holds, Lemma 2 tell us that running $T = \frac{4(c+1)\sigma^2}{4\sigma^2 - 4\nu(s-1)} \cdot M \log M = O(M \log M)$ iterations of Glauber dynamics gives, with probability one (over $\boldsymbol{x}, \boldsymbol{y}$) an output $\boldsymbol{x}^{(T)}$ whose law is M^{-c}-close to the law of $\widehat{\boldsymbol{x}}_{SAMP}$, in total variation distance—here, $c > 0$ can be taken as large as one likes. Recall that total variation distance is just $\mathbf{d_{TV}}(\mathbb{P}, \mathbb{Q}) = \max_{0 \leq f \leq 1}\{\mathbb{E}_{\widehat{\boldsymbol{x}} \sim \mathbb{P}}[f(\widehat{\boldsymbol{x}})] - \mathbb{E}_{\widehat{\boldsymbol{x}} \sim \mathbb{Q}}[f(\widehat{\boldsymbol{x}})]\}$; maximization here is done over all bounded functions $f : \{0,1\}^M \to [0,1]$ (recall that the maximum is actually attained at the indicator function $f = \mathbf{1}_S$, where $S = \{x : \mathbb{P}(x)/\mathbb{Q}(x) \geq 1\}$). Plugging $f(\widehat{\boldsymbol{x}}) = \frac{1}{k}\sum_{i=1}^{M} \mathbf{1}_{x_i \neq \widehat{x}_i}$ and noting that f is non-negative and bounded by M/k, we deduce

$$\begin{aligned} \mathrm{BER}(\boldsymbol{x}, \boldsymbol{x}^{(T)}) &\leq \mathrm{BER}(\boldsymbol{x}, \widehat{\boldsymbol{x}}_{\mathrm{SAMP}}) + \frac{M}{k}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\mathbf{d_{TV}}(\mathbb{P}_{\widehat{\boldsymbol{x}}^{(t)}}, \mathbb{P}_{\widehat{\boldsymbol{x}}_{\mathrm{SAMP}}}) \\ &\leq \mathrm{BER}(\boldsymbol{x}, \widehat{\boldsymbol{x}}_{\mathrm{SAMP}}) + M^{-c+1}/k\,, \end{aligned}$$

which, by Lemma 1, is bounded by twice the optimum BER, up to an inverse polynomial (in M) error.

As a remark, we mention that in practice, MCMC methods are often implemented using *annealing*, which in our case amounts to basically running Glauber dynamics with a noise variance σ^2 which is larger than the true noise. This can help steer the system away from local maxima of $\mathbb{Q}$, by "smoothing" it out.

We would like to emphasize that condition (6) is very pessimistic, and in practice Glauber dynamics appears to mix rapidly at substantially lower noise levels than predicted there. For example, in the setup we consider below, $M = 2^{14}$, $n = 2^{11}$, $\nu = 16$, $s = 128$, so that the bound $\sigma_0^2 = \nu(s-1)/4 = 508$, translates into energy per transmitted bit as $E_b/N_0 = \frac{\nu}{2\sigma^2 \cdot \lg_2(M) \cdot} = \frac{1}{7(s-1)} \approx 0.001$, which is roughly -29.5 dB. This E_b/N_0 is very far from sufficient for reliable recovery of $\widehat{x}$ even when k is small (see experiments in Section 3.1). In this regime, while indeed Lemma 2 holds in the sense that Glauber dynamics mixes fast, the error rate of the optimal estimator is too high to be of use. Thus, Lemma 2 should **not** be thought of as an accurate predictor for the performance of Glauber dynamics for binary compressed sensing. Instead, it should be thought of as a "sanity check"—evidence that Glauber dynamics is a reasonable thing to do, at least in *some* regime of the problem.

On the same note, we observed, that when k is large, Glauber dynamics tends sometimes to get stuck at "bad" local maxima, even when the noise is moderate. To mitigate this, one can initialize $\boldsymbol{x}^{(0)}$ *reasonably close* to the true signal $\boldsymbol{x}$, using an off-the-shelf compressed sensing solver like NNLS—and then use Glauber dynamics as a *refinement step*. Applying this additional step of Glauber dynamics may improve the performance *substantially* (see the numerical results in Section 3.1). Of course, the result of Lemma 2 does not predict in any way this behavior; rather, it is completely agnostic to the starting location. Additionally, the bound on the mixing time there does not depend at all on k, which, as just mentioned, is crucial for the behavior of Glauber dynamics in practical regimes. A more sophisticated analysis of Glauber dynamics for compressed sensing of binary signals, which takes into account the points above, is an interesting problem, and, to the best of our judgement, highly nontrivial.

3. Simulation Results

3.1. Performance in Compressed Sensing of Binary Signals

We start by demonstrating the performance of Glauber dynamics in the compressed sensing of binary signals setup of Section 2.

We run many random recovery experiments, to recover $\boldsymbol{x} \in \{0,1\}^M$ from $\boldsymbol{y} = A\boldsymbol{x} + \sigma \boldsymbol{z} \in \mathbb{R}^n$. In all the experiments, we use $M = 2^J$, $J = 14$, $n = 2^{11}$ and sparsity values $k \in \{50, 100, 200, 300\}$. These parameters are representative of a typical setup for unsourced random access (see Section 3.2). For each k, we vary the energy per transmitted bit, $E_b/N_0 = \frac{E_m}{2\sigma^2 \cdot J}$ (here, E_m is the average energy per transmitting user—the energy of a column of A) and plot the corresponding bit error rate.

We plot the performance under the following schemes:

1. The scheme of Amalladinne et al. [7]: A based on BCH codes, and NNLS decoder. To obtain a binary estimator from the NNLS solution, we simply assign every entry to its closest binary value (that is, according to whether it is smaller or greater than $1/2$).
2. A given by a sparse LDPC matrix, with parameters $\nu = 16$ (consequently, $s = 128$), under the following decoding algorithms:
   (a) NNLS.
   (b) Glauber dynamics with initialization at $\boldsymbol{x}^{(0)} = \mathbf{0}$.
   (c) Glauber dynamics, with $\boldsymbol{x}^{(0)}$ initialized at the NNLS solution.

   When using Glauber dynamics, we always let it run for $T = 10M \lg_2 M = 10\,MJ$ iterations.
3. A is a dense random i.i.d. Gaussian matrix of mean 0 and variance $1/n$, with Approximate Message Passing (AMP) decoder (thus, $E_m = 1$; of course, in the experiments, the noise level σ is normalized according to the appropriate choice of E_b/N_0). The denoiser used in AMP is the optimal denoiser for the i.i.d. Bernoulli source, essentially as proposed by Fengler et al. [70]. AMP is a state-of-the-art algorithm for compressed sensing of binary signals, and is our main benchmark. For convenience, the exact implementation details of AMP are given in Appendix B.

Our results are summarized in Figure 1. We see that when the sparsity is moderate (up to $k = 200$), our proposed scheme attains essentially state-of-the-art performance. However, when k is large ($k = 300$), performance falls short of AMP: if initialized at zero, Glauber dynamics consistently gets stuck in a local maximum, far away from the true signal; on the other hand, if one initializes Glauber dynamics with the NNLS solution, the combined scheme eventually attains performance which is substantially better than off-the-self compressed sensing solvers.

In Figure 2, we plot the evolution, across consecutive iterations, of both the BER and the "energy" $E(\boldsymbol{x}^{(t)}) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\boldsymbol{x}^{(t)}\|^2 + \lambda\|\boldsymbol{x}^{(t)}\|_1$ along a single run of Glauber dynamics (initialized at $\boldsymbol{x}^{(0)} = 0$). We use $k = 100$ and $E_b/N_0 = 1\,\text{dB}$. Note that the iterations are given in units of $MJ = M \lg_2 M$ (meaning, it is t/MJ). Ignoring stochastic fluctuations, we see that Glauber dynamics essentially monotonically minimizes the energy (the error, however, is not monotonically decreasing).

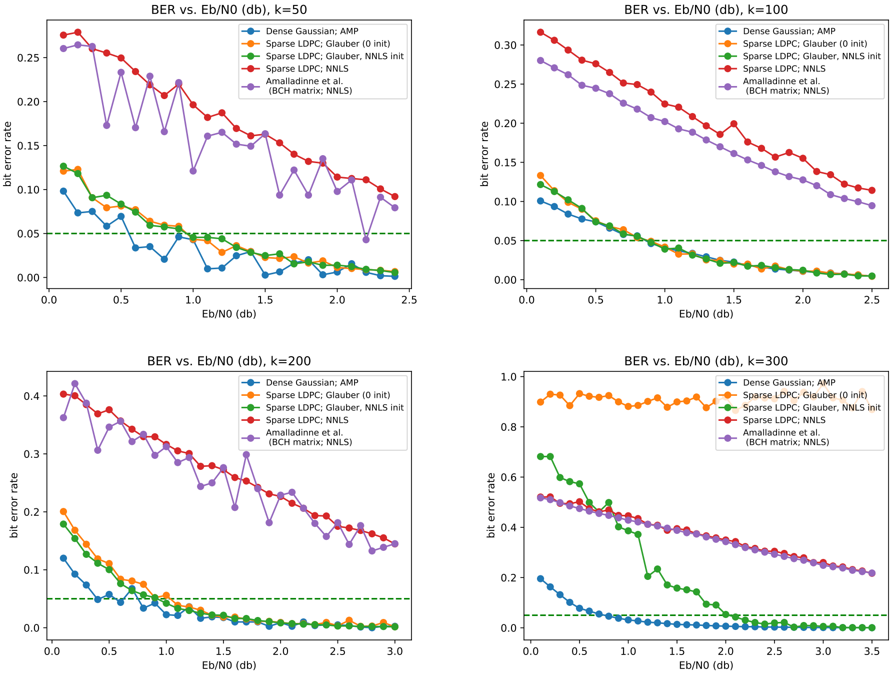

Figure 1. BER vs. E_b/N_0 for several sparsity levels k. When k is small to moderate, our proposal achieves state-of-the-art performance, on par with AMP on a dense Gaussian matrix. Each point on a curve is the average BER over a 100 random experiments. Dashed horizontal line: BER $= 0.05$.

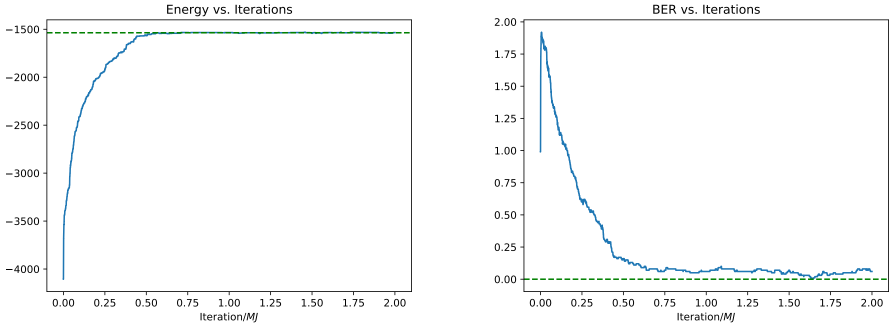

Figure 2. Energy and error along a typical trajectory of Glauber dynamics, with $k = 100$ and $E_b/N_0 = 1.0\,\text{dB}$. The dashed horizontal curve correspond to the energy and error respectively of the true signal $\mathbf{x}$.

3.2. End-to-End Performance in Grant-Based Random Access

As mentioned in the Introduction, the compressed sensing of binary signals problem is an important component of many schemes that have been proposed for communication over the unsourced random access channel. In this model [1], communication is performed in blocks of n channel uses of a Gaussian multiple access channel

$$\boldsymbol{y} = \sum_{i=1}^{K_{\text{tot}}} s_i \boldsymbol{x}_i + \sigma \boldsymbol{z},$$

where $(s_1, \ldots, s_{K_{\text{tot}}}) \in \{0,1\}^{K_{\text{tot}}}$ is the "activity pattern" vector whose Hamming weight is k, $\boldsymbol{x}_i \in \mathbb{R}^n$ is the codeword transmitted by user i assuming it was active and $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is additive white Gaussian noise (AWGN). Note that this channel model implicitly assumes perfect power and phase control, which is often difficult to attain in practice. We further assume that all active users have a message of B bits to transmit, and that each of these messages is independently and uniformly distributed over $[2^B]$. The activity pattern is assumed *unknown* to the decoder, and known only locally to the transmitters, i.e., each user only knows whether or not it is active, but does not know which of the other users are active. The decoder's goal is to output a list of k messages that contains as many transmitted messages as possible. The per-user probability of error (PUPE) is defined as the number of transmitted messages that did not enter the list, normalized by k.

In this section, we use the scheme developed above for compressed sensing of binary signals as a building block for an end-to-end communication scheme for the unsourced random access channel. We slightly deviate from the mainstream literature on unsourced random access, by allowing for some feedback to be sent from the receiver to all potential users through a broadcast channel. This option was mostly avoided until now, with the exception of Facenda and Silva [12], as it was believed that the large number of potential users and the small payloads for each active users renders scheduling too wasteful. Recently, Kang and Yu [71] established a connection between scheduling for the unsourced random access channel and *perfect hashing* and demonstrated that in fact scheduling for the unsourced random access channel can be attained with a very small cost. Based on their observation, we propose the following scheme for the unsourced random access channel with an unbounded number K_{tot} of potential users, among which k are active users that have to send a B bits message each, over n channel uses:

- **Phase 1:** Each active user transmits the first J bits of its message over $n_1 < n$ channel uses. To that end, we use a sensing matrix A drawn from the $\mathbf{LDPC}(\nu, s; M, n_1)$ ensemble, with $M = 2^J$. Each active user chooses one of the $M = 2^J$ columns of A, corresponding to the first J bits in its message, scales it by $\alpha > 0$ and transmits them over the channel. Since there are k active users, the channel output after n_1 uses is $\boldsymbol{y}_1 = \alpha A\boldsymbol{x} + \sigma\boldsymbol{z}$. The vector $\boldsymbol{x}$ consists of entries in $\mathbb{Z}_+$ (all non-negative integers) and satisfies $\|\boldsymbol{x}\|_1 = k$. If all k active users chose messages that begin with a different string of J bits, the vector $\boldsymbol{x}$ will further be in $\{0,1\}^M$. For our choices of J and k described below, typically almost all entries of $\boldsymbol{x}$ will be binary. The basestation (which is now the receiver) applies Algorithm 1 to estimate $\boldsymbol{x}$. In the end, we compute $p_1^{(T+1)}(i)$ for any $i \in [M]$, and output a list consisting of the k coordinates with the highest $p_1^{(T+1)}(i)$.
- **Phase 2:** The basestation applies a set partitioning scheme for collision-free feedback, as described in [71], for broadcasting to the users a list of the k strings of J prefixes it has decoded in phase 1. Naively, this would require broadcasting a message of $k \cdot J$ bits. However, as shown in [71] using a more intelligent scheme, this can information theoretically be done with about $k \cdot \lg_2(e)$ bits, and practical schemes can encode this information using less than $2k$ bits. Each active user decodes the message transmitted by the basestation and finds the location of the J bits prefix of its message within the list of k prefixes that was transmitted.
- **Phase 3:** The remaining $n_2 = n - n_1$ channel uses are split to k slots, each of length $n' = n_2/k$. Each active user transmits the remaining $B - J$ bits of its message during the slot whose index it has decoded in Phase 2. To this end, off-the-shelf point-to-point codes are used. Active users that did not find their J bits prefix in the list of Phase 2, do not transmit a thing in Phase 3.

Note that in the end of this procedure the receiver outputs a list of at most k messages. The message sent by a particular active user enters the list the decoder outputs whenever neither of the following error events occur:

(i) Another active user chose a message with the same J bits prefix, causing a collision in Phase 1 above.
(ii) The J bits prefix of the user's message did not enter the list produced by the basestation in Phase 2.
(iii) The user failed to decode the message sent from the basestation in Phase 2.
(iv) There was a decoding error in the point-to-point transmission of that user in Phase 3.

For the remainder of this discussion, we neglect the cost of Phase 2 in terms of channel resources (energy and bandwidth) and its contribution to the error probability. We do this to avoid the need to model the broadcast channel from the basestation to the active users. In light of Kang and Yu [71], the message sent by the basestation in Phase 2 is significantly shorter than the messages sent by the active users. Adding this to the fact that the basestation is typically less power-constrained than the end-devices in machine-to-machine type communication, it follows that indeed Phase 2 will usually have negligible effect in both aspects (bandwidth and error probability). As mentioned above, our performance figure of merit is the per-user error probability.

We conducted experiments to estimate the expected performance of this end-to-end scheme. In each experiment, each of k users generates a random message of B bits to be transmitted. Let $\boldsymbol{x} \in \{0, 1, \ldots, k\}^{2^J}$ be such that $x_m =$ the number of users who sent message m during Phase 1. The per-user error probability for Phase 1 is

$$\varepsilon_1 = \frac{1}{k} \sum_{i=1}^{k} \Pr\Big(x_{m(i)} > 1 \vee m(i) \notin \mathcal{L} \Big),$$

where $\mathcal{L}$ is the list of k messages returned by the base station, and $m(i)$ is the message transmitted by user i. The error probability ε_1 is estimated via Monte-Carlo simulation. For the error of the second phase, we use the finite block normal approximation of Polyanskiy–Poor–Verdú ([72], Theorem 54):

$$\frac{B-J}{n'} \approx C(P) - \sqrt{\frac{V(P)}{n'}} Q^{-1}(\varepsilon_2), \tag{7}$$

where $n'P$ is the total energy per user, $C(P) = \frac{1}{2} \lg_2(1+P)$ is the AWGN capacity and $V(P) = \frac{P(P+2)}{2(P+1)^2} (\lg_2(e))^2$ is the AWGN channel dispersion. Given a target error probability ε_2, we can solve (7) with an equality to obtain an achievability estimate P^* on the power P necessary to attain user-basestation point-to-point error probability at most ε_2. The total energy per transmitted bit (per user) is just

$$E_b/N_0 = \frac{\frac{1}{2} n' P^* + J \cdot (E_b/N_0)_{phase1}}{B},$$

where, as in the previous section, $(E_b/N_0)_{phase1} = \frac{E_m}{2\sigma^2 \cdot J}$, E_m being the energy of a column of A. For every k, we wanted to find the smallest E_b/N_0 that achieves total per-user error $\varepsilon_1 + \varepsilon_2 = 0.05$. This optimization have has performed numerically.

The performance attained by this end-to-end scheme is plotted in Figure 3. We plot the performance corresponding to Phase 1 implemented by the sensing matrix and recovery algorithm introduced in this paper, as well as an i.i.d. Gaussian sensing matrix and AMP recovery. Both implementations for Phase 1 correspond to similar performance, with slight preference for the latter, and substantially improve the state-of-the-art for unsourced random access with (a small amount of) feedback [12].

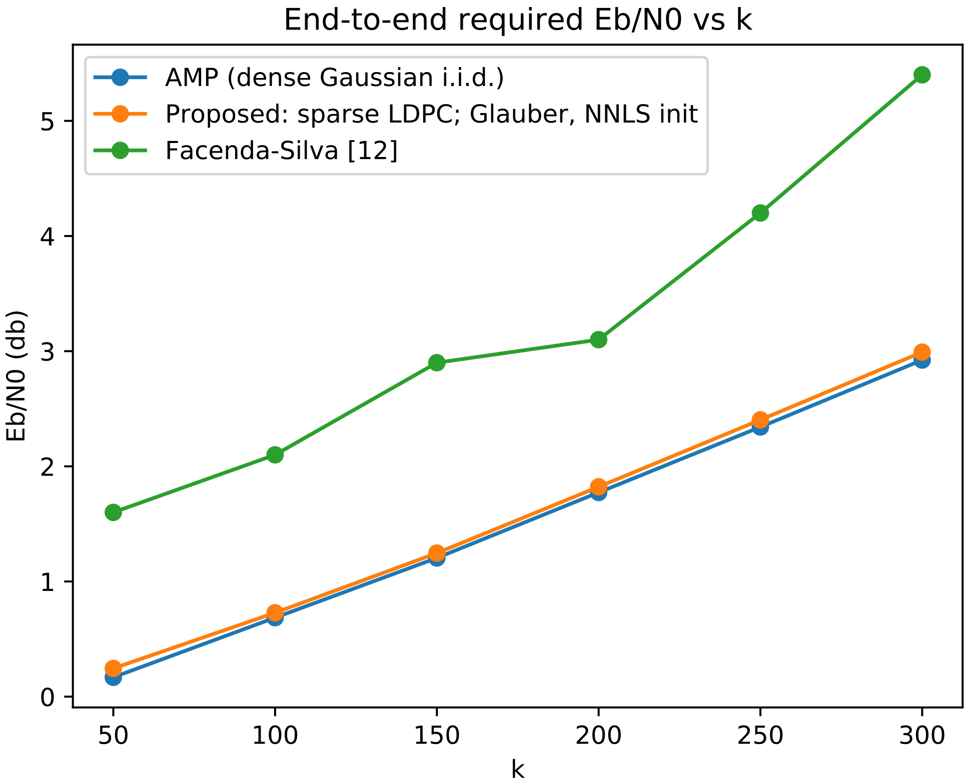

Figure 3. Total E_b/N_0 required to achieve end-to-end PUPE ≤ 0.05. We see that, by using a better compressed sensing algorithm for binary signals, significant gains can be achieved over the current state of the art [12].

4. Conclusions and Additional Discussion

We propose a scheme for compressed sensing of binary signals, consisting of a sparse sensing matrix, based on Gallager's ensemble of LDPC codes, and a decoder based on MCMC. When used as a building block in grant-based random access, the scheme is demonstrated numerically to attain essentially state-of-the-art performance. To conclude, we mention several points that rise up as follow-up questions to our results.

Belief Propagation. One of the most popular algorithms for decoding LDPC codes is Belief Propagation (BP) (see, e.g., [56,63]). We conducted very limited experiments with sum-product and max-product BP (not reported in this paper); our preliminary findings suggest that our MCMC decoder outperforms BP (in terms of its tolerance to noise), at least in the regime considered in Section 3.1. A possible explanation for this could be that the sensing matrix A has many small cycles, which severely violates the tree assumption and is common in BP analysis of LDPC codes. A thorough study of BP for compressed sensing with binary signals is left as an interesting direction for future research.

Grantless unsourced random access. In Section 3.2, we demonstrate that our scheme can attain essentially state-of-the-art performance in grant-based unsourced random access (wherein a compressed sensing problem is solved in the first, scheduling, step). However, most previous works on unsourced random access have considered a different approach, which does not allow for feedback. The idea is to divide transmission into several blocks and perform coding in two steps: (1) an outer code, to allow the decoder to relate ("stitch") messages across different blocks to one another; and (2) an inner code, wherein each user codes its message (payload + parity bits) over an AWGN multiple access channel—in this framework, decoding the inner code boils down to solving a compressed sensing problem with a binary signal. An interesting question is whether our proposed scheme can provide any gains if used to construct an inner code in this framework. In [11], the authors proposed to use a certain tree code (outer code) and an i.i.d. Gaussian sensing matrix for the inner code, together with a certain AMP decoder, which in decoding iteratively passes information between the inner and outer codes. We tried replacing the AMP decoder with our scheme. Specifically, we considered an iterative procedure that alternates between the following steps: (1) run Glauber dynamics on each block, producing a soft decision rule for the (sparse, per-block) activity pattern; and (2) a tree code inference step, that takes the per-block "likelihoods" produced by Glauber dynamics and computes a posterior over the entire activity pattern, by integrating information across all the blocks. The next time we decode the inner code, this posterior is used for the new prior of the signal. Our preliminary experiments indicate that the performance of this combined scheme is rather disappointing and quite far off from state of the art [11].

Generalizing to non-equal channel gains. When discussing random access, we modeled the received signal at the base station as $\boldsymbol{y} = A\boldsymbol{x} + \sigma\boldsymbol{z}$ where $\boldsymbol{x} \in \{0,1\}^M$ is the pattern of active users and $\sigma\boldsymbol{z}$ is Gaussian noise; namely, the channel between the users and the basestation is an AWGN multiple access channel where all gains are equal. This model is based on the assumption of perfect power and phase control, which is not always realistic, and designing communication schemes for the fading model, where channel gains are not assumed equal, is desired. Generalizing our MCMC decoder to incorporate fading looks somewhat challenging. Consider a model $\boldsymbol{y} = AH\boldsymbol{x} + \sigma\boldsymbol{z}$ where $H = \mathrm{diag}(h_1, \ldots, h_m)$ is a diagonal matrix of (random) fading coefficients. We would like to sample from the posterior of $\boldsymbol{x}$ given $\boldsymbol{y}$:

$$\Pr(\boldsymbol{x} = \widehat{\boldsymbol{x}}|\boldsymbol{y}) \propto \mathbb{E}_H \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - AH\widehat{\boldsymbol{x}}\|^2 + \lambda\|\widehat{\boldsymbol{x}}\|_1\right\},$$

where notice that we now need to marginalize over $H = \mathrm{diag}(h_1, \ldots, h_M)$. This marginalization appears to complicate things considerably: in particular, in contrast to the case where H is the identity matrix, in the general case, it is not so straightforward to sample x_i conditioned on all other coordinates. Devising an MCMC decoder that can handle fading is an interesting problem for future research.

Author Contributions: Conceptualization, E.R. and O.O.; methodology, E.R. and O.O.; software, E.R.; validation, E.R.; formal analysis, E.R. and O.O.; investigation, E.R. and O.O.; writing—original draft preparation, E.R. and O.O.; writing—review and editing, E.R. and O.O.; visualization, E.R.; supervision, O.O.; project administration, O.O.; and funding acquisition, O.O. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by ISF under Grant 1791/17 and in part by the GENESIS Consortium via the Israel Ministry of Economy and Industry. The work of Elad Romanov was supported in part by an Einstein-Kaye fellowship from the Hebrew University of Jerusalem.

Data Availability Statement: Python code implementing the proposed algorithm is available in the following url: https://github.com/dalevonamor/BinaryCompressedSensingEntropy.

Acknowledgments: We thank Vamsi K. Amalladinne, Jean-Francois Chamberland and Krishna R. Narayanan for valuable discussions and for kindly sharing with us their code for the scheme described in [11] and Uri Erez for valuable discussions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Omitted Proofs

Appendix A.1. Proof of Lemma 1

Clearly, it suffices to show that $\Pr(\widehat{x}_{\mathrm{SAMP},i} \neq x_i) \leq 2\Pr(\widehat{x}_{\mathrm{BER},i} \neq x_i)$ for all i. In addition, recall that the ith coordinate, $\widehat{x}_{\mathrm{SAMP},i}$, is simply sampled from the posterior $\Pr(x_i|\boldsymbol{y})$.

Note that $\{\widehat{x}_{\mathrm{SAMP},i} = x_i\} \supset \{\widehat{x}_{\mathrm{SAMP},i} = \widehat{x}_{\mathrm{BER},i} \wedge \widehat{x}_{\mathrm{BER},i} = x_i\}$, and therefore

$$\begin{aligned}\Pr(\widehat{x}_{\mathrm{SAMP},i} \neq x_i) &\leq \Pr(\widehat{x}_{\mathrm{SAMP},i} \neq \widehat{x}_{\mathrm{BER},i} \vee \widehat{x}_{\mathrm{BER},i} \neq x_i) \\ &\leq \Pr(\widehat{x}_{\mathrm{SAMP},i} \neq \widehat{x}_{\mathrm{BER},i}) + \Pr(\widehat{x}_{\mathrm{BER},i} \neq x_i)\,.\end{aligned}$$

Thus, we are done once we show that $\Pr(\widehat{x}_{\mathrm{SAMP},i} \neq \widehat{x}_{\mathrm{BER},i}) = \Pr(\widehat{x}_{\mathrm{BER},i} \neq x_i)$. By definition, for any x'_i which is deterministic given $\boldsymbol{y}$,

$$\Pr\big(\widehat{x}_{\mathrm{SAMP},i} \neq x'_i \,|\, \boldsymbol{y}\big) = \Pr\big(x_i \neq x'_i \,|\, \boldsymbol{y}\big)\,,$$

where, on the left, probability is taken only with respect to the sampling procedure. Choosing $x'_i = \widehat{x}_{\mathrm{BER},i}$, and taking the expectation over $\boldsymbol{y}$,

$$\begin{aligned}\Pr(\widehat{x}_{\mathrm{SAMP},i} \neq \widehat{x}_{\mathrm{BER},i}) &= \mathbb{E}\big[\Pr\big(\widehat{x}_{\mathrm{SAMP},i} \neq \widehat{x}_{\mathrm{BER},i} \,|\, \boldsymbol{y}\big)\big] \\ &= \mathbb{E}\big[\Pr\big(x_i \neq \widehat{x}_{\mathrm{BER},i} \,|\, \boldsymbol{y}\big)\big] \\ &= \Pr(x_i \neq \widehat{x}_{\mathrm{BER},i})\,.\end{aligned}$$

Appendix A.2. Proof of Lemma 2

The proof uses the path coupling method, which is a fundamental technique in the theory of Markov chains [61].

Before getting to the proof of Lemma 2, let us start by recalling some useful notions and set some notation.

- **Distance and neighbors on the hypercube**: Denote by $\mathcal{X} = \{0,1\}^M$ the M-dimensional hypercube. $\mathcal{X}$ has a natural graph structure: two vertices $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ are neighbors, denoted $\boldsymbol{x} \sim \boldsymbol{x}'$, iff they differ in exactly one coordinate. Denote by $\mathbf{d_H}(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to [0,\infty)$ the Hamming distance:

$$\mathbf{d_H}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{M} \mathbf{1}_{x_i \neq x'_i}\,.$$

  Of course, Hamming distance coincides with the shortest path distance with respect to the graph structure on $\mathcal{X}$.
- **Coupling:** Let X and X' be two random variables taking values on $\mathcal{X}$. Denote by $\mathbb{P}_X$ and $\mathbb{P}_{X'}$ the laws of X, X', respectively. A coupling between X, X' is a probability distribution $\mathbb{P}_{X,X'}$ on $\mathcal{X} \times \mathcal{X}$, whose X-marginal is $\mathbb{P}_X$ and X'-marginal is $\mathbb{P}_{X'}$. In other words, a coupling is an embedding of two random variables onto a joint probability space, defined by a joint law.

Notation. For $\boldsymbol{x} \in \mathcal{X}$, let $X_{\boldsymbol{x}}$ be the $\mathcal{X}$-valued random variable whose law corresponds to running one step of Glauber dynamics, starting from the initial state $\boldsymbol{x}$ (using the notations of Algorithm 1, one has $\boldsymbol{x}^{(0)} = \boldsymbol{x}$ and so $X_{\boldsymbol{x}} \overset{d}{=} \boldsymbol{x}^{(1)}$, where $\overset{d}{=}$ indicates equality in distribution).

The following result follows from ([61], Corollary 14.7):

Theorem A1. *Suppose that there is $0 \leq \eta < 1$ with the following property: for any two neighbors $\boldsymbol{x} \sim \boldsymbol{x}'$, there exists a contracting coupling of $X_{\boldsymbol{x}}$ and $X_{\boldsymbol{x}'}$ with*

$$\mathbb{E}[\mathbf{d_H}(X_{\boldsymbol{x}}, X_{\boldsymbol{x}'})] \leq \eta\,.$$

Then, for any initial state $\boldsymbol{x}^{(0)} = \boldsymbol{x}$ and $t \geq 1$, one has

$$\mathbf{d_{TV}}\Big(\mathbb{Q}^{(t)}, \mathbb{Q}^{(\infty)}\Big) \leq M \cdot \eta^t\,.$$

Here, $\mathbb{Q}^{(t)}$ is the law of $\boldsymbol{x}^{(t)}$, the state of Glauber dynamics at time t, starting from $\boldsymbol{x}^0 = \boldsymbol{x}$, and $\mathbb{Q}^{(\infty)}$ is the stationary distribution.

The proof of Lemma 2 follows by constructing a contracting coupling between $X_{\boldsymbol{x}}$ and $X_{\boldsymbol{x}'}$ for any $\boldsymbol{x} \sim \boldsymbol{x}'$, and applying Theorem A1. The construction proceeds as follows. Let $\boldsymbol{i} \sim \text{Uniform}([M])$ be a random coordinate to update. For all $\ell \in [M]$, let

$$q_0(\ell) = \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\boldsymbol{x}_{\ell=0}\|^2\right\}, \quad q_1(\ell) = \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - A\boldsymbol{x}_{\ell=1}\|^2 + \lambda\right\},$$

and

$$p_1(\ell) = \frac{q_1(\ell)}{q_0(\ell) + q_1(\ell)} = \varphi\left(\lambda + \frac{1}{2\sigma^2}\left\{\|\boldsymbol{y} - A\boldsymbol{x}_{\ell=0}\|^2 - \|\boldsymbol{y} - A\boldsymbol{x}_{\ell=1}\|^2\right\}\right),$$

where $\varphi(x) = 1/(1 + e^{-x})$ is the logistic function. Let $p_1(\ell)'$ be defined similarly, with $\boldsymbol{x}$ replaced by $\boldsymbol{x}'$. To obtain $X_{\boldsymbol{x}}$ and $X_{\boldsymbol{x}'}$, sample $U \sim \text{Uniform}[0,1]$, independent of $\boldsymbol{i}$. $X_{\boldsymbol{x}}, X_{\boldsymbol{x}'}$ coincide with $\boldsymbol{x}, \boldsymbol{x}'$, respectively, on all coordinates $\ell \neq \boldsymbol{i}$; as for the $\boldsymbol{i}$th coordinate, set $(X_{\boldsymbol{x}})_{\boldsymbol{i}} = 1$ if $U \leq p_1(\boldsymbol{i})$ (and otherwise set to 0), and likewise set $(X_{\boldsymbol{x}'})_{\boldsymbol{i}} = 1$ if $U \leq p_1(\boldsymbol{i})'$. Clearly, the random variables $(X_{\boldsymbol{x}}, X_{\boldsymbol{x}'})$ constructed in this manner have the "correct" marginal distribution; thus, we have defined a legitimate coupling.

It remains to show that this coupling is contracting. Since $\boldsymbol{x} \sim \boldsymbol{x}'$, there is a unique coordinate on which they differ, call it ℓ_0. Observe that, conditioned on $\boldsymbol{i} = \ell_0$, we have $X_{\boldsymbol{x}} = X_{\boldsymbol{x}'}$ exactly. On the other hand, when $\boldsymbol{i} \neq \ell_0$, the Hamming distance either stays the same or increases by 1, depending on U. Indeed, the distance increases if and only if $\min\{p_1(\boldsymbol{i}), p_1(\boldsymbol{i})'\} < U \leq \max\{p_1(\boldsymbol{i}), p_1(\boldsymbol{i})'\}$, and, conditioned on $\boldsymbol{i}$, this happens with probability $|p_1(\boldsymbol{i}) - p_1(\boldsymbol{i})'|$. Thus,

$$\mathbb{E}[\mathbf{d_H}(X_{\boldsymbol{x}}, X_{\boldsymbol{x}'})] = 1 - \frac{1}{M} + \frac{1}{M}\sum_{\ell \in [M],\, \ell \neq \ell_0} |p_1(\ell) - p_1(\ell)'|\,.$$

It remains to bound the expression on the right. For a variable $\ell \in [M]$, let $F(\ell) \subset [n]$ be all the factors to which it is connected in A; similarly, for a factor $f \in [n]$, let $V(f) \subset [M]$ be all the variables to which it is connected. Now,

$$\begin{aligned}
p_1(\ell) &= \varphi\left(\lambda + \frac{1}{2\sigma^2}\left\{\|\boldsymbol{y} - A\boldsymbol{x}_{\ell=0}\|^2 - \|\boldsymbol{y} - A\boldsymbol{x}_{\ell=1}\|^2\right\}\right) \\
&= \varphi\left(\lambda + \frac{1}{2\sigma^2}\sum_{f \in F(\ell)}\left\{\left(y_f - \sum_{k \in V(f)\setminus\{\ell\}} x_k\right)^2 - \left((y_f - 1) - \sum_{k \in V(f)\setminus\{\ell\}} x_k\right)^2\right\}\right) \\
&= \varphi\left(\lambda + \frac{1}{\sigma^2}\sum_{f \in F(\ell)}\left(y_f - \frac{1}{2} - \sum_{k \in V(f)\setminus\{\ell\}} x_k\right)\right),
\end{aligned}$$

and a similar expression holds for $p_1(\ell)'$, with $\boldsymbol{x}$ replaced by $\boldsymbol{x}'$. Since $\varphi(\cdot)$ is $1/4$-Lipschitz,

$$\begin{aligned}
\sum_{\ell \in [M]\setminus\{\ell_0\}} |p_1(\ell) - p_1(\ell)'| &\leq \frac{1}{4\sigma^2}\sum_{\ell \in [M]\setminus\{\ell_0\}}\left|\sum_{f \in F(\ell)}\sum_{k \in V(f)\setminus\{\ell\}}(x_k - x_k')\right| \\
&= \frac{1}{4\sigma^2}\sum_{\ell \in [M]\setminus\{\ell_0\}}\sum_{f \in F(\ell)} \mathbf{1}_{\ell_0 \in V(f)}\,,
\end{aligned}$$

where we used that $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ only on ℓ_0. Observe that the double sum simply counts the number of pairs (ℓ, f) such that ℓ_0 and $\ell \neq \ell_0$ are both connected to the factor f.

Recalling that the degree of all variables is ν and the degree of all factors is s, which is just $\nu(s-1)$. We conclude that

$$\mathbb{E}[\mathbf{d_H}(X_{\boldsymbol{x}}, X_{\boldsymbol{x}'})] \le 1 - \frac{1}{M}\left[1 - \frac{\nu(s-1)}{4\sigma^2}\right]$$

which is < 1 whenever $4\sigma^2 > \nu(s-1)$; this is exactly the condition (6), appearing in the statement of Lemma 2. Applying Theorem A1,

$$\mathbf{d_{TV}}\left(\mathbb{Q}^{(t)}, \mathbb{Q}^{(\infty)}\right) \le M \cdot \left(1 - \frac{1}{M}\left[1 - \frac{\nu(s-1)}{4\sigma^2}\right]\right)^t \le e^{-\frac{t}{4\sigma^2 M}(4\sigma^2 - \nu(s-1)) + \log M}.$$

This bound is $\le\varepsilon$ whenever t is exceeds the quantity in Lemma 2.

Appendix B. Approximate Message Passing (AMP)

In this section, we provide implementation details for the AMP algorithm used in the numeric comparisons of Section 3.

Introduced by Donoho et al. [73], AMP is a state-of-the-art recovery algorithm for solving linear inverse problems $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{z}$. Originally developed for sparse vector recovery (compressed sensing), AMP and its extensions have been shown to yield state-of-the-art algorithms for several other linear inverse problems, with Gaussian or "Gaussian-like" (for example, orthogonally invariant or with some spatially coupled structure) random sensing matrices. For a very partial, selective list of references, see, for example, [73–84]. In unsourced random access, AMP was first used by Fengler et al. [9] and extended by Amalladinne et al. [11].

The AMP algorithm of Donoho et al. [73] uses an iteration of the following form, starting from $\boldsymbol{x}^{(0)} = \mathbf{0}$, $\boldsymbol{r}^{(0)} = \mathbf{0}$:

$$\begin{aligned} \boldsymbol{b}^{(t)} &= A^\top \boldsymbol{r}^{(t)} + \boldsymbol{x}^{(t)} \\ \boldsymbol{x}^{(t+1)} &= f_t\left(\boldsymbol{b}^{(t)}\right) \\ \boldsymbol{r}^{(t+1)} &= \boldsymbol{y} - A\boldsymbol{x}^{(t+1)} + \frac{1}{n}\sum_{i=1}^{n} f_t'\left(\boldsymbol{b}^{(t)}\right). \end{aligned} \tag{A1}$$

Here, $f_t : \mathbb{R} \to \mathbb{R}$ is a sequence of univariate functions (for a vector $\boldsymbol{b}$, $f(\boldsymbol{b})$ stands for applying f separately to each coordinate). The main idea of AMP ("State Evolution") is that in the large-dimensional limit (and under certain technical assumptions), the iterates $\boldsymbol{b}^{(t)}$ behave as $\boldsymbol{b}^{(t)} \approx \boldsymbol{x} + \sigma_t \mathcal{N}(0, I)$; that is, additively corrupted measurements of the true signal $\boldsymbol{x}$. The variance σ_t^2 can be estimated from $\boldsymbol{r}^{(t)}$, which approximately has the law $\mathcal{N}(0, \sigma_t^2 I)$; conversely, it can be tracked via an explicit recursive formula. For our experiments, we use the proposal of Donoho et al. [74], and use the robust estimator $\hat{\sigma}_t = \mathrm{median}(|\boldsymbol{r}^{(t)}|)/\Phi^{-1}(0.75)$, where $\Phi^{-1}(\cdot)$ is the inverse Gaussian CDF.

The details of the AMP algorithm (A1) rely on the specific choice of functions f_t. In compressed sensing of binary signals (1), a natural choice is the MSE-optimal estimator for $\boldsymbol{x}$ from $\boldsymbol{b} = \boldsymbol{x} + \sigma_t \mathcal{N}(0, I)$, namely,

$$f_t(\boldsymbol{b})_i = \mathbb{E}[x_i \mid b_i] = \frac{\frac{k}{M} \cdot e^{-\frac{1}{2\sigma_t^2}(b_i - 1)^2}}{\left(1 - \frac{k}{M}\cdot\right) e^{-\frac{1}{2\sigma_t^2} b_i^2} + \frac{k}{M} \cdot e^{-\frac{1}{2\sigma_t^2}(b_i - 1)^2}}.$$

This is what we used for the experiments presented in Section 3.

References

1. Polyanskiy, Y. A perspective on massive random-access. In Proceedings of the International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 2523–2527.
2. Vem, A.; Narayanan, K.R.; Chamberland, J.; Cheng, J. A User-Independent Successive Interference Cancellation Based Coding Scheme for the Unsourced Random Access Gaussian Channel. *IEEE Trans. Commun.* **2019**, *67*, 8258–8272. [CrossRef]
3. Ordentlich, O.; Polyanskiy, Y. Low complexity schemes for the random access Gaussian channel. In Proceedings of the International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 2528–2532.
4. Marshakov, E.; Balitskiy, G.; Andreev, K.; Frolov, A. A polar code based unsourced random access for the Gaussian MAC. In Proceedings of the Vehicular Technology Conference, Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
5. Calderbank, R.; Thompson, A. CHIRRUP: A practical algorithm for unsourced multiple access. *Inf. Inference J. IMA* **2020**, *9*, 875–897. [CrossRef]
6. Chen, Z.; Sohrabi, F.; Liu, Y.F.; Yu, W. Covariance Based Joint Activity and Data Detection for Massive Random Access with Massive MIMO. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Tokyo, Japan, 15–20 July 2019; pp. 1–6. [CrossRef]
7. Amalladinne, V.K.; Chamberland, J.F.; Narayanan, K.R. A Coded Compressed Sensing Scheme for Unsourced Multiple Access. *IEEE Trans. Inf. Theory* **2020**, *66*, 6509–6533. [CrossRef]
8. Glebov, A.; Matveev, N.; Andreev, K.; Frolov, A.; Turlikov, A. Achievability Bounds for T-Fold Irregular Repetition Slotted ALOHA Scheme in the Gaussian MAC. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakech, Morocco, 15–18 April 2019; pp. 1–6.
9. Fengler, A.; Jung, P.; Caire, G. SPARCs and AMP for unsourced random access. In Proceedings of the International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 2843–2847.
10. Kowshik, S.S.; Andreev, K.; Frolov, A.; Polyanskiy, Y. Energy efficient random access for the quasi-static fading MAC. In Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 2768–2772.
11. Amalladinne, V.K.; Pradhan, A.K.; Rush, C.; Chamberland, J.F.; Narayanan, K.R. On Approximate Message Passing for Unsourced Access with Coded Compressed Sensing. In Proceedings of the International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020; pp. 2995–3000.
12. Facenda, G.K.; Silva, D. Efficient scheduling for the massive random access Gaussian channel. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7598–7609. [CrossRef]
13. Decurninge, A.; Land, I.; Guillaud, M. Tensor-Based Modulation for Unsourced Massive Random Access. *arXiv* **2020**, arXiv:2006.06797.
14. Shyianov, V.; Bellili, F.; Mezghani, A.; Hossain, E. Massive Unsourced Random Access Based on Uncoupled Compressive Sensing: Another Blessing of Massive MIMO. *arXiv* **2020**, arXiv:2002.03044.
15. Wu, Y.; Gao, X.; Zhou, S.; Yang, W.; Polyanskiy, Y.; Caire, G. Massive Access for Future Wireless Communication Systems. *IEEE Wirel. Commun.* **2020**, *27*, 148–156. [CrossRef]
16. Cormode, G.; Muthukrishnan, S. Combinatorial algorithms for compressed sensing. In Proceedings of the International Colloquium on Structural Information and Communication Complexity, Chester, UK, 2–5 July 2006; pp. 280–294.
17. Gilbert, A.C.; Strauss, M.J.; Tropp, J.A.; Vershynin, R. One sketch for all: Fast algorithms for compressed sensing. In Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 11–13 June 2007; pp. 237–246.
18. Ngo, H.Q.; Porat, E.; Rudra, A. Efficiently decodable compressed sensing by list-recoverable codes and recursion. In Proceedings of the 29th Symposium on Theoretical Aspects of Computer Science (STACS'12), Paris, France, 29 February–3 March 2012; Volume 14, pp. 230–241.
19. Amalladinne, V.K.; Vem, A.; Soma, D.K.; Narayanan, K.R.; Chamberland, J. A Coupled Compressive Sensing Scheme for Unsourced Multiple Access. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 6628–6632. [CrossRef]
20. Amalladinne, V.K.; Pradhan, A.K.; Rush, C.; Chamberland, J.F.; Narayanan, K.R. Unsourced random access with coded compressed sensing: Integrating AMP and belief propagation. *arXiv* **2020**, arXiv:2010.04364.
21. Schouhamer Immink, K.; Siegel, P.; Wolf, J. Codes for digital recorders. *IEEE Trans. Inf. Theory* **1998**, *44*, 2260–2299. [CrossRef]
22. Cai, Y.; Haratsch, E.F.; Mutlu, O.; Mai, K. Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling. In Proceedings of the 2013 Design, Automation Test in Europe Conference Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 1285–1290.
23. Mazumdar, A.; Chandar, V.; Wornell, G.W. Update-Efficiency and Local Repairability Limits for Capacity Approaching Codes. *IEEE J. Sel. Areas Commun.* **2014**, *32*, 976–988. [CrossRef]
24. Aldridge, M.; Johnson, O.; Scarlett, J. Group Testing: An Information Theory Perspective. *Found. Trends Commun. Inf. Theory* **2019**, *15*, 196–392. [CrossRef]
25. Foucart, S.; Rauhut, H. A Mathematical Introduction to Compressive Sensing. In *Applied and Numerical Harmonic Analysis*; Springer: New York, NY, USA, 2013.
26. Donoho, D.L.; Tanner, J. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 9446–9451. [CrossRef] [PubMed]
27. Candes, E.J.; Tao, T. Decoding by linear programming. *IEEE Trans. Inf. Theory* **2005**, *51*, 4203–4215. [CrossRef]

28. Candes, E.J.; Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory* **2006**, *52*, 5406–5425. [CrossRef]
29. Candès, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [CrossRef]
30. Baraniuk, R.G. Compressive sensing [lecture notes]. *IEEE Signal Process. Mag.* **2007**, *24*, 118–121. [CrossRef]
31. Duarte, M.F.; Eldar, Y.C. Structured compressed sensing: From theory to applications. *IEEE Trans. Signal Process.* **2011**, *59*, 4053–4085. [CrossRef]
32. Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
33. Eldar, Y.C.; Kutyniok, G. *Compressed Sensing: Theory and Applications*; Cambridge University Press: Cambridge, UK, 2012.
34. Marques, E.C.; Maciel, N.; Naviner, L.; Cai, H.; Yang, J. A review of sparse recovery algorithms. *IEEE Access* **2018**, *7*, 1300–1322. [CrossRef]
35. Brunel, L.; Boutros, J. Euclidean space lattice decoding for joint detection in CDMA systems. In Proceedings of the ITW'99, Kruger National Park, South Africa, 25 June 1999; p. 129.
36. Thrampoulidis, C.; Zadik, I.; Polyanskiy, Y. A simple bound on the BER of the MAP decoder for massive MIMO systems. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 4544–4548.
37. Reeves, G.; Xu, J.; Zadik, I. The all-or-nothing phenomenon in sparse linear regression. In Proceedings of the Conference on Learning Theory, Phoenix, AZ, USA, 25–28 June 2019; pp. 2652–2663.
38. Jin, Y.; Kim, Y.; Rao, B.D. Limits on Support Recovery of Sparse Signals via Multiple-Access Communication Techniques. *IEEE Trans. Inf. Theory* **2011**, *57*, 7877–7892. [CrossRef]
39. Hastie, T.; Tibshirani, R.; Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations*; CRC Press: Boca Raton, FL, USA, 2015.
40. Wainwright, M.J. Sharp thresholds for High-Dimensional and noisy sparsity recovery using ℓ_1-Constrained Quadratic Programming (Lasso). *IEEE Trans. Inf. Theory* **2009**, *55*, 2183–2202. [CrossRef]
41. Gamarnik, D.; Zadik, I. Sparse high-dimensional linear regression. algorithmic barriers and a local search algorithm. *arXiv* **2017**, arXiv:1711.04952.
42. Bruckstein, A.M.; Elad, M.; Zibulevsky, M. On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Trans. Inf. Theory* **2008**, *54*, 4813–4820. [CrossRef]
43. Slawski, M.; Hein, M. Sparse recovery by thresholded non-negative least squares. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 1926–1934.
44. Meinshausen, N. Sign-constrained least squares estimation for high-dimensional regression. *Electron. J. Stat.* **2013**, *7*, 1607–1631. [CrossRef]
45. Foucart, S.; Koslicki, D. Sparse recovery by means of nonnegative least squares. *IEEE Signal Process. Lett.* **2014**, *21*, 498–502. [CrossRef]
46. Kueng, R.; Jung, P. Robust nonnegative sparse recovery and the nullspace property of 0/1 measurements. *IEEE Trans. Inf. Theory* **2017**, *64*, 689–703. [CrossRef]
47. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
48. Indyk, P.; Ruzic, M. Near-optimal sparse recovery in the l1 norm. In Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science, Philadelphia, PA, USA, 26–28 October 2008; pp. 199–207.
49. Berinde, R.; Gilbert, A.C.; Indyk, P.; Karloff, H.; Strauss, M.J. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In Proceedings of the 2008 46th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 23–26 September 2008; pp. 798–805.
50. Gilbert, A.; Indyk, P. Sparse recovery using sparse matrices. *Proc. IEEE* **2010**, *98*, 937–947. [CrossRef]
51. Jafarpour, S.; Xu, W.; Hassibi, B.; Calderbank, R. Efficient and robust compressed sensing using optimized expander graphs. *IEEE Trans. Inf. Theory* **2009**, *55*, 4299–4308. [CrossRef]
52. Arora, S.; Daskalakis, C.; Steurer, D. Message-passing algorithms and improved LP decoding. *IEEE Trans. Inf. Theory* **2012**, *58*, 7260–7271. [CrossRef]
53. Dimakis, A.G.; Smarandache, R.; Vontobel, P.O. LDPC codes for compressed sensing. *IEEE Trans. Inf. Theory* **2012**, *58*, 3093–3114. [CrossRef]
54. Zhang, F.; Pfister, H.D. Verification decoding of high-rate LDPC codes with applications in compressed sensing. *IEEE Trans. Inf. Theory* **2012**, *58*, 5042–5058. [CrossRef]
55. Khajehnejad, M.A.; Dimakis, A.G.; Xu, W.; Hassibi, B. Sparse recovery of nonnegative signals with minimal expansion. *IEEE Trans. Signal Process.* **2010**, *59*, 196–208. [CrossRef]
56. Richardson, T.; Urbanke, R. *Modern Coding Theory*; Cambridge University Press: Cambridge, UK, 2008.
57. Burshtein, D.; Miller, G. Expander graph arguments for message-passing algorithms. *IEEE Trans. Inf. Theory* **2001**, *47*, 782–790. [CrossRef]
58. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]

59. Kudekar, S.; Kumar, S.; Mondelli, M.; Pfister, H.D.; Urbanke, R. Comparing the bit-MAP and block-MAP decoding thresholds of Reed-Muller codes on BMS channels. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 10–15 July 2016; pp. 1755–1759.
60. Liu, J.; Cuff, P.; Verdú, S. On α-decodability and α-likelihood decoder. In Proceedings of the 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 3–6 October 2017; pp. 118–124.
61. Levin, D.A.; Peres, Y. *Markov Chains and Mixing Times*; American Mathematical Soc.: Providence, RI, USA, 2017; Volume 107.
62. Neal, R.M. *Monte Carlo Decoding of LDPC Codes*; Technical Report; 2001. Available online: https://webcache.googleusercontent.com/search?q=cache:ujdA8n8zmD8J:https://www.cs.toronto.edu/~radford/ftp/mcdecode-talk.ps+&cd=3&hl=en&ct=clnk&gl=hk (accessed on 14 May 2021).
63. Mezard, M.; Montanari, A. *Information, Physics, and Computation*; Oxford University Press: Oxford, UK, 2009.
64. Hansen, M.; Hassibi, B.; Dimakis, A.G.; Xu, W. Near-optimal detection in MIMO systems using Gibbs sampling. In Proceedings of the GLOBECOM 2009—2009 IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
65. Hassibi, B.; Dimakis, A.G.; Papailiopoulos, D. MCMC methods for integer least-squares problems. In Proceedings of the 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 29 September–1 October 2010; pp. 495–501.
66. Hassibi, B.; Hansen, M.; Dimakis, A.G.; Alshamary, H.A.J.; Xu, W. Optimized Markov chain Monte Carlo for signal detection in MIMO systems: An analysis of the stationary distribution and mixing time. *IEEE Trans. Signal Process.* **2014**, *62*, 4436–4450. [CrossRef]
67. Bhatt, A.; Huang, J.; Kim, Y.; Ryu, J.J.; Sen, P. Monte Carlo Methods for Randomized Likelihood Decoding. In Proceedings of the 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2–5 October 2018; pp. 204–211. [CrossRef]
68. Doucet, A.; Wang, X. Monte Carlo methods for signal processing: A review in the statistical signal processing context. *IEEE Signal Process. Mag.* **2005**, *22*, 152–170. [CrossRef]
69. Lucka, F. Fast Markov chain Monte Carlo sampling for sparse Bayesian inference in high-dimensional inverse problems using L1-type priors. *Inverse Probl.* **2012**, *28*, 125012. [CrossRef]
70. Fengler, A.; Jung, P.; Caire, G. SPARCs for unsourced random access. *arXiv* **2019**, arXiv:1901.06234.
71. Kang, J.; Yu, W. Minimum Feedback for Collision-Free Scheduling in Massive Random Access. In Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020; pp. 2989–2994.
72. Polyanskiy, Y.; Poor, H.V.; Verdú, S. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory* **2010**, *56*, 2307–2359. [CrossRef]
73. Donoho, D.L.; Maleki, A.; Montanari, A. Message-passing algorithms for compressed sensing. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 18914–18919. [CrossRef] [PubMed]
74. Donoho, D.L.; Johnstone, I.; Montanari, A. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising. *IEEE Trans. Inf. Theory* **2013**, *59*, 3396–3433. [CrossRef]
75. Bayati, M.; Montanari, A. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Trans. Inf. Theory* **2011**, *57*, 764–785. [CrossRef]
76. Krzakala, F.; Mézard, M.; Sausset, F.; Sun, Y.; Zdeborová, L. Statistical-physics-based reconstruction in compressed sensing. *Phys. Rev. X* **2012**, *2*, 021005. [CrossRef]
77. Rangan, S. Generalized approximate message passing for estimation with random linear mixing. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 31 July–5 August 2011; pp. 2168–2172.
78. Donoho, D.L.; Javanmard, A.; Montanari, A. Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing. *IEEE Trans. Inf. Theory* **2013**, *59*, 7434–7464. [CrossRef]
79. Metzler, C.A.; Maleki, A.; Baraniuk, R.G. From denoising to compressed sensing. *IEEE Trans. Inf. Theory* **2016**, *62*, 5117–5144. [CrossRef]
80. Romanov, E.; Gavish, M. Near-optimal matrix recovery from random linear measurements. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 7200–7205. [CrossRef]
81. Schniter, P.; Rangan, S. Compressive phase retrieval via generalized approximate message passing. *IEEE Trans. Signal Process.* **2014**, *63*, 1043–1055. [CrossRef]
82. Parker, J.T.; Schniter, P.; Cevher, V. Bilinear generalized approximate message passing—Part I: Derivation. *IEEE Trans. Signal Process.* **2014**, *62*, 5839–5853. [CrossRef]
83. Rush, C.; Greig, A.; Venkataramanan, R. Capacity-achieving sparse superposition codes via approximate message passing decoding. *IEEE Trans. Inf. Theory* **2017**, *63*, 1476–1500. [CrossRef]
84. Berthier, R.; Montanari, A.; Nguyen, P.M. State evolution for approximate message passing with non-separable functions. *Inf. Inference J. IMA* **2020**, *9*, 33–79. [CrossRef]