# Complexity-Matching Universal Signal Estimation in Compressed Sensing

Junan Zhu, *Student Member, IEEE,* Dror Baron, *Senior Member, IEEE,* and

Marco F. Duarte, *Member, IEEE*

**Abstract**

We study the compressed sensing (CS) signal estimation problem where an input signal is measured via a linear matrix multiplication under additive noise. While this setup usually assumes sparsity or compressibility in the input signal during recovery, the signal structure that can be leveraged is often not known *a priori*. In this paper, we consider *universal* CS recovery, where the statistics of a stationary ergodic signal source are estimated simultaneously with the signal itself. Inspired by Kolmogorov complexity and minimum description length, we focus on a maximum *a posteriori* (MAP) estimation framework that leverages universal priors to match the complexity of the source. Our framework can also be applied to general linear inverse problems where more measurements than in CS might be needed. We provide theoretical results that support the algorithmic feasibility of universal MAP estimation using a Markov chain Monte Carlo implementation, which is computationally challenging. We incorporate some techniques to accelerate the algorithm while providing comparable and in many cases better reconstruction quality than existing algorithms. Experimental results show the promise of universality in CS, particularly for low-complexity sources that do not exhibit standard sparsity or compressibility.

**Index Terms**

Compressed sensing, MAP estimation, Markov chain Monte Carlo, universal algorithms.

## I. INTRODUCTION

Since many systems in science and engineering are approximately linear, linear inverse problems have attracted great attention in the signal processing community. An input signal $x \in \mathbb{R}^N$ is recorded via a linear operator under

additive noise:

$$y = \Phi x + z, \tag{1}$$

where $\Phi$ is an $M \times N$ matrix and $z \in \mathbb{R}^M$ denotes the noise. The goal is to estimate $x$ from the measurements $y$ given knowledge of $\Phi$ and a model for the noise $z$. When $M \ll N$, the setup is known as compressed sensing (CS) and the estimation problem is commonly referred to as recovery or reconstruction; by posing a sparsity or compressibility requirement on the signal and using this requirement as a prior during recovery, it is indeed possible to accurately estimate $x$ from $y$ [4, 5]. On the other hand, we might need more measurements than the signal length when the signal is dense or the noise is substantial.

Wu and Verdú [6] have shown that independent and identically distributed (i.i.d.) Gaussian sensing matrices achieve the same phase-transition threshold as optimal nonlinear encoding, for any discrete continuous mixture. Hence, in CS the acquisition can be designed independently of the particular signal prior through the use of randomized Gaussian matrices $\Phi$. Nevertheless, the majority of (if not all) existing recovery algorithms require knowledge of the sparsity structure of $x$, i.e., the choice of a *sparsifying transform* $W$ that renders a sparse coefficient vector $\theta = W^{-1}x$ for the signal.

The large majority of recovery algorithms pose a sparsity prior on the signal $x$, e.g., [4, 5, 7]. A second, separate class of Bayesian CS recovery algorithms poses a probabilistic prior for the coefficients of $x$ in a known transform domain [8–12]. Given a probabilistic model, some related message passing approaches learn the parameters of the signal model and achieve the minimum mean squared error (MMSE) in some settings; examples include EM-GM-AMP-MOS [13], turboGAMP [14], and AMP-MixD [15]. As a third alternative, complexity-penalized least square methods [16–20] can use arbitrary prior information on the signal model and provide analytical guarantees, but are only computationally efficient for specific signal models, such as the independent-entry Laplacian model [18]. For example, Donoho et al. [17] relies on Kolmogorov complexity, which cannot be computed [21, 22]. As a fourth alternative, there exist algorithms that can formulate dictionaries that yield sparse representations for the signals of interest when a large amount of training data is available [23–25]. When the signal is non-i.i.d., existing algorithms require either prior knowledge of the probabilistic model [14] or the use of training data [26].

In certain cases, one might not be certain about the structure or statistics of the source prior to recovery. Uncertainty about such structure may result in a sub-optimal choice of the sparsifying transform $W$, yielding a coefficient vector $\theta$ that requires more measurements to achieve reasonable estimation quality; uncertainty about the statistics of the source will make it difficult to select a prior or model for Bayesian algorithms. Thus, it would be desirable to formulate algorithms to estimate $x$ that are agnostic to the particular statistics of the signal. Therefore, we shift our focus from the standard sparsity or compressibility priors to *universal* priors [27, 28]. Such concepts have been previously leveraged in the Kolmogorov sampler universal denoising algorithm [29], which minimizes Kolmogorov complexity [2, 3, 22, 30–34]. Related approaches based on minimum description length (MDL) [35–37] minimize the complexity of the estimated signal with respect to some class of parametric sources.

Unfortunately, MDL can provide a suitable algorithmic recovery framework primarily for parametric sources [38]. Alternative approaches for non-parametric sources based on Kolmogorov complexity are not computable in prac-

tice [21, 22]. To address this computational problem, we confine our attention to stationary ergodic sources and develop an algorithmic framework for universal signal estimation in CS systems, which can be applied to general linear inverse problems where more measurements might be needed. Our framework leverages the fact that for stationary ergodic sources, both the per-symbol empirical entropy and Kolmogorov complexity converge asymptotically almost surely to the entropy rate of the source [21]. We aim to minimize the empirical entropy; our minimization is regularized by introducing a log likelihood for the noise model, which is equivalent to the standard least squares under additive white Gaussian noise. Other noise distributions are readily supported.

We make several contributions toward our universal CS framework. First, we apply a specific quantization grid to a maximum *a posteriori* (MAP) estimator driven by a universal prior, providing a finite-computation universal estimation scheme; our scheme can also be applied to general linear inverse problems where more measurements might be needed. Second, we propose a recovery algorithm based on Markov chain Monte Carlo (MCMC) [39] to approximate this estimation procedure. Third, we prove that for a sufficiently large number of iterations the output of our MCMC recovery algorithm converges to the correct MAP estimate. Fourth, we identify computational bottlenecks in the implementation of our MCMC estimator and show approaches to reduce their complexity. Fifth, we develop an adaptive quantization scheme that tailors a set of reproduction levels to minimize the quantization error within the MCMC iterations and that provides an accelerated implementation. Sixth, we propose a framework that adaptively adjusts the cardinality (or size for short) of the adaptive quantizer to match the complexity of the input signal, in order to further reduce the quantization error and computation. We note in passing that averaging over the outputs of different runs of the same signal with the same measurements will yield lower mean squared error (MSE) for our proposed algorithm.

To showcase the potential of our universal estimation approach, Fig. 1 illustrates recovery results from Gaussian measurement matrices for a four-state Markov source of length $N = 10000$ that generates the pattern $+1, +1, -1, -1, +1, +1, -1, -1 \ldots$ with 3% errors in state transitions, resulting in the signal switching from $-1$ to $+1$ or vice versa either too early or too late (see Section VI for details on the simulation setup). Note that the reconstruction algorithm does not know that this source is a binary source. While it is well known that sparsity-promoting recovery algorithms [7, 14, 40] can recover sparse sources from linear measurements, the aforementioned switching source is not sparse in conventional sparsifying bases (e.g., Fourier, wavelet, and discrete cosine transforms), rendering such sparsifying transforms not applicable. Signals generated by this Markov source can be sparsified using an averaging analysis matrix [41] whose diagonal and first three lower sub-diagonals are filled with $+1$, and all other entries are 0; this transform yields 6% non-zeros in the sparse coefficient vector. However, even if this matrix is known *a priori*, existing algorithms based on analysis sparsity [41] do not perform satisfactorily, yielding mean signal-to-distortion ratios below 5 dB (cf. Section VI). In contrast, our size- and level-adaptive MCMC (SLA-MCMC, cf. Section V) algorithm estimates this source with high fidelity when a moderate number of measurements $M$ are available. We provide more experimental results in Section VI to show that the performance of MCMC is comparable to and in many cases better than existing algorithms.

This paper is organized as follows. Section II provides background content. Section III overviews MAP estimation, quantization, and introduces universal MAP estimation. Section IV formulates an initial MCMC algorithm for
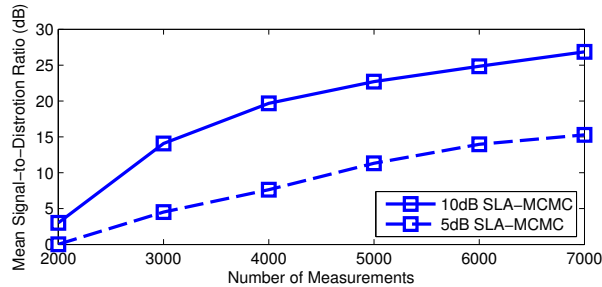
Fig. 1. *SLA-MCMC estimation results for a four-state Markov switching source as a function of the number of Gaussian random measurements $M$ for different signal-to-noise ratio (SNR) values ($N = 10000$). Existing CS algorithms fail at reconstructing this signal, because it is not sparse.*

universal MAP estimation, Section V describes several improvements to this initial algorithm, and Section VI presents experimental results. We conclude in Section VII. The proof of our main theoretical result appears in the appendix.

## II. BACKGROUND AND RELATED WORK

### A. Compressed sensing

Consider the noisy measurement setup via a linear operator (1). The input signal $x \in \mathbb{R}^N$ is generated by a stationary ergodic source $X$, and must be estimated from $y$ and $\Phi$. *The distribution $f_X$ that generates $x$ is unknown.* The matrix $\Phi \in \mathbb{R}^{M \times N}$ has i.i.d. Gaussian entries, $\Phi(m, n) \sim \mathcal{N}(0, \frac{1}{M})$.[1] These moments ensure that the columns of the matrix have unit norm on average. For concrete analysis, we assume that the noise $z \in \mathbb{R}^M$ is i.i.d. Gaussian, with mean zero and known[2] variance $\sigma_Z^2$ for simplicity.

We focus on the setting where $M, N \to \infty$ and the aspect ratio is positive:

$$R \triangleq \lim_{N \to \infty} \frac{M}{N} > 0. \tag{2}$$

Similar settings have been discussed in the literature [42, 43]. When $M \ll N$, this setup is known as CS; otherwise, it is a general linear inverse problem setting. Since $x$ is generated by an unknown source, we must search for an estimation mechanism that is agnostic to the specific distribution $f_X$.

### B. Related work

For a scalar channel with a discrete-valued signal $x$, e.g., $\Phi$ is an identity matrix and $y = x + z$, Donoho proposed the Kolmogorov sampler (KS) for denoising [29],

$$x_{KS} \triangleq \arg\min_w K(w) \text{ s.t. } \|w - y\|^2 < \tau, \tag{3}$$

---

[1]In contrast to our analytical and numerical results, the algorithm presented in Section IV is not dependent on a particular choice for the matrix $\Phi$.

[2]We assume that the noise variance is known or can be estimated [8, 15].

where $K(x)$ denotes the Kolmogorov complexity of $x$, defined as the length of the shortest input to a Turing machine [44] that generates the output $x$ and then halts,[3] and $\tau = N\sigma_Z^2$ controls for the presence of noise. It can be shown that $K(x)$ asymptotically captures the statistics of the stationary ergodic source $X$, and the per-symbol complexity achieves the entropy rate $H \triangleq H(X)$, i.e., $\lim_{N \to \infty} \frac{1}{N} K(x) = H$ almost surely [21, p. 154, Theorem 7.3.1]. Noting that universal lossless compression algorithms [27, 28] achieve the entropy rate for any discrete-valued finite state machine source $X$, we see that these algorithms achieve the per-symbol Kolmogorov complexity almost surely.

Donoho et al. expanded KS to the linear CS measurement setting $y = \Phi x$ but did not consider measurement noise [17]. Recent papers by Jalali and coauthors [33, 34], which appeared simultaneously with our work [2, 3], provide an analysis of a modified KS suitable for measurements corrupted by noise of bounded magnitude. Inspired by Donoho et al. [17], we estimate $x$ from noisy measurements $y$ using the empirical entropy as a proxy for the Kolmogorov complexity (cf. Section IV-A).

Separate notions of complexity-penalized least squares have also been shown to be well suited for denoising and CS recovery [16–20, 35–37]. For example, minimum description length (MDL) [20, 35–37] provides a framework composed of classes of models for which the signal complexity can be defined sharply. In general, complexity-penalized least square approaches can yield MDL-flavored CS recovery algorithms that are adaptive to parametric classes of sources [16–19]. An alternative universal denoising approach computes the universal conditional expectation of the signal [3, 15].

## III. Universal MAP estimation and discretization

This section briefly reviews MAP estimation and then applies it over a quantization grid, where a universal prior is used for the signal. Additionally, we provide a conjecture for the MSE achieved by our universal MAP scheme.

### A. Discrete MAP estimation

In this subsection, we assume for exposition purposes that we know the signal statistics $f_X$. Given the measurements $y$, the MAP estimator for $x$ has the form

$$x_{MAP} \triangleq \arg \max_w f_X(w) f_{Y|X}(y|w). \tag{4}$$

Because $z$ is i.i.d. Gaussian with mean zero and known variance $\sigma_Z^2$, $f_{Y|X}(y|w) = c_1 e^{-c_2 \|y - \Phi w\|^2}$, where $c_1 = (2\pi\sigma_Z^2)^{-M/2}$ and $c_2 = \frac{1}{2\sigma_Z^2}$ are constants, and $\|\cdot\|$ denotes the Euclidean norm.[4] Plugging into (4) and taking log likelihoods, we obtain $x_{MAP} = \arg \min_w \Psi^X(w)$, where $\Psi^X(\cdot)$ denotes the objective function (risk) $\Psi^X(w) \triangleq -\ln(f_X(w)) + c_2 \|y - \Phi w\|^2$; our ideal risk would be $\Psi^X(x_{MAP})$.

---

[3]For real-valued $x$, Kolmogorov complexity (KC) can be approximated using a fine quantizer. Note that the algorithm developed in this paper uses a coarse quantizer and does not rely on KC due to the absence of a feasible method for its computation [21, 22] (cf. Section V).

[4]Other noise distributions are readily supported, e.g., for i.i.d. Laplacian noise, we need to change the $\ell_2$ norm to an $\ell_1$ norm and adjust $c_1$ and $c_2$ accordingly.

Instead of performing continuous-valued MAP estimation, we optimize for the MAP in the discretized domain $\mathcal{R}^N$, with $\mathcal{R}$ being defined as follows. Adapting the approach of Baron and Weissman [45], we define the set of data-independent reproduction levels for quantizing $x$ as

$$\mathcal{R} \triangleq \left\{ \ldots, -\frac{1}{\gamma}, 0, \frac{1}{\gamma}, \ldots \right\}, \tag{5}$$

where $\gamma = \lceil \ln(N) \rceil$. As $N$ increases, $\mathcal{R}$ will quantize $x$ to a greater resolution. These reproduction levels simplify the estimation problem from continuous to discrete.

Having discussed our reproduction levels in the set $\mathcal{R}$, we provide a technical condition on boundedness of the signal.

*Condition 1:* We require that the probability density $f_X$ has bounded support, i.e., there exists $\Lambda = [x_{\min}, x_{\max}]$ such that $f_X(x) = 0$ for $x \notin \Lambda^N$.

A limitation of the data-independent reproduction level set (5) is that $\mathcal{R}$ has infinite cardinality (or size for short). Thanks to Condition 1, for each value of $\gamma$ there exists a constant $c_3 > 0$ such that a finite set of reproduction levels

$$\mathcal{R}_F \triangleq \left\{ -\frac{c_3\gamma^2}{\gamma}, -\frac{c_3\gamma^2 - 1}{\gamma}, \ldots, \frac{c_3\gamma^2}{\gamma} \right\} \tag{6}$$

will quantize the range of values $\Lambda$ to the same accuracy as that of (5). We call $\mathcal{R}_F$ the *reproduction alphabet*, and each element in it a (*reproduction*) *level*. This finite quantizer reduces the complexity of the estimation problem from infinite to combinatorial. In fact, $x_i \in [x_{\min}, x_{\max}]$ under Condition 1. Therefore, for all $c_3 > 0$ and sufficiently large $N$, this set of levels will cover the range $[x_{\min}, x_{\max}]$. The resulting reduction in complexity is due to the structure in $\mathcal{R}_F$ and independent of the particular statistics of the source $X$.

Now that we have set up a quantization grid $(\mathcal{R}_F)^N$ for $x$, we convert the distribution $f_X$ to a probability mass function (PMF) $\mathbb{P}_X$ over $(\mathcal{R}_F)^N$. Let $f_{\mathcal{R}_F} \triangleq \sum_{w \in (\mathcal{R}_F)^N} f_X(w)$, and define the PMF $\mathbb{P}_X(\cdot)$ as $\mathbb{P}_X(w) \triangleq \frac{f_X(w)}{f_{\mathcal{R}_F}}$. Then $x_{MAP}(\mathcal{R}_F) \triangleq \arg\min_{w \in (\mathcal{R}_F)^N} \left( -\ln(\mathbb{P}_X(w)) + c_2 \|y - \Phi w\|^2 \right)$ gives the MAP estimate of $x$ over $(\mathcal{R}_F)^N$. As $N$ increases, $\mathbb{P}_X$ will approximate $f_X$ more closely under (6).

### B. Universal MAP estimation

We now describe a universal estimator for CS over a quantized grid. Consider a prior $\mathbb{P}_U$ that might involve Kolmogorov complexity [30–32], e.g., $\mathbb{P}_U(w) = 2^{-K(w)}$, or MDL complexity with respect to some class of parametric sources [35–37]. We call $\mathbb{P}_U$ a *universal prior* if it has the fortuitous property that for every stationary ergodic source $X$ and fixed $\epsilon > 0$, there exists some minimum $N_0(X, \epsilon)$ such that $-\ln(\mathbb{P}_U(w))/N < -\ln(\mathbb{P}_X(w))/N + \epsilon$ for all $w \in (\mathcal{R}_F)^N$ and $N > N_0(X, \epsilon)$ [27, 28]. We optimize over an objective function that incorporates $\mathbb{P}_U$ and the presence of additive white Gaussian noise in the measurements:

$$\Psi^U(w) \triangleq -\ln(\mathbb{P}_U(w)) + c_2 \|y - \Phi w\|^2, \tag{7}$$

resulting in[5] $x_{MAP}^U \triangleq \arg\min_{w \in (\mathcal{R}_F)^N} \Psi^U(w)$. Our universal MAP estimator does not require $M \ll N$, and $x_{MAP}^U$ can be used in general linear inverse problems.

---

[5] This formulation of $x_{MAP}^U$ corresponds to a Lagrangian relaxation of the approach studied in [33, 34].

## C. Conjectured MSE performance

Donoho [29] showed for the scalar channel $y = x + z$ that: ($i$) the Kolmogorov sampler $x_{KS}$ (3) is drawn from the posterior distribution $\mathbb{P}_{X|Y}(x|y)$; and ($ii$) the MSE of this estimate $E_{X,Z,\Phi}[\|y - x_{KS}\|^2]$ is no greater than twice the MMSE. Based on this result, which requires a large reproduction alphabet, we now present a conjecture on the quality of the estimation $x_{MAP}^U$. Our conjecture is based on observing that $x_{MAP}^U$ also samples from the posterior distribution; some experimental evidence to assess this conjecture is presented in Figs. 3 and 5.

*Conjecture 1:* Assume that $\Phi \in \mathbb{R}^{M \times N}$ is an i.i.d. Gaussian measurement matrix where each entry has mean zero and variance $1/M$. Suppose that Condition 1 holds, the aspect ratio $R > 0$ in (2), and the noise $z \in \mathbb{R}^M$ is i.i.d. zero-mean Gaussian with finite variance. Then for all $\epsilon > 0$, the mean squared error of the universal MAP estimator $x_{MAP}^U$ satisfies

$$\frac{E_{X,Z,\Phi}\left[\|x - x_{MAP}^U\|^2\right]}{N} < \frac{2E_{X,Z,\Phi}\left[\|x - E_X[x|y,\Phi]\|^2\right]}{N} + \epsilon$$

for sufficiently large $N$.

## IV. FIXED REPRODUCTION ALPHABET ALGORITHM

Although the results of the previous section are theoretically appealing, a brute force optimization of $x_{MAP}^U$ is computationally intractable. Instead, we propose an algorithmic approach based on MCMC methods [39]. Our approach is reminiscent of the framework for lossy data compression [45–48].

## A. Universal compressor

We propose a universal lossless compression formulation following the conventions of Weissman and coauthors [45–47]. We refer to the estimate as $w$ in our algorithm. Our goal is to characterize $-\ln(\mathbb{P}_U(w))$, cf. (7). Although we are inspired by the Kolmogorov sampler approach [29], KC cannot be computed [21, 22], and we instead use empirical entropy. For stationary ergodic sources, the empirical entropy converges to the per-symbol entropy rate almost surely [21].

To define the empirical entropy, we first define the empirical symbol counts:

$$n_q(w, \alpha)[\beta] \triangleq |\{i \in [q+1, N] : w_{i-q}^{i-1} = \alpha, w_i = \beta\}|, \tag{8}$$

where $q$ is the context depth [28, 49], $\beta \in \mathcal{R}_F$, $\alpha \in (\mathcal{R}_F)^q$, $w_i$ is the $i^{th}$ symbol of $w$, and $w_i^j$ is the string comprising symbols $i$ through $j$ within $w$. We now define the order $q$ conditional empirical probability for the context $\alpha$ as

$$\mathbb{P}_q(w, \alpha)[\beta] \triangleq \frac{n_q(w, \alpha)[\beta]}{\sum_{\beta' \in \mathcal{R}_F} n_q(w, \alpha)[\beta']}, \tag{9}$$

and the order $q$ conditional empirical entropy,

$$H_q(w) \triangleq -\frac{1}{N} \sum_{\alpha \in (\mathcal{R}_F)^q, \beta \in \mathcal{R}_F} n_q(w, \alpha)[\beta] \log_2 \left(\mathbb{P}_q(w, \alpha)[\beta]\right), \tag{10}$$

where the sum is only over non-zero counts and probabilities.

Allowing the context depth $q \triangleq q_N = o(\log(N))$ to grow slowly with $N$, various universal compression algorithms can achieve the empirical entropy $H_q(\cdot)$ asymptotically [27, 28, 49]. On the other hand, no compressor can outperform the entropy rate. Additionally, for large $N$, the empirical symbol counts with context depth $q$ provide a sufficiently precise characterization of the source statistics. Therefore, $H_q$ provides a concise approximation to the per-symbol coding length of a universal compressor.

## B. Markov chain Monte Carlo

Having approximated the coding length, we now describe how to optimize our objective function. We define the energy $\Psi^{H_q}(w)$ in an analogous manner to $\Psi^U(w)$ (7), using $H_q(w)$ as our universal coding length:

$$\Psi^{H_q}(w) \triangleq NH_q(w) + c_4\|y - \Phi w\|^2, \tag{11}$$

where $c_4 = c_2 \log_2(e)$. The minimization of this energy is analogous to minimizing $\Psi^U(w)$.

Ideally, our goal is to compute the globally minimum energy solution $x_{MAP}^{H_q} \triangleq \arg\min_{w \in (\mathcal{R}_F)^N} \Psi^{H_q}(w)$. We use a stochastic MCMC relaxation [39] to achieve the globally minimum solution in the limit of infinite computation. To assist the reader in appreciating how MCMC is used to compute $x_{MAP}^{H_q}$, we include pseudocode for our approach in Algorithm 1. The algorithm, called basic MCMC (B-MCMC), will be used as a building block for our latter Algorithms 2 and 3 in Section V. The initial estimate $w$ is obtained by quantizing the *initial point* $x^* \in \mathbb{R}^N$ to $(\mathcal{R}_F)^N$. The initial point $x^*$ could be the output of any signal reconstruction algorithm, and because $x^*$ is a preliminary estimate of the signal that does not require high fidelity, we let $x^* = \Phi^T y$ for simplicity, where $(\cdot)^T$ denotes transpose. We refer to the processing of a single entry of $w$ as an iteration and group the processing of all entries of $w$, randomly permuted, into super-iterations.

The Boltzmann PMF is defined as

$$\mathbb{P}_s(w) \triangleq \frac{1}{\zeta_s} \exp(-s\Psi^{H_q}(w)), \tag{12}$$

where $s > 0$ is inversely related to the temperature in simulated annealing and $\zeta_s$ is a normalization constant. MCMC samples from the Boltzmann PMF (12) using a *Gibbs sampler*: in each iteration, a single element $w_n$ is generated while the rest of $w$, $w^{\backslash n} \triangleq \{w_i : n \neq i\}$, remains unchanged. We denote by $w_1^{n-1}\beta w_{n+1}^N$ the concatenation of the initial portion of the output vector $w_1^{n-1}$, the symbol $\beta \in \mathcal{R}_F$, and the latter portion of the output $w_{n+1}^N$. The Gibbs sampler updates $w_n$ by resampling from the PMF:

$$\begin{aligned}
\mathbb{P}_s(w_n = a | w^{\backslash n}) &= \frac{\exp\left(-s\Psi^{H_q}(w_1^{n-1}aw_{n+1}^N)\right)}{\sum_{b \in \mathcal{R}_F} \exp\left(-s\Psi^{H_q}(w_1^{n-1}bw_{n+1}^N)\right)} \\
&= \frac{1}{\sum_{b \in \mathcal{R}_F} \exp\left(-s\left[N\Delta H_q(w, n, b, a) + c_4\Delta d(w, n, b, a)\right]\right)},
\end{aligned} \tag{13}$$

where

$$\Delta H_q(w, n, b, a) \triangleq H_q(w_1^{n-1}bw_{n+1}^N) - H_q(w_1^{n-1}aw_{n+1}^N)$$

is the change in empirical entropy $H_q(w)$ (10) when $w_n = a$ is replaced by $b$, and

$$\Delta d(w, n, b, a) \triangleq \|y - \Phi(w_1^{n-1}bw_{n+1}^N)\|^2 - \|y - \Phi(w_1^{n-1}aw_{n+1}^N)\|^2 \tag{14}$$

---

**Algorithm 1** Basic MCMC for universal CS – Fixed alphabet

---

1: **Inputs**: Initial estimate $w$, reproduction alphabet $\mathcal{R}_F$, noise variance $\sigma_Z^2$, number of super–iterations $r$, temperature constant $c > 1$, and context depth $q$

2: Compute $n_q(w, \alpha)[\beta]$, $\forall\ \alpha \in (\mathcal{R}_F)^q$, $\beta \in \mathcal{R}_F$

3: **for** $t = 1$ to $r$ **do**    *// super-iteration*

4:     $s \leftarrow \ln(t)/(cN\Delta_q)$    *// $s = s_t$, cf. (16)*

5:     Draw permutation $\{1, \ldots, N\}$ at random

6:     **for** $t' = 1$ to $N$ **do**    *// iteration*

7:         Let $n$ be component $t'$ in permutation

8:         **for** all $\beta$ in $\mathcal{R}_F$ **do**    *// possible new $w_n$*

9:             Compute $\Delta H_q(w, n, \beta, w_n)$

10:             Compute $\Delta d(w, n, \beta, w_n)$

11:             Compute $\mathbb{P}_s(w_n = \beta | w^{\setminus n})$

12:         Generate $w_n$ using $\mathbb{P}_s(\cdot | w^{\setminus n})$    *// Gibbs*

13:         Update $n_q(w, \alpha)[\beta]$, $\forall\ \alpha \in (\mathcal{R}_F)^q$, $\beta \in \mathcal{R}_F$

14: **Output:** Return approximation $w$ of $x_{MAP}^U$

---

is the change in $\|y - \Phi w\|^2$ when $w_n = a$ is replaced by $b$. The maximum change in the energy within an iteration of Algorithm 1 is then bounded by

$$\Delta_q = \max_{1 \leq n \leq N} \max_{w \in (\mathcal{R}_F)^N} \max_{a, b \in \mathcal{R}_F} |N\Delta H_q(w, n, b, a) + c_4 \Delta d(w, n, b, a)|. \tag{15}$$

Note that $x$ is assumed bounded (cf. Condition 1) so that (14–15) are bounded as well.

In MCMC, the space $w \in (\mathcal{R}_F)^N$ is analogous to a statistical mechanical system, and at low temperatures the system tends toward low energies. Therefore, during the execution of the algorithm, we set a sequence of decreasing temperatures that takes into account the maximum change given in (15):

$$s_t \triangleq \ln(t + r_0)/(cN\Delta_q) \text{ for some } c > 1, \tag{16}$$

where $r_0$ is a temperature offset. At low temperatures, i.e., large $s_t$, a small difference in energy $\Psi^{H_q}(w)$ drives a big difference in probability, cf. (12). Therefore, we begin at a high temperature where the Gibbs sampler can freely move around $(\mathcal{R}_F)^N$. As the temperature is reduced, the PMF becomes more sensitive to changes in energy (12), and the trend toward $w$ with lower energy grows stronger. In each iteration, the Gibbs sampler modifies $w_n$ in a random manner that resembles heat bath concepts in statistical mechanics. Although MCMC could sink into a local minimum, Geman and Geman [39] proved that if we decrease the temperature according to (16), then the randomness of Gibbs sampling will eventually drive MCMC out of the local minimum toward the globally optimal $x_{MAP}^U$. In order to help B-MCMC approach the global minimum with reasonable runtime, we will refine B-MCMC in Section V.

The following theorem is proven in Appendix A, following the framework espoused by Jalali and Weissman [46, 47].

*Theorem 1:* Let $X$ be a stationary ergodic source that obeys Condition 1. Then the outcome $w^r$ of Algorithm 1 in the limit of an infinite number of super-iterations $r$ obeys $\lim_{r \to \infty} \Psi^{H_q}(w^r) = \min_{\widetilde{w} \in (\mathcal{R}_F)^N} \Psi^{H_q}(\widetilde{w}) = \Psi^{H_q}\left(x_{MAP}^{H_q}\right).$

Theorem 1 shows that Algorithm 1 matches the best-possible performance of the universal MAP estimator as measured by the objective function $\Psi^{H_q}$, which should yield an MSE that is twice the MMSE (cf. Conjecture 1). To gain some insight about the convergence process of MCMC, we focus on a fixed arbitrary sub-optimal sequence $w \in (\mathcal{R}_F)^N$. Suppose that at super-iteration $t$ the energy for the algorithm's output $\Psi^{H_q}(w)$ has converged to the steady state (see Appendix A for details on convergence). We can then focus on the probability ratio $\rho_t = \mathbb{P}_{s_t}(w)/\mathbb{P}_{s_t}(x_{MAP}^{H_q})$; $\rho_t < 1$ because $x_{MAP}^{H_q}$ is the global minimum and has the largest Boltzmann probability over all $w \in (\mathcal{R}_F)^N$, whereas $w$ is sub-optimal. We then consider the same sequence $w$ at super-iteration $t^2$; the inverse temperature is $2s_t$ and the corresponding ratio at super-iteration $t^2$ is (cf. (12))

$$\frac{\mathbb{P}_{2s_t}(w)}{\mathbb{P}_{2s_t}(x_{MAP}^{H_q})} = \frac{\exp(-2s_t \Psi^{H_q}(w))}{\exp(-2s_t \Psi^{H_q}(x_{MAP}^{H_q}))} = \left(\frac{\mathbb{P}_{s_t}(w)}{\mathbb{P}_{s_t}(x_{MAP}^{H_q})}\right)^2.$$

That is, between super-iterations $t$ and $t^2$ the probability ratio $\rho_t$ is also squared, and the Gibbs sampler is less likely to generate samples that differ from $x_{MAP}^{H_q}$. We infer from this argument that the probability concentration of our algorithm around the globally optimal $x_{MAP}^{H_q}$ is linear in the number of super-iterations.

## C. Computational challenges

Studying the pseudocode of Algorithm 1, we recognize that Lines 9–11 must be implemented efficiently, as they run $rN|\mathcal{R}_F|$ times. Lines 9 and 10 are especially challenging.

For Line 9, a naive update of $H_q(w)$ has complexity $O(|\mathcal{R}_F|^{q+1})$, cf. (10). To address this problem, Jalali and Weissman [46, 47] recompute the empirical conditional entropy in $O(q|\mathcal{R}_F|)$ time only for the $O(q)$ contexts whose corresponding counts are modified [46, 47]. The same approach can be used in Line 13, again reducing computation from $O(|\mathcal{R}_F|^{q+1})$ to $O(q|\mathcal{R}_F|)$. Some straightforward algebra allows us to convert Line 10 to a form that requires aggregate runtime of $O(Nr(M + |\mathcal{R}_F|))$. Combined with the computation for Line 9, and since $M \gg q|\mathcal{R}_F|^2$ (because $|\mathcal{R}_F| = \gamma^2, \gamma = \lceil \ln(N) \rceil, q = o(\log(N))$, and $M = O(N)$) in practice, the entire runtime of our algorithm is $O(rMN)$.

The practical value of Algorithm 1 may be reduced due to its high computational cost, dictated by the number of super-iterations $r$ required for convergence to $x_{MAP}^{H_q}$ and the large size of the reproduction alphabet. Nonetheless, Algorithm 1 provides a starting point toward further performance gains of more practical algorithms for computing $x_{MAP}^{H_q}$, which are presented in Section V. Furthermore, our experiments in Section VI will show that the performance of the algorithm of Section V is comparable to and in many cases better than existing algorithms.

## V. ADAPTIVE REPRODUCTION ALPHABET

While Algorithm 1 is a first step toward universal signal estimation in CS, $N$ must be large enough to ensure that $\mathcal{R}_F$ quantizes a broad enough range of values of $\mathbb{R}$ finely enough to represent the estimate $x_{MAP}^{H_q}$ well. For large $N$, the estimation performance using the reproduction alphabet (6) could suffer from high computational complexity. On the other hand, for small $N$ the number of reproduction levels employed is insufficient to obtain acceptable performance. Nevertheless, using an excessive number of levels will slow down the convergence. Therefore, in this section, we explore techniques that tailor the reproduction alphabet adaptively to the signal being observed.

### A. Adaptivity in reproduction levels

To estimate better with finite $N$, we utilize reproduction levels that are *adaptive* instead of the fixed levels in $\mathcal{R}_F$. To do so, instead of $w \in (\mathcal{R}_F)^N$, we optimize over a sequence $u \in \mathcal{Z}^N$, where $|\mathcal{Z}| < |\mathcal{R}_F|$ and $|\cdot|$ denotes the size. The new reproduction alphabet $\mathcal{Z}$ does not directly correspond to real numbers. Instead, there is an adaptive mapping $\mathcal{A} : \mathcal{Z} \to \mathbb{R}$, and the reproduction levels are $\mathcal{A}(\mathcal{Z})$. Therefore, we call $\mathcal{Z}$ the *adaptive* reproduction alphabet. Since the mapping $\mathcal{A}$ is one-to-one, we also refer to $\mathcal{Z}$ as reproduction levels. Considering the energy function (11), we now compute the empirical symbol counts $n_q(u, \alpha)[\beta]$, order $q$ conditional empirical probabilities $\mathbb{P}_q(u, \alpha)[\beta]$, and order $q$ conditional empirical entropy $H_q(u)$ using $u \in \mathcal{Z}^N$, $\alpha \in \mathcal{Z}^q$, and $\beta \in \mathcal{Z}$, cf. (8), (9), and (10). Similarly, we use $\|y - \Phi\mathcal{A}(u)\|^2$ instead of $\|y - \Phi w\|^2$, where $\mathcal{A}(u)$ is the straightforward vector extension of $\mathcal{A}$. These modifications yield an adaptive energy function $\Psi_a^{H_q}(u) \triangleq NH_q(u) + c_4\|y - \Phi\mathcal{A}(u)\|^2$.

We choose $\mathcal{A}_{opt}$ to optimize for minimum squared error,

$$\mathcal{A}_{opt} \triangleq \arg\min_{\mathcal{A}} \|y - \Phi\mathcal{A}(u)\|^2 = \arg\min_{\mathcal{A}} \left[\sum_{m=1}^{M} (y_m - [\Phi\mathcal{A}(u)]_m)^2\right],$$

where $[\Phi\mathcal{A}(u)]_m$ denotes the $m^{th}$ entry of the vector $\Phi\mathcal{A}(u)$. The optimal mapping depends entirely on $y$, $\Phi$, and $u$. From a coding perspective, describing $\mathcal{A}_{opt}(u)$ requires $H_q(u)$ bits for $u$ and $|\mathcal{Z}|b\log\log(N)$ bits for $\mathcal{A}_{opt}$ to match the resolution of the non-adaptive $\mathcal{R}_F$, with $b > 1$ an arbitrary constant [45]. The resulting coding length defines our universal prior.

**Optimization of reproduction levels:** We now describe the optimization procedure for $\mathcal{A}_{opt}$, which must be computationally efficient. Write

$$\Upsilon(\mathcal{A}) \triangleq \|y - \Phi\mathcal{A}(u)\|^2 = \sum_{m=1}^{M} \left(y_m - \sum_{n=1}^{N} \Phi_{mn}\mathcal{A}(u_n)\right)^2,$$

where $\Phi_{mn}$ is the entry of $\Phi$ at row $m$ and column $n$. For $\Upsilon(\mathcal{A})$ to be minimum, we need zero-valued derivatives

$$\frac{d\Upsilon(\mathcal{A})}{d\mathcal{A}(\beta)} = -2\sum_{m=1}^{M} \left(y_m - \sum_{n=1}^{N} \Phi_{mn}\mathcal{A}(u_n)\right)\left(\sum_{n=1}^{N} \Phi_{mn}1_{\{u_n=\beta\}}\right) = 0, \ \forall \ \beta \in \mathcal{Z}, \tag{17}$$

where $1_{\{A\}}$ is the indicator function for event $A$. Define the location sets $\mathcal{L}_\beta \triangleq \{n : 1 \le n \le N, u_n = \beta\}$ for each $\beta \in \mathcal{Z}$, and rewrite the derivatives of $\Upsilon(\mathcal{A})$,

$$\frac{d\Upsilon(\mathcal{A})}{d\mathcal{A}(\beta)} = -2\sum_{m=1}^{M} \left(y_m - \sum_{\lambda \in \mathcal{Z}}\sum_{n \in \mathcal{L}_\lambda} \Phi_{mn}\mathcal{A}(\lambda)\right)\left(\sum_{n \in \mathcal{L}_\beta} \Phi_{mn}\right). \tag{18}$$

Let the per-character sum column values be

$$\mu_{m\beta} \triangleq \sum_{n \in \mathcal{L}_\beta} \Phi_{mn}, \qquad (19)$$

for each $m \in \{1, \ldots, M\}$ and $\beta \in \mathcal{Z}$. We desire the derivatives to be zero, cf. (18):

$$0 = \sum_{m=1}^{M} \left( y_m - \sum_{\lambda \in \mathcal{Z}} \mathcal{A}(\lambda)\mu_{m\lambda} \right) \mu_{m\beta}.$$

Thus, the system of equations must be satisfied,

$$\sum_{m=1}^{M} y_m \mu_{m\beta} = \sum_{m=1}^{M} \left( \sum_{\lambda \in \mathcal{Z}} \mathcal{A}(\lambda)\mu_{m\lambda} \right) \mu_{m\beta} \qquad (20)$$

for each $\beta \in \mathcal{Z}$. Consider now the right hand side,

$$\sum_{m=1}^{M} \left( \sum_{\lambda \in \mathcal{Z}} \mathcal{A}(\lambda)\mu_{m\lambda} \right) \mu_{m\beta} = \sum_{\lambda \in \mathcal{Z}} \mathcal{A}(\lambda) \sum_{m=1}^{M} \mu_{m\lambda}\mu_{m\beta},$$

for each $\beta \in \mathcal{Z}$. The system of equations can be described in matrix form

$$\overbrace{\begin{bmatrix} \sum_{m=1}^{M} \mu_{m\beta_1}\mu_{m\beta_1} & \cdots & \sum_{m=1}^{M} \mu_{m\beta_{|\mathcal{Z}|}}\mu_{m\beta_1} \\ \vdots & \ddots & \vdots \\ \sum_{m=1}^{M} \mu_{m\beta_1}\mu_{m\beta_{|\mathcal{Z}|}} & \cdots & \sum_{m=1}^{M} \mu_{m\beta_{|\mathcal{Z}|}}\mu_{m\beta_{|\mathcal{Z}|}} \end{bmatrix}}^{\Omega} \overbrace{\begin{bmatrix} \mathcal{A}(\beta_1) \\ \vdots \\ \mathcal{A}(\beta_{|\mathcal{Z}|}) \end{bmatrix}}^{\mathcal{A}(\mathcal{Z})} = \overbrace{\begin{bmatrix} \sum_{m=1}^{M} y_m \mu_{m\beta_1} \\ \vdots \\ \sum_{m=1}^{M} y_m \mu_{m\beta_{|\mathcal{Z}|}} \end{bmatrix}}^{\Theta}. \qquad (21)$$

Note that by writing $\mu$ as a matrix with entries indexed by row $m$ and column $\beta$ given by (19), we can write $\Omega$ as a Gram matrix, $\Omega = \mu^T \mu$, and we also have $\Theta = \mu^T y$, cf. (20). The optimal $\mathcal{A}$ can be computed as a $|\mathcal{Z}| \times 1$ vector $\mathcal{A}_{opt} = \Omega^{-1}\Theta = (\mu^T \mu)^{-1}\mu^T y$ if $\Omega \in \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{Z}|}$ is invertible. We note in passing that numerical stability can be improved by regularizing $\Omega$. Note also that

$$\|y - \Phi\mathcal{A}(u)\|^2 = \sum_{m=1}^{M} \left( y_m - \sum_{\beta \in \mathcal{Z}} \mu_{m\beta}\mathcal{A}_{opt}(\beta) \right)^2, \qquad (22)$$

which can be computed in $O(M|\mathcal{Z}|)$ time instead of $O(MN)$.

**Computational complexity:** Pseudocode for level-adaptive MCMC (L-MCMC) appears in Algorithm 2, which resembles Algorithm 1. The initial mapping $\mathcal{A}$ is inherited from a quantization of the initial point $x^*$, $r_0 = 0$ ($r_0$ takes different values in Section V-B), and other minor differences between B-MCMC and L-MCMC appear in lines marked by asterisks.

We discuss computational requirements for each line of the pseudocode that is run within the inner loop.

- Line 10 can be computed in $O(q|\mathcal{Z}|)$ time (see discussion of Line 9 of B-MCMC in Section IV-C).
- Line 11 updates $\mu_{m\beta}$ for $m = 1, ..., M$ in $O(M)$ time.
- Line 12 updates $\Omega$. Because we only need to update $O(1)$ columns and $O(1)$ rows, each such column and row contains $O(|\mathcal{Z}|)$ entries, and each entry is a sum over $O(M)$ terms, we need $O(M|\mathcal{Z}|)$ time.
- Line 13 requires inverting $\Omega$ in $O(|\mathcal{Z}|^3)$ time.
- Line 14 requires $O(M|\mathcal{Z}|)$ time, cf. (22).

---

**Algorithm 2** Level-adaptive MCMC

---

1: *\*Inputs*: Initial mapping $\mathcal{A}$, sequence $u$, adaptive alphabet $\mathcal{Z}$, noise variance $\sigma_Z^2$, number of super-iterations $r$,

   temperature constant $c > 1$, context depth $q$, and temperature offset $r_0$

2: Compute $n_q(u, \alpha)[\beta]$, $\forall\ \alpha \in \mathcal{Z}^q$, $\beta \in \mathcal{Z}$

3: \*Initialize $\Omega$

4: **for** $t = 1$ to $r$ **do**    // *super-iteration*

5:     $s \leftarrow \ln(t + r_0)/(cN\Delta_q)$    // $s = s_t$, *cf. (16)*

6:     Draw permutation $\{1, \ldots, N\}$ at random

7:     **for** $t' = 1$ to $N$ **do**    // *iteration*

8:         Let $n$ be component $t'$ in permutation

9:         **for** all $\beta$ in $\mathcal{Z}$ **do**    // *possible new $u_n$*

10:             Compute $\Delta H_q(u, n, \beta, u_n)$

11:             \*Compute $\mu_{m\beta}, \forall\ m \in \{1, \ldots, M\}$

12:             \*Update $\Omega$    // $O(1)$ *rows and columns*

13:             \*Compute $\mathcal{A}_{opt}$    // *invert* $\Omega$

14:             Compute $\|y - \Phi\mathcal{A}(u_1^{n-1}\beta u_{n+1}^N)\|^2$

15:             Compute $\mathbb{P}_s(u_n = \beta | u^{\backslash n})$

16:         \*$\widetilde{u}_n \leftarrow u_n$    // *save previous value*

17:         Generate $u_n$ using $\mathbb{P}_s(\cdot | u^{\backslash n})$    // *Gibbs*

18:         Update $n_q(\cdot)[\cdot]$ at $O(q)$ relevant locations

19:         \*Update $\mu_{m\beta}$, $\forall\ m$, $\beta \in \{u_n, \widetilde{u}_n\}$

20:         \*Update $\Omega$    // $O(1)$ *rows and columns*

21: *\*Outputs*: Return approximation $\mathcal{A}(u)$ of $x_{MAP}^U$, $\mathcal{Z}$, and temperature offset $r_0 + r$

---

- Line 15 requires $O(|\mathcal{Z}|)$ time.

In practice we typically have $M \gg |\mathcal{Z}|^2$, and so the aggregate complexity is $O(rMN|\mathcal{Z}|)$, which is greater than the computational complexity of Algorithm 1 by a factor of $O(|\mathcal{Z}|)$.

### B. Adaptivity in reproduction alphabet size

While Algorithm 2 adaptively maps $u$ to $\mathbb{R}^N$, the signal estimation quality heavily depends on $|\mathcal{Z}|$. Denote the true alphabet of the signal by $\mathcal{X}$, $x \in \mathcal{X}^N$; if the signal is continuous-valued, then $|\mathcal{X}|$ is infinite. Ideally we want to employ as many levels as the runtime allows for continuous-valued signals, whereas for discrete-valued signals we want $|\mathcal{Z}| = |\mathcal{X}|$. Inspired by this observation, we propose to begin with some initial $|\mathcal{Z}|$, and then adaptively adjust $|\mathcal{Z}|$ hoping to match $|\mathcal{X}|$. Hence, we propose the size- and level-adaptive MCMC algorithm (Algorithm 3), which invokes L-MCMC (Algorithm 2) several times.
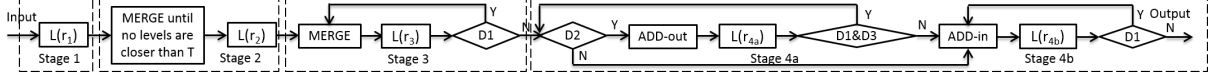
Fig. 2. *Flowchart of Algorithm 3 (size- and level-adaptive MCMC). L(r) denotes running L-MCMC for r super-iterations. The parameters $r_1,r_2,r_3,r_{4a}$, and $r_{4b}$ are the number of super-iterations used in Stages 1 through 4, respectively. Criteria $D1 - D3$ are described in the text.*

**Three basic procedures:** In order to describe the size- and level-adaptive MCMC (SLA-MCMC) algorithm in detail, we introduce three alphabet adaptation procedures as follows.

- *MERGE*: First, find the closest adjacent levels $\beta_1, \beta_2 \in \mathcal{Z}$. Create a new level $\beta_3$ and add it to $\mathcal{Z}$. Let $\mathcal{A}(\beta_3) = (\mathcal{A}(\beta_1) + \mathcal{A}(\beta_2))/2$. Replace $u_i$ by $\beta_3$ whenever $u_i \in \{\beta_1, \beta_2\}$. Next, remove $\beta_1$ and $\beta_2$ from $\mathcal{Z}$.

- *ADD-out*: Define the range $R_{\mathcal{A}} = [\min \mathcal{A}(\mathcal{Z}), \max \mathcal{A}(\mathcal{Z})]$, and $\mathcal{I}_{R_{\mathcal{A}}} = \max \mathcal{A}(\mathcal{Z}) - \min \mathcal{A}(\mathcal{Z})$. Add a *lower* level $\beta_3$ and/or *upper level* $\beta_4$ to $\mathcal{Z}$ with $\mathcal{A}(\beta_3) = \min \mathcal{A}(\mathcal{Z}) - \dfrac{\mathcal{I}_{R_{\mathcal{A}}}}{|\mathcal{Z}| - 1}, \mathcal{A}(\beta_4) = \max \mathcal{A}(\mathcal{Z}) + \dfrac{\mathcal{I}_{R_{\mathcal{A}}}}{|\mathcal{Z}| - 1}$. Note that $|\{u_i : u_i = \beta_3 \text{ or } \beta_4, i = 1, ..., N\}| = 0$, i.e., the new levels are empty.

- *ADD-in*: First, find the most distant adjacent levels, $\beta_1$ and $\beta_2$. Then, add a level $\beta_3$ to $\mathcal{Z}$ with $\mathcal{A}(\beta_3) = (\mathcal{A}(\beta_1) + \mathcal{A}(\beta_2))/2$. For $i \in \{1, ..., |\mathcal{Z}|\}$ s.t. $u_i = \beta_1$, replace $u_i$ by $\beta_3$ with probability $\mathbb{P}_s(u_i = \beta_2)/(\mathbb{P}_s(u_i = \beta_1) + \mathbb{P}_s(u_i = \beta_2))$, where $\mathbb{P}_s$ is given in (12); for $i \in \{1, ..., |\mathcal{Z}|\}$ s.t. $u_i = \beta_2$, replace $u_i$ by $\beta_3$ with probability $\mathbb{P}_s(u_i = \beta_1)/(\mathbb{P}_s(u_i = \beta_1) + \mathbb{P}_s(u_i = \beta_2))$. Note that $|\{u_i : u_i = \beta_3, i = 1, ..., N\}|$ is typically non-zero, i.e., $\beta_3$ tends not to be empty.

We call the process of running one of these procedures followed by running L-MCMC a *round*.

**Size- and level-adaptive MCMC:** SLA-MCMC is conceptually illustrated in the flowchart in Fig. 2. It has four stages, and in each stage we will run L-MCMC for several super-iterations; we denote the execution of L-MCMC for $r$ super-iterations by L(r). The parameters $r_1, r_2, r_3, r_{4a}$, and $r_{4b}$ are the number of super-iterations used in Stages 1 through 4, respectively. The choice of these parameters reflects a trade-off between runtime and estimation quality.

In Stage 1, SLA-MCMC uses a fixed-size adaptive reproduction alphabet $\mathcal{Z}$ to tentatively estimate the signal. The initial point of Stage 1 is obtained in the same way as L-MCMC. After Stage 1, the initial point and temperature offset for each instance of L-MCMC correspond to the respective outputs of the previous instance of L-MCMC. If the source is discrete-valued and $|\mathcal{Z}| > |\mathcal{X}|$ in Stage 1, then multiple levels in the output $\mathcal{Z}$ of Stage 1 may correspond to a single level in $\mathcal{X}$. To alleviate this problem, in Stage 2 we merge levels closer than $T = \mathcal{I}_{R_{\mathcal{A}}}/(K_1 \times (|\mathcal{Z}| - 1))$, where $K_1$ is a parameter.

However, $|\mathcal{Z}|$ might still be larger than needed; hence in Stage 3 we tentatively merge the closest adjacent levels. The criterion $D1$ evaluates whether the current objective function is lower (better) than in the previous round; we do not leave Stage 3 until $D1$ is violated. Note that if $|\mathcal{X}| > |\mathcal{Z}|$ (this always holds for continuous-valued signals), then ideally SLA-MCMC should not merge any levels in Stage 3, because the objective function would increase if we merge any levels.

Define the outlier set $S = \{x_i : x_i \notin R_{\mathcal{A}}, i = 1, ..., N\}$. Under Condition 1, $S$ might be small or even empty. When $S$ is small, L-MCMC might not assign levels to represent the entries of $S$. To make SLA-MCMC more robust

to outliers, in Stage 4a we add empty levels outside the range $R_{\mathcal{A}}$ and then allow L-MCMC to change entries of $u$ to the new levels during Gibbs sampling; we call this *populating* the new levels. If a newly added outside level is not populated, then we remove it from $\mathcal{Z}$. Seeing that the optimal mapping $\mathcal{A}_{opt}$ in L-MCMC tends not to map symbols to levels with low population, we consider a criterion $D2$ where we will will add an outside upper (lower) level if the population of the current upper (lower) level is smaller than $N/(K_2|\mathcal{Z}|)$, where $K_2$ is a parameter. That is, the criterion $D2$ is violated if both populations of the current upper and lower levels are sufficient (at least $N/(K_2|\mathcal{Z}|)$); in this case we do not need to add outside levels because $\mathcal{A}_{opt}$ will map some of the current levels to represent the entries in $S$. The criterion $D3$ is violated if all levels added outside are not populated by the end of the round. SLA-MCMC keeps adding levels outside $R_{\mathcal{A}}$ until it is wide enough to cover most of the entries of $x$.

Next, SLA-MCMC considers adding levels inside $R_{\mathcal{A}}$ (Stage 4b). If the signal is discrete-valued, this stage should stop when $|\mathcal{Z}| = |\mathcal{X}|$. Else, for continuous-valued signals SLA-MCMC can add levels until the runtime expires.

In practice, SLA-MCMC runs L-MCMC at most a constant number of times, and the computational complexity is in the same order of L-MCMC, i.e., $O(rMN|\mathcal{Z}|)$. On the other hand, SLA-MCMC allows varying $|\mathcal{Z}|$, which often improves the estimation quality.

*C. Mixing*

Donoho proved for the scalar channel setting that $x_{KS}$ is sampled from the posterior $\mathbb{P}_{X|Y}(x|y)$ [29]. Seeing that the Gibbs sampler used by MCMC (cf. Section IV-B) generates random samples, and the outputs of our algorithm will be different if its random number generator is initialized with different *random seeds*, we speculate that running SLA-MCMC several times will also yield independent samples from the posterior, where we note that the runtime grows linearly in the number of times that we run SLA-MCMC. By mixing (averaging over) several outputs of SLA-MCMC, we obtain $\widehat{x}_{\text{avg}}$, which may have lower squared error with respect to the true $x$ than the average squared error obtained by a single SLA-MCMC output. Numerical results suggest that mixing indeed reduces the MSE (cf. Fig. 7); this observation suggests that mixing the outputs of multiple algorithms, including running a random reconstruction algorithm several times, may reduce the squared error.

## VI. NUMERICAL RESULTS

In this section, we demonstrate that SLA-MCMC is comparable and in many cases better than existing algorithms in reconstruction quality, and that SLA-MCMC is applicable when $M > N$. Additionally, some numerical evidence is provided to justify Conjecture 1 in Section III-C. Then, the advantage of SLA-MCMC in estimating low-complexity signals is demonstrated. Finally, we compare B-MCMC, L-MCMC, and SLA-MCMC performance.

We implemented SLA-MCMC in Matlab[6] and tested it using several stationary ergodic sources. Except when noted, for each source, signals $x$ of length $N = 10000$ were generated. Each such $x$ was multiplied by a Gaussian random matrix $\Phi$ with normalized columns and corrupted by i.i.d. Gaussian measurement noise $z$. Except when noted, the number of measurements $M$ varied between 2000 and 7000. The noise variance $\sigma_Z^2$ was selected to ensure

---

[6]A toolbox that runs the simulations in this paper is available at http://people.engr.ncsu.edu/dzbaron/software/UCS_BaronDuarte/
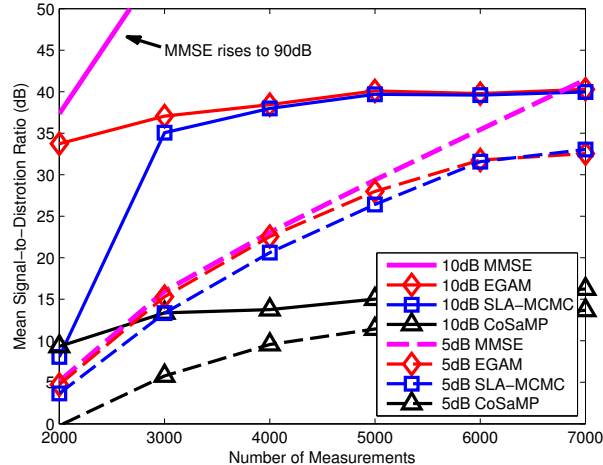
Fig. 3. *SLA-MCMC, EGAM, and CoSaMP estimation results for a source with i.i.d. Bernoulli entries with non-zero probability of 3% as a function of the number of Gaussian random measurements $M$ for different SNR values ($N = 10000$).*

that the signal-to-noise ratio (SNR) was 5 or 10 dB; SNR was defined as $\text{SNR} = 10 \log_{10} \left[ (NE[x^2])/(M\sigma_Z^2) \right]$. We set the context depth $q = 2$, the number of super-iterations in different stages of SLA-MCMC $r_1 = 50$ and $r_2 = r_3 = r_{4a} = r_{4b} = 10$, the maximum total number of super-iterations to be 240, the initial number of levels $|\mathcal{Z}| = 7$, and the tuning parameter from Section V-B $K_1, K_2 = 10$. SLA-MCMC was not given the true alphabet $\mathcal{X}$ for any of the sources presented in this paper; our expectation is that it should adaptively adjust $|\mathcal{Z}|$ to match $|\mathcal{X}|$. The final estimate $\widehat{x}_{\text{avg}}$ of each signal was obtained by averaging over the outputs $\widehat{x}$ of 5 runs of SLA-MCMC, where in each run we initialized the random number generator with another random seed, cf. Section V-C. These choices of parameters seemed to provide a reasonable compromise between runtime and estimation quality.

We chose our performance metric as the mean signal-to-distortion ratio (MSDR) defined as

$$\text{MSDR} = 10 \log_{10} \left( E[x^2]/\text{MSE} \right).$$

For each $M$ and SNR, the MSE was obtained after averaging over the squared errors of $\widehat{x}_{\text{avg}}$ for 50 draws of $x$, $\Phi$, and $z$. We compared the performance of SLA-MCMC to that of (*i*) compressive sensing matching pursuit (CoSaMP) [40], a greedy method; (*ii*) gradient projection for sparse reconstruction (GPSR) [7], an optimization-based method; and (*iii*) message passing approaches (for each source, we chose best-matched algorithms between EM-GM-AMP-MOS (EGAM for short) [13] and turboGAMP (tG for short) [14]). Typical runtimes are 1 hour (for continuous-valued signals) and 15 minutes (discrete-valued) per random seed for SLA-MCMC, 30 minutes for EGAM [13] and tG [14], and 10 minutes for CoSaMP [40] and GPSR [7].

Among these baseline algorithms designed for i.i.d. signals, GPSR [7] and EGAM [13] only need $y$ and $\Phi$, and CoSaMP [40] also needs the number of non-zeros in $x$. Only tG [14] is designed for non-i.i.d. signals; however, it must be aware of the probabilistic model of the source. Finally, GPSR [7] performance was similar to that of CoSaMP [40] for all sources considered in this section, and thus is not plotted.
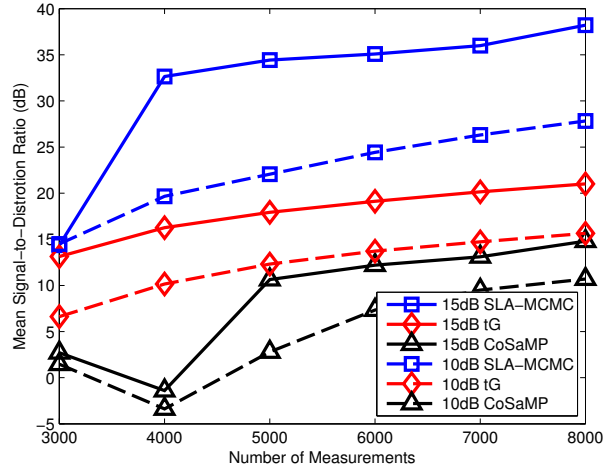
Fig. 4. *SLA-MCMC, tG, and CoSaMP estimation results for a dense two-state Markov source with non-zero entries drawn from a Rademacher ($\pm 1$) distribution as a function of the number of Gaussian random measurements $M$ for different SNR values ($N = 5000$).*

## A. Performance on discrete-valued sources

**Bernoulli source:** We first present results for an i.i.d. Bernoulli source. The Bernoulli source followed the distribution $f_X(x) = 0.03\delta(x-1) + 0.97\delta(x)$, where $\delta(\cdot)$ is the Dirac delta function. Note that SLA-MCMC did not know $\mathcal{X} = \{0, 1\}$ and had to estimate it on the fly. We chose EGAM [13] for message passing algorithms because it fits the signal with Gaussian mixtures (GM), which can accurately characterize signals from an i.i.d. Bernoulli source. The resulting MSDRs for SLA-MCMC, EGAM [13], and CoSaMP [40] are plotted in Fig. 3. We can see that when SNR = 5 dB, EGAM [13] approaches the MMSE [50] performance for low to medium $M$; although SLA-MCMC is often worse than EGAM [13], it is within 3 dB of the MMSE performance. This observation that SLA-MCMC approaches the MMSE for SNR = 5 dB partially substantiates Conjecture 1 in Section III-C. When SNR = 10 dB, SLA-MCMC is comparable to EGAM [13] when $M \geq 3000$. CoSaMP [40] has worse MSDR.

**Dense Markov-Rademacher source:** Considering that most algorithms are designed for i.i.d. sources, we now illustrate the performance of SLA-MCMC on non-i.i.d. sources by simulating a dense Markov-Rademacher (MRad for short) source. The non-zero entries of the dense MRad signal were generated by a two-state Markov state machine (non-zero and zero states). The transition from zero to non-zero state for adjacent entries had probability $\mathbb{P}_{01} = \frac{3}{70}$, while the transition from non-zero to zero state for adjacent entries had probability $\mathbb{P}_{10} = 0.10$; these parameters yielded 30% non-zero entries on average. The non-zeros were drawn from a Rademacher distribution, which took values $\pm 1$ with equal probability. With such denser signals, we may need to take more measurements and/or require higher SNRs to achieve similar performance to previous examples. Seeing that the runtime of CoSaMP [40] is cubic in $M$ and the number of non-zeros in the signal, we only generated dense MRad signals with length $N = 5000$ to save runtime. The number of measurements varied from 3000 to 8000, with SNR = 10 and 15 dB. Although tG [14] does not provide an option that accurately characterize the MRad source, we still chose to compare against its
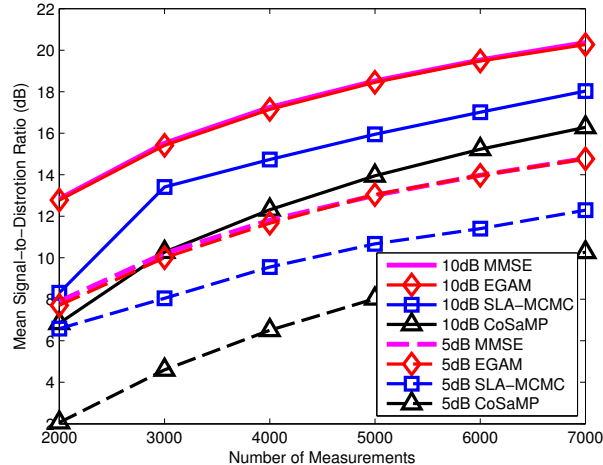
Fig. 5. *SLA-MCMC, EGAM, and CoSaMP estimation results for an i.i.d. sparse Laplace source as a function of the number of Gaussian random measurements $M$ for different SNR values ($N = 10000$).*

performance because it is applicable to non-i.i.d. signals. The MSDRs for SLA-MCMC, tG [14], and CoSaMP [40] are plotted in Fig. 4. We can see that CoSaMP [40] has poor performance because it does not consider the source memory;[7] although tG [14] is designed for non-i.i.d. sources, it is nonetheless outperformed by SLA-MCMC. This example shows that SLA-MCMC reconstructs non-i.i.d. signals well and is applicable to general linear inverse problems.

### B. Performance on continuous sources

We now discuss the performance of SLA-MCMC in estimating continuous sources.

**Sparse Laplace (i.i.d.) source:** For unbounded continuous-valued signals, which do not adhere to Condition 1, we simulated an i.i.d. sparse Laplace source following the distribution $f_X(x) = 0.03\mathcal{L}(0,1) + 0.97\delta(x)$, where $\mathcal{L}(0,1)$ denotes a Laplacian distribution with mean zero and variance one. We chose EGAM [13] for message passing algorithms because it fits the signal with GM, which can accurately characterize signals from an i.i.d. sparse Laplace source. The MSDRs for SLA-MCMC, EGAM [13], and CoSaMP [40] are plotted in Fig. 5. We can see that EGAM [13] approaches the MMSE [50] performance in all settings; SLA-MCMC outperforms CoSaMP [40], while it is approximately 2 dB worse than the MMSE. Recall from Conjecture 1 that we expect to achieve twice the MMSE, which is approximately 3 dB below the signal-to-distortion ratio of MMSE, and thus SLA-MCMC performance is reasonable. This example of SLA-MCMC performance approaching the MMSE further substantiates Conjecture 1.

**Markov-Uniform source:** For bounded continuous-valued signals, which adhere to Condition 1, we simulated a Markov-Uniform (MUnif for short) source, whose non-zero entries were generated by a two-state Markov state machine (non-zero and zero states) with $\mathbb{P}_{01} = \frac{3}{970}$ and $\mathbb{P}_{10} = 0.10$; these parameters yielded $3\%$ non-zeros entries on average. The non-zero entries were drawn from a uniform distribution between $0$ and $1$. We chose tG with Markov support and GM model options [14] for message passing algorithms. We plot the resulting MSDRs for

---

[7]We believe that the unexpected dip in each of the curves for CoSaMP [40] was due to its least-squares estimation step.
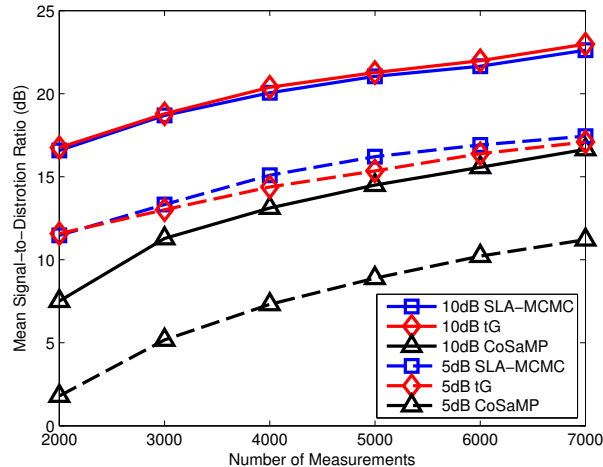
Fig. 6. *SLA-MCMC, tG, and CoSaMP estimation results for a two-state Markov source with non-zero entries drawn from a uniform distribution $U[0, 1]$ as a function of the number of Gaussian random measurements $M$ for different SNR values ($N = 10000$).*

SLA-MCMC, tG [14], and CoSaMP [40] in Fig. 6. We can see that the CoSaMP [40] lags behind in MSDR. The SLA-MCMC curve is close to that of tG [14] when SNR = 10 dB, and it is slightly better than tG [14] when SNR = 5 dB.

When the signal model is known, the message passing approaches EGAM [13] and tG [14] achieve quite low MSE's, because they can get close to the Bayesian MMSE. Sometimes the model is only known imprecisely, and SLA-MCMC can improve over message passing; for example, it is better than tG [14] in estimating MUnif signals (Fig. 6), because tG [14] approximates the uniformly distributed non-zeros by GM.

### C. Comparison between discrete and continuous sources

When the source is continuous (Figs. 5 and 6), SLA-MCMC might be worse than the existing message passing approaches (EGAM [13] and tG [14]). One reason for the under-performance of SLA-MCMC is the 3 dB gap of Conjecture 1. The second reason is that SLA-MCMC can only assign finitely many levels to approximate continuous-valued signals, leading to under-representation of the signal. However, when it comes to discrete-valued signals that have finite size alphabets (Figs. 3 and 4), SLA-MCMC is comparable to and in many cases better than existing algorithms.

### D. Performance on low-complexity signals

SLA-MCMC promotes low complexity due to the complexity-penalized term in the objective function (11). Hence, it tends to perform well for signals with low complexity such as the signals in Figs. 3 and 4 (note that the Bernoulli signal is sparse while the MRad signal is denser). In this subsection, we simulated a non-sparse low-complexity signal. We show that complexity-penalized approaches such as SLA-MCMC might estimate low-complexity signals well.

**Four-state Markov source:** To evaluate the performance of SLA-MCMC for discrete-valued non-i.i.d. and non-sparse signals, we examined a four-state Markov source (Markov4 for short), which has been discussed in Section I
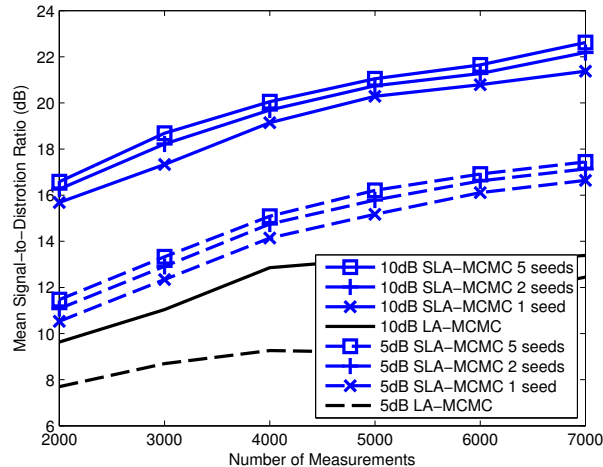
Fig. 7. *SLA-MCMC with different number of random seeds and L-MCMC estimation results for the Markov-Uniform source described in Fig. 6 as a function of the number of Gaussian random measurements $M$ for different SNR values ($N = 10000$).*

and Fig. 1. We tried conventional sparsifying transforms, but none of them yielded coefficients that were reasonably sparse. As pointed out in Section I, there exists a special transform that yields a sparse representation of the Markov4 signals, but using specialized transforms requires prior knowledge of the source. Despite knowing the sparsifying matrix, all our baseline algorithms performed poorly (yielding MSDRs below 5 dB). Thus, we did not include the results for these baseline algorithms in Fig. 1. On the other hand, Markov4 signals have low complexity in the time domain, and hence, SLA-MCMC successfully reconstructed Markov4 signals with reasonable quality even when $M$ was relatively small. This Markov4 source highlights the special advantage of our approach in reconstructing low-complexity signals.

### E. Comparison of B-MCMC, L-MCMC, and SLA-MCMC

We compare the performance of B-MCMC, L-MCMC, and SLA-MCMC with different numbers of seeds (cf. Section V-C) by examining the MUnif source (cf. Section VI-B). We ran B-MCMC with the fixed uniform alphabet $\mathcal{R}_F$ in (6) with $|\mathcal{R}_F| = 10$ levels. L-MCMC was initialized in the same way as Stage 1 of SLA-MCMC. B-MCMC and L-MCMC ran for 100 super-iterations before outputting the estimates; this number of super-iterations was sufficient because it was greater than $r_1 = 50$ in Stage 1 of SLA-MCMC. The results are plotted in Fig. 7. B-MCMC did not perform well given the $\mathcal{R}_F$ in (6) and is not plotted. We can see that SLA-MCMC outperforms L-MCMC. Averaging over more seeds provides an increase of 1 dB in MSDR.[8] Finally, we tried a "good" reproduction alphabet in B-MCMC, $\widetilde{\mathcal{R}}_F = \dfrac{1}{|\mathcal{R}_F| - 1/2}\{0, ..., |\mathcal{R}_F| - 1\}$, and the results were close to those of SLA-MCMC. Indeed, B-MCMC is quite sensitive to the reproduction alphabet, and Stages 2–4 of SLA-MCMC find a good set of levels. Example output levels $\mathcal{A}(\mathcal{Z})$ of SLA-MCMC were: $\{-0.001, 0.993\}$ for Bernoulli signals, $\{-0.998, 0.004, 1.004\}$ for dense MRad signals, 21 levels spread in the range $[-3.283, 4.733]$ for i.i.d. sparse Laplace signals, 22 levels spread in the range $[-0.000, 0.955]$ for MUnif signals, and $\{-1.010, 0.996\}$ for Markov4 signals; we can see that

---

[8]For other sources, we observed an increase in MSDR of up to 2 dB.

SLA-MCMC adaptively adjusted $|\mathcal{Z}|$ to match $|\mathcal{X}|$ so that these levels represented each signal well. Also, we can see from Figs. 3–5 that SLA-MCMC did not perform well in the low measurements and high SNR setting, which was due to mismatch between $|\mathcal{Z}|$ and $|\mathcal{X}|$.

## VII. CONCLUSIONS

This paper provides universal algorithms for signal estimation from linear measurements. Here, universality denotes the property that the algorithm need not be informed of the probability distribution for the recorded signal prior to acquisition; rather, the algorithm simultaneously builds estimates both of the observed signal and its distribution. Inspired by the Kolmogorov sampler [29] and motivated by the need for a computationally tractable framework, our contribution focuses on stationary ergodic signal sources and relies on a MAP estimation algorithm. The algorithm is then implemented via a MCMC formulation that is proven to be convergent in the limit of infinite computation. We reduce the computation and improve the estimation quality of the proposed algorithm by adapting the reproduction alphabet to match the complexity of the input signal. Our experiments have shown that the performance of the proposed algorithm is comparable to and in many cases better than existing algorithms, particularly for low-complexity sources that do not exhibit standard sparsity or compressibility.

As we were finishing this paper, Jalali and Poor [51] have independently shown that our formulation (11) also provides an implementable version of Rényi entropy minimization. Their theoretical findings further motivate our proposed universal MCMC formulation.

## APPENDIX A. PROOF OF THEOREM 1

Our proof mimics a very similar proof presented in [46, 47] for lossy source coding; we include all details for completeness. The proof technique relies on mathematical properties of non-homogeneous (e.g., time-varying) Markov Chains (MCs) [52]. Through the proof, $\mathcal{S} \triangleq (\mathcal{R}_F)^N$ denotes the state space of the MC of codewords generated by Algorithm 1, with size $|\mathcal{S}| = |\mathcal{R}_F|^N$. We define a stochastic transition matrix $P_{(t)}$ from $\mathcal{S}$ to itself given by the Boltzmann distribution for super-iteration $t$ in Algorithm 1. Similarly, $\pi_{(t)}$ defines the stable-state distribution on $\mathcal{S}$ for $P_{(t)}$, satisfying $\pi_{(t)} P_{(t)} = \pi_{(t)}$.

*Definition 1:* [52] *Dobrushin's ergodic coefficient* of a MC transition matrix $P$ is denoted by $\xi(P)$ and defined as $\xi(P) \triangleq \max_{1 \leq i,j \leq N} \frac{1}{2} \|p_i - p_j\|_1$, where $p_i$ denotes row $i$ of $P$.

From the definition, $0 \leq \xi(P) \leq 1$. Moreover, the ergodic coefficient can be rewritten as

$$\xi(P) = 1 - \min_{1 \leq i,j \leq N} \sum_{k=1}^{N} \min(p_{ik}, p_{jk}), \tag{23}$$

where $p_{ij}$ denotes the entry of $P$ at row $i$ and column $j$.

We group the product of transition matrices across super-iterations as $P_{(t_1 \rightarrow t_2)} = \prod_{t=t_1}^{t_2} P_{(t)}$. There are two common characterizations for the stable-state behavior of a non-homogeneous MC.

*Definition 2:* [52] A non-homogeneous MC is called *weakly ergodic* if for any distributions $\eta$ and $\nu$ over the state space $\mathcal{S}$, and any $t_1 \in \mathbb{N}$, $\limsup_{t_2 \rightarrow \infty} \|\eta P_{(t_1 \rightarrow t_2)} - \nu P_{(t_1 \rightarrow t_2)}\|_1 = 0$, where $\| \cdot \|_1$ denotes the $\ell_1$ norm. Similarly, a non-homogeneous MC is called *strongly ergodic* if there exists a distribution $\pi$ over the state space

$\mathcal{S}$ such that for any distribution $\eta$ over $\mathcal{S}$, and any $t_1 \in \mathbb{N}$, $\limsup_{t_2 \to \infty} \|\eta P_{(t_1 \to t_2)} - \pi\|_1 = 0$. We will use the following two theorems from [52] in our proof.

*Theorem 2:* [52] A MC is weakly ergodic if and only if there exists a sequence of integers $0 \leq t_1 \leq t_2 \leq \ldots$ such that $\sum_{i=1}^{\infty} \left(1 - \xi\left(P_{(t_i \to t_{i+1})}\right)\right) = \infty$.

*Theorem 3:* [52] Let a MC be weakly ergodic. Assume that there exists a sequence of probability distributions $\{\pi_{(t)}\}_{i=1}^{\infty}$ on the state space $\mathcal{S}$ such that $\pi_{(t)} P_{(t)} = \pi_{(t)}$. Then the MC is strongly ergodic if $\sum_{t=1}^{\infty} \|\pi_{(t)} - \pi_{(t+1)}\|_1 < \infty$.

The rest of proof is structured as follows. First, we show that the sequence of stable-state distributions for the MC used by Algorithm 1 converges to a uniform distribution over the set of sequences that minimize the energy function as the iteration count $t$ increases. Then, we show using Theorems 2 and 3 that the non-homogeneous MC used in Algorithm 1 is strongly ergodic, which by the definition of strong ergodicity implies that Algorithm 1 always converges to the stable distribution found above. This implies that the outcome of Algorithm 1 converges to a minimum-energy solution as $t \to \infty$, completing the proof of Theorem 1.

We therefore begin by finding the stable-state distribution for the non-homogeneous MC used by Algorithm 1. At each super-iteration $t$, the distribution defined as

$$\pi_{(t)}(w) \triangleq \frac{\exp(-s_t \Psi^{H_q}(w))}{\sum_{z \in \mathcal{S}} \exp(-s_t \Psi^{H_q}(z))} = \frac{1}{\sum_{z \in \mathcal{S}} \exp(-s_t(\Psi^{H_q}(z) - \Psi^{H_q}(w)))} \tag{24}$$

satisfies $\pi_{(t)} P_{(t)} = \pi_{(t)}$, cf. (13). We can show that the distribution $\pi_{(t)}$ converges to a uniform distribution over the set of sequences that minimize the energy function, i.e.,

$$\lim_{t \to \infty} \pi_{(t)}(w) = \begin{cases} 0 & w \notin \mathcal{H}, \\ \frac{1}{|\mathcal{H}|} & w \in \mathcal{H}, \end{cases} \tag{25}$$

where $\mathcal{H} = \{w \in \mathcal{S} \text{ s.t. } \Psi^{H_q}(w) = \min_{z \in \mathcal{S}} \Psi^{H_q}(z)\}$. To show (25), we will show that $\pi_{(t)}(w)$ is increasing for $w \in \mathcal{H}$ and eventually decreasing for $w \in \mathcal{H}^C$. Since for $w \in \mathcal{H}$ and $\widetilde{w} \in \mathcal{S}$ we have $\Psi^{H_q}(\widetilde{w}) - \Psi^{H_q}(w) \geq 0$, for $t_1 < t_2$ we have

$$\sum_{\widetilde{w} \in \mathcal{S}} \exp(-s_{t_1}(\Psi^{H_q}(\widetilde{w}) - \Psi^{H_q}(w))) \geq \sum_{\widetilde{w} \in \mathcal{S}} \exp(-s_{t_2}(\Psi^{H_q}(\widetilde{w}) - \Psi^{H_q}(w))),$$

which together with (24) implies $\pi_{(t_1)}(w) \leq \pi_{(t_2)}(w)$. On the other hand, if $w \in \mathcal{H}^C$, then we obtain

$$\pi_{(t)}(w) = \Bigg[ \sum_{\widetilde{w}:\Psi^{H_q}(\widetilde{w}) \geq \Psi^{H_q}(w)} \exp(-s_t(\Psi^{H_q}(\widetilde{w}) - \Psi^{H_q}(w))) + $$
$$\sum_{\widetilde{w}:\Psi^{H_q}(\widetilde{w}) < \Psi^{H_q}(w)} \exp(-s_t(\Psi^{H_q}(\widetilde{w}) - \Psi^{H_q}(w))) \Bigg]^{-1}. \tag{26}$$

For sufficiently large $s_t$, the denominator of (26) is dominated by the second term, which increases when $s_t$ increases, and therefore $\pi_{(t)}(w)$ decreases for $w \in \mathcal{H}^C$ as $t$ increases. Finally, since all sequences $w \in \mathcal{H}$ have the same energy $\Psi^{H_q}(w)$, it follows that the distribution is uniform over the symbols in $\mathcal{H}$.

Having shown convergence of the non-homogenous MC's stable-state distributions, we now show that the non-homogeneous MC is strongly ergodic. The transition matrix $P_{(t)}$ of the MC at iteration $t$ depends on the temperature

$s_t$ in (16) used within Algorithm 1. We first show that the MC used in Algorithm 1 is weakly ergodic via Theorem 2; the proof of the following Lemma is given in Appendix B.

*Lemma 1:* The ergodic coefficient of $P_{(t)}$ for any $t \geq 0$ is upper bounded by $\xi\left(P_{(t)}\right) \leq 1 - \exp(-s_t N \Delta_q)$, where $\Delta_q$ is defined in (15).

We note in passing that Condition 1 ensures that $\Delta_q$ is finite. Using Lemma 1 and (16), we can evaluate the sum given in Theorem 2 as

$$\sum_{j=1}^{\infty} \left(1 - \xi\left(P_{(j)}\right)\right) \geq \sum_{j=1}^{\infty} \exp(-s_j N \Delta_q) = \sum_{j=1}^{\infty} \frac{1}{j^{1/c}} = \infty,$$

and so the non-homogeneous MC defined by $\{P_{(t)}\}_{t=1}^{\infty}$ is weakly ergodic. Now we use Theorem 3 to show that the MC is strongly ergodic by proving that $\sum_{t=1}^{\infty} \|\pi_{(t)} - \pi_{(t+1)}\|_1 < \infty$. Since we know from earlier in the proof that $\pi_{(t)}(w)$ is increasing for $w \in \mathcal{H}$ and eventually decreasing for $w \in \mathcal{H}^C$, there exists a $t_0 \in \mathbb{N}$ such that for any $t_1 > t_0$, we have

$$\begin{aligned}
\sum_{t=t_0}^{t_1} \|\pi_{(t)} - \pi_{(t+1)}\|_1 &= \sum_{w \in \mathcal{H}} \sum_{t=t_0}^{t_1} \left(\pi_{(t+1)}(w) - \pi_{(t)}(w)\right) + \sum_{w \notin \mathcal{H}} \sum_{t=t_0}^{t_1} \left(\pi_{(t)}(w) - \pi_{(t+1)}(w)\right) \\
&= \sum_{w \in \mathcal{H}} \left(\pi_{(t_1+1)}(w) - \pi_{(t_0)}(w)\right) + \sum_{w \notin \mathcal{H}} \left(\pi_{(t_0)}(w) - \pi_{(t_1+1)}(w)\right) \\
&= \|\pi_{(t_1+1)} - \pi_{(t_0)}\|_1 \leq \|\pi_{(t_1+1)}\|_1 + \|\pi_{(t_0)}\|_1 = 2.
\end{aligned} \tag{27}$$

Since the right hand side does not depend on $t_1$, we have that $\sum_{t=1}^{\infty} \|\pi_{(t)} - \pi_{(t+1)}\|_1 < \infty$. This implies that the non-homogeneous MC used by Algorithm 1 is strongly ergodic, and thus completes the proof of Theorem 1.

## APPENDIX B. PROOF OF LEMMA 1

Let $w', w''$ be two arbitrary sequences in $\mathcal{S}$. The probability of transitioning from a given state to a neighboring state in an iteration within iteration $t'$ of super-iteration $t$ of Algorithm 1 is given by (13), and can be rewritten as

$$\begin{aligned}
P_{(t,t')}(w_1^{t'-1} a w_{t'+1}^N | w_1^{t'-1} b w_{t'+1}^N) &= p_{s_t}(w_{t'} = a | w^{\backslash t'}) = \frac{\exp\left(-s_t \Psi^{H_q}(w_1^{t'-1} a w_{t'+1}^N)\right)}{\sum_{b \in \mathcal{R}_F} \exp\left(-s_t \Psi^{H_q}(w_1^{t'-1} b w_{t'+1}^N)\right)} \\
&= \frac{\exp\left(-s_t \left(\Psi^{H_q}(w_1^{t'-1} a w_{t'+1}^N) - \Psi_{\min,t'}^{H_q}(w_1^{t'-1}, w_{t'+1}^N)\right)\right)}{\sum_{b \in \mathcal{R}_F} \exp\left(-s_t \left(\Psi^{H_q}(w_1^{t'-1} b w_{t'+1}^N) - \Psi_{\min,t'}^{H_q}(w_1^{t'-1}, w_{t'+1}^N)\right)\right)} \geq \frac{\exp(-s_t \Delta_q)}{|\mathcal{R}_F|},
\end{aligned} \tag{28}$$

where $\Psi_{\min,t'}^{H_q}(w_1^{t'-1}, w_{t'+1}^N) = \min_{\beta \in \mathcal{R}_F} \Psi^{H_q}(w_1^{t'-1} \beta w_{t'+1}^N)$. Therefore, the smallest probability of transition from $w'$ to $w''$ within super-iteration $t$ of Algorithm 1 is bounded by

$$\min_{w', w'' \in \mathcal{R}_F} P_{(t)}(w'' | w') \geq \prod_{t'=1}^{N} \frac{\exp(-s_t \Delta_q)}{|\mathcal{R}_F|} = \frac{\exp(-s_t N \Delta_q)}{|\mathcal{R}_F|^N} = \frac{\exp(-s_t N \Delta_q)}{|\mathcal{S}|}.$$

Using the alternative definition of the ergodic coefficient (23),

$$\xi\left(P_{(t)}\right) = 1 - \min_{w', w'' \in \mathcal{S}} \sum_{\widetilde{w} \in \mathcal{S}} \min(P_{(t)}(\widetilde{w} | w'), P_{(t)}(\widetilde{w} | w'')) \leq 1 - |\mathcal{S}| \frac{\exp(-s_t N \Delta_q)}{|\mathcal{S}|} = 1 - \exp(-s_t N \Delta_q),$$

proving the lemma.

## References

[1] J. Zhu, D. Baron, and M. F. Duarte, "Complexity–adaptive universal signal estimation for compressed sensing," in *Proc. IEEE Stat. Sig. Proc. Workshop (SSP)*, June 2014, pp. 416–419.

[2] D. Baron and M. F. Duarte, "Universal MAP estimation in compressed sensing," in *Proc. Allerton Conf. Comm., Control, Computing*, Sept. 2011, pp. 768–775.

[3] D. Baron, "Information complexity and estimation," in *Workshop Inf. Theoretic Methods Science Eng. (WITMSE)*, Aug. 2011.

[4] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[5] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[6] Y. Wu and S. Verdú, "Optimal phase transitions in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 10, pp. 6241 – 6263, Oct. 2012.

[7] M. Figueiredo, R. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Top. Sign. Proces.*, vol. 1, pp. 586–597, Dec. 2007.

[8] D. L. Donoho, A. Maleki, and A. Montanari, "Message Passing Algorithms for Compressed Sensing: I. Motivation and Construction," in *IEEE Inf. Theory Workshop*, Jan. 2010.

[9] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2011, pp. 2168–2172.

[10] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[11] M. W. Seeger and H. Nickisch, "Compressed sensing and Bayesian experimental design," in *Proc. Int. Conf. Machine Learning*, Aug 2008, pp. 912–919.

[12] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, pp. 269–280, Jan. 2010.

[13] J. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[14] J. Ziniel, S. Rangan, and P. Schniter, "A generalized framework for learning and recovery of structured sparse signals," in *Proc. IEEE Stat. Sig. Proc. Workshop (SSP)*, Aug. 2012, pp. 325–328.

[15] Y. Ma, J. Tan, N. Krishnan, and D. Baron, "Empirical Bayes and full Bayes for signal estimation," *arXiv:1405.2113v1*, May 2014.

[16] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.

[17] D. Donoho, H. Kakavand, and J. Mammen, "The simplest solution to an underdetermined system of linear equations," in *Proc. Int. Symp. Inf. Theory (ISIT)*, July 2006, pp. 1924–1928.

[18] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4036–4048, Sept. 2006.

[19] J. D. Haupt and R. Nowak, "Adaptive sensing for sparse recovery," in *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[20] I. Ramírez and G. Sapiro, "An MDL framework for sparse coding and dictionary learning," *IEEE Trans. Signal Proc.*, vol. 60, no. 6, pp. 2913–2927, June 2012.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York, NY, USA: Wiley-Interscience, 2006.

[22] M. Li and P. M. B. Vitanyi, *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, New York, 2008.

[23] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Workshop Neural Info. Proc. Sys. (NIPS)*, Vancouver, Canada, Dec. 2008.

[25] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[26] P. J. Garrigues and B. A. Olshausen, "Learning horizontal connections in a sparse coding model of natural images," in *Workshop Neural Info. Proc. Sys. (NIPS)*, Dec. 2007, pp. 1–8.

[27] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[28] J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory*, vol. 29, no. 5, pp. 656–664, Sept. 1983.

[29] D. L. Donoho, "The Kolmogorov sampler," Department of Statistics Technical Report 2002-4, Stanford University, Stanford, CA, Jan. 2002.

[30] G. J. Chaitin, "On the length of programs for computing finite binary sequences," *J. ACM*, vol. 13, no. 4, pp. 547–569, 1966.

[31] R. J. Solomonoff, "A formal theory of inductive inference. Part I," *Inf. and Control*, vol. 7, no. 1, pp. 1–22, Mar. 1964.

[32] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems Inf. Transmission*, vol. 1, no. 1, pp. 1–7, 1965.

[33] S. Jalali and A. Maleki, "Minimum complexity pursuit," in *Proc. Allerton Conf. Commun., Control, Computing*, Sept. 2011, pp. 1764–1770.

[34] S. Jalali, A. Maleki, and R. G. Baraniuk, "Minimum complexity pursuit for universal compressed sensing," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2253–2268, Apr. 2014.

[35] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, Sept. 1978.

[36] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[37] C. S. Wallace and D. M. Boulton, "An information measure for classification," *Comput J*, vol. 11, no. 2, pp. 185–194, 1968.

[38] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2743–2760, Oct. 1998.

[39] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[40] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harm. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[41] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 1, pp. 59–73, July 2011.

[42] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," *ArXiv preprint arXiv:1001.2228*, Jan. 2010.

[43] D. Guo and C. C. Wang, "Multiuser detection of sparsely spread CDMA," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, pp. 421–431, Apr. 2008.

[44] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, Oct. 1950.

[45] D. Baron and T. Weissman, "An MCMC approach to universal lossy compression of analog sources," *IEEE Trans. Sig. Process.*, vol. 60, pp. 5230–5240, Oct. 2012.

[46] S. Jalali and T. Weissman, "Rate-distortion via Markov chain Monte Carlo," in *Proc. Int. Symp. Inf. Theory (ISIT)*, July 2008, pp. 852–856.

[47] S. Jalali and T. Weissman, "Block and sliding-block lossy compression via MCMC," *IEEE Trans. Comm.*, vol. 60, no. 8, pp. 2187–2198, Aug. 2012.

[48] E. Yang, Z. Zhang, and T. Berger, "Fixed-slope universal lossy data compression," *IEEE Trans. Inf. Theory*, vol. 43, no. 5, pp. 1465–1476, Sept. 1997.

[49] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[50] J. Zhu and D. Baron, "Performance regions in compressed sensing from noisy measurements," in *Proc. 2013 Conf. Inf. Sciences Systems*, Baltimore, MD, Mar. 2013, pp. 1–6.

[51] S. Jalali and H. V. Poor, "Universal compressed sensing of Markov sources," *Arxiv preprint arXiv:1406.7807*, June 2014.

[52] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, vol. 31, Springer Verlag, 1999.