Dynamic Compressive Sensing of Time-Varying Signals via Approximate Message Passing

Justin Ziniel and Philip Schniter

Abstract

In this work the dynamic compressive sensing (CS) problem of recovering sparse, correlated, time-varying signals from sub-Nyquist, non-adaptive, linear measurements is explored from a Bayesian perspective. While there has been a handful of previously proposed Bayesian dynamic CS algorithms in the literature, the ability to perform inference on high-dimensional problems in a computationally efficient manner remains elusive. In response, we propose a probabilistic dynamic CS signal model that captures both amplitude and support correlation structure, and describe an approximate message passing algorithm that performs soft signal estimation and support detection with a computational complexity that is linear in all problem dimensions. The algorithm, DCS-AMP, can perform either causal filtering or non-causal smoothing, and is capable of learning model parameters adaptively from the data through an expectation-maximization learning procedure. We provide numerical evidence that DCS-AMP performs within 3 dB of oracle bounds on synthetic data under a variety of operating conditions. We further describe the result of applying DCS-AMP to two real dynamic CS datasets, as well as a frequency estimation task, to bolster our claim that DCS-AMP is capable of offering state-of-the-art performance and speed on real-world high-dimensional problems.

I. Introduction

In this work, we consider the *dynamic compressive sensing* (dynamic CS) problem, in which a sparse, vector-valued time series is recovered from a second time series of noisy, sub-Nyquist, linear

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, Ohio. E-mail: {zinielj, schniter}@ece.osu.edu.

Work supported in part by NSF grant CCF-1018368, DARPA/ONR grant N66001-10-1-4090, and an allocation of computing time from the Ohio Supercomputer Center.

Portions of this work were previously presented at the 2010 Asilomar Conference on Signals, Systems, and Computing [1].

measurements. Such a problem finds application in dynamic MRI [2], high-speed video capture [3], and underwater channel estimation [4] amongst others.

Framed mathematically, the objective of the dynamic CS problem is to recover the time series $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}\}$, where $\boldsymbol{x}^{(t)} \in \mathbb{C}^N$ is the signal at timestep t, from a time series of measurements, $\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}\}$. Each $\boldsymbol{y}^{(t)} \in \mathbb{C}^M$ is obtained from the linear measurement process,

$$\boldsymbol{y}^{(t)} = \boldsymbol{A}^{(t)}\boldsymbol{x}^{(t)} + \boldsymbol{e}^{(t)}, \qquad t = 1, \ldots, T, \tag{1}$$

with $\boldsymbol{e}^{(t)}$ representing corrupting noise. The measurement matrix $\boldsymbol{A}^{(t)}$ (which may be time-varying or time-invariant, i.e., $\boldsymbol{A}^{(t)} = \boldsymbol{A} \,\forall\, t$) is known in advance, and is generally wide, leading to an underdetermined system of equations. The problem is regularized by assuming that $\boldsymbol{x}^{(t)}$ is sparse (or compressible),[1] having relatively few non-zero (or large) entries.

In many real-world scenarios, the underlying time-varying sparse signal exhibits substantial temporal correlation. This temporal correlation may manifest itself in two interrelated ways: *(i)* the support of the signal may change slowly over time, and *(ii)* the amplitudes of the large coefficients may vary smoothly in time [2], [3], [5].

In such scenarios, incorporating an appropriate model of temporal structure into a recovery technique makes it possible to drastically outperform structure-agnostic CS algorithms. From an analytical standpoint, Vaswani and Lu demonstrate that the restricted isometry property (RIP) sufficient conditions for perfect recovery in the dynamic CS problem are significantly weaker than those found in the traditional single measurement vector (SMV) CS problem when accounting for the additional structure [6]. In this work, we take a Bayesian approach to modeling this structure, which contrasts those dynamic CS algorithms inspired by convex relaxation, such as the Dynamic LASSO [5] and the Modified-CS algorithm [6]. Our Bayesian framework is also distinct from those hybrid techniques that blend elements of Bayesian dynamical models like the Kalman filter with more traditional CS approaches of exploiting sparsity through convex relaxation [2], [7] or greedy methods [8].

In particular, we propose a probabilistic model that treats the time-varying signal support as a set of independent binary Markov processes and the time-varying coefficient amplitudes as a set of independent Gauss-Markov processes. As detailed in Section II, this model leads to coefficient marginal distributions that are Bernoulli-Gaussian (i.e., "spike-and-slab"). Later, in Section V, we describe a generalization of the aforementioned model that yields Bernoulli-Gaussian-mixture coefficient marginals with an arbitrary

[1]Without loss of generality, we assume $\boldsymbol{x}^{(t)}$ is sparse/compressible in the canonical basis. Other sparsifying bases can be incorporated into the measurement matrix $\boldsymbol{A}^{(t)}$ without changing our model.

number of mixture components. The models that we propose thus differ substantially from those used in other Bayesian approaches to dynamic CS, [9] and [10]. In particular, Sejdinović et al. [9] combine a linear Gaussian dynamical system model with a sparsity-promoting Gaussian-scale-mixture prior, while Shahrasbi et al. [10] employ a particular spike-and-slab Markov model that couples amplitude evolution together with support evolution.

Our inference method also differs from those used in the alternative Bayesian dynamic CS algorithms [9] and [10]. In [9], Sejdinović et al. perform inference via a sequential Monte Carlo sampler [11]. Sequential Monte Carlo techniques are appealing for their applicability to complicated non-linear, non-Gaussian inference tasks like the Bayesian dynamic CS problem. Nevertheless, there are a number of important practical issues related to selection of the importance distribution, choice of the resampling method, and the number of sample points to track, since in principle one must increase the number of points exponentially over time to combat degeneracy [11]. Additionally, Monte Carlo techniques can be computationally expensive in high-dimensional inference problems. An alternative inference procedure that has recently proven successful in a number of applications is loopy belief propagation (LBP) [12]. In [10], Shahrasbi et al. extend the conventional LBP method proposed in [13] for standard CS under a sparse measurement matrix $\boldsymbol{A}$ to the case of dynamic CS under sparse $\boldsymbol{A}^{(t)}$. Nevertheless, the confinement to sparse measurement matrices is very restrictive, and, without this restriction, the methods of [10], [13] become computationally intractable.

Our inference procedure is based on the recently proposed framework of approximate message passing (AMP) [14], and in particular its "turbo" extension [15]. AMP, an unconventional form of LBP, was originally proposed for standard CS with a dense measurement matrix [14], and its noteworthy properties include: *(i)* a rigorous analysis (as $M, N \to \infty$ with M/N fixed, under i.i.d. Gaussian $\boldsymbol{A}$) establishing that its solutions are governed by a state-evolution whose fixed points are optimal in several respects [16], and *(ii)* extremely fast runtimes (as a consequence of the fact that it needs relatively few iterations, each requiring only one multiplication by $\boldsymbol{A}$ and its transpose). The turbo-AMP framework originally proposed in [15] offers a way to extend AMP to structured-sparsity problems such as compressive imaging [17], joint communication channel/symbol estimation [18], and—as we shall see in this work—the dynamic CS problem.

A. Notation

Boldfaced lower-case letters, e.g., $\boldsymbol{a}$, denote column vectors, while boldfaced upper-case letters, e.g., $\boldsymbol{A}$, denote matrices. The letter t is strictly used to index a timestep, $t = 1, 2, \ldots, T$, the letter n is

strictly used to index the coefficients of a signal, $n = 1, \ldots, N$, and the letter m is strictly used to index the measurements, $m = 1, \ldots, M$. The superscript $^{(t)}$ indicates a timestep-dependent quantity, while a superscript without parentheses, such as ^{k}, indicates a quantity whose value changes according to some algorithmic iteration index k. Subscript notations such as $x_n^{(t)}$ are used to denote the n^{th} element of the vector $\boldsymbol{x}^{(t)}$, while set subscript notation, e.g., $\boldsymbol{x}_{\mathcal{S}}^{(t)}$, denotes the sub-vector of $\boldsymbol{x}^{(t)}$ consisting of indices contained in $\mathcal{S}$. The m^{th} row of the matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{a}_m^{\mathsf{T}}$, an M-by-M identity matrix is denoted by $\boldsymbol{I}_M$, and a length-N vector of ones is given by $\mathbf{1}_N$. Finally, $\mathcal{CN}(\boldsymbol{a}; \boldsymbol{b}, \boldsymbol{C})$ refers to the circularly symmetric complex normal distribution that is a function of the vector $\boldsymbol{a}$, with mean $\boldsymbol{b}$ and covariance matrix $\boldsymbol{C}$.

II. Signal Model

We assume that the measurement process can be accurately described by the linear model of (1). We further assume that $\boldsymbol{A}^{(t)} \in \mathbb{C}^{M \times N}$, $t = 1, \ldots, T$, are measurement matrices known in advance, whose columns have been scaled to be of unit norm.[2] We model the noise as a stationary, circularly symmetric, additive white Gaussian noise (AWGN) process, with $\boldsymbol{e}^{(t)} \sim \mathcal{CN}(\mathbf{0}, \sigma_e^2 \boldsymbol{I}_M) \; \forall \; t$.

As noted in Section I, the sparse time series, $\{\boldsymbol{x}^{(t)}\}_{t=1}^{T}$, often exhibits a high degree of correlation from one timestep to the next. In this work, we model this correlation through a slow time-variation of the signal support, and a smooth evolution of the amplitudes of the non-zero coefficients. To do so, we introduce two hidden random processes, $\{\boldsymbol{s}^{(t)}\}_{t=1}^{T}$ and $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^{T}$. The binary vector $\boldsymbol{s}^{(t)} \in \{0,1\}^N$ describes the support of $\boldsymbol{x}^{(t)}$, denoted $\mathcal{S}^{(t)}$, while the vector $\boldsymbol{\theta}^{(t)} \in \mathbb{C}^N$ describes the amplitudes of the active elements of $\boldsymbol{x}^{(t)}$. Together, $\boldsymbol{s}^{(t)}$ and $\boldsymbol{\theta}^{(t)}$ completely characterize $\boldsymbol{x}^{(t)}$ as follows:

$$x_n^{(t)} = s_n^{(t)} \cdot \theta_n^{(t)} \quad \forall n, t. \tag{2}$$

Therefore, $s_n^{(t)} = 0$ sets $x_n^{(t)} = 0$ and $n \notin \mathcal{S}^{(t)}$, while $s_n^{(t)} = 1$ sets $x_n^{(t)} = \theta_n^{(t)}$ and $n \in \mathcal{S}^{(t)}$.

To model slow changes in the support $\mathcal{S}^{(t)}$ over time, we model the n^{th} coefficient's support across time, $\{s_n^{(t)}\}_{t=1}^{T}$, as a Markov chain defined by two transition probabilities: $p_{10} \triangleq \Pr\{s_n^{(t)} = 1 | s_n^{(t-1)} = 0\}$, and $p_{01} \triangleq \Pr\{s_n^{(t)} = 0 | s_n^{(t-1)} = 1\}$, and employ independent chains across $n = 1, \ldots, N$. We further assume that each Markov chain operates in steady-state, such that $\Pr\{s_n^{(t)} = 1\} = \lambda \; \forall \, n, t$. This steady-state assumption implies that these Markov chains are completely specified by the parameters λ and p_{01}, which together determine the remaining transition probability $p_{10} = \lambda p_{01}/(1-\lambda)$. Depending on how

[2]Our algorithm can be generalized to support $\boldsymbol{A}^{(t)}$ without equal-norm columns, a number of measurements M_t that change as a function of time t, and real-valued matrices/signals as well.

p_{01} is chosen, the prior distribution can favor signals that exhibit a nearly static support across time, or it can allow for signal supports that change substantially from timestep to timestep. For example, it can be shown that $1/p_{01}$ specifies the average run length of a sequence of ones in the Markov chains.

The second form of temporal structure that we capture in our signal model is the correlation in active coefficient amplitudes across time. We model this correlation through independent stationary steady-state Gauss-Markov processes for each n, wherein $\{\theta_n^{(t)}\}_{t=1}^T$ evolves in time according to

$$\theta_n^{(t)} = (1-\alpha)\big(\theta_n^{(t-1)} - \zeta\ \big) + \alpha w_n^{(t)} + \zeta, \tag{3}$$

where $\zeta \in \mathbb{C}$ is the mean of the process, $w_n^{(t)} \sim \mathcal{CN}(0,\rho)$ is an i.i.d. circular white Gaussian perturbation, and $\alpha \in [0,1]$ controls the temporal correlation. At one extreme, $\alpha = 0$, the amplitudes are totally correlated, (i.e., $\theta_n^{(t)} = \theta_n^{(t-1)}$), while at the other extreme, $\alpha = 1$, the amplitudes evolve according to an uncorrelated Gaussian random process with mean ζ.

At this point, we would like to make a few remarks about our signal model. First, due to (2), it is clear that $p\big(x_n^{(t)}|s_n^{(t)},\theta_n^{(t)}\big) = \delta\big(x_n^{(t)} - s_n^{(t)}\theta_n^{(t)}\big)$, where $\delta(\cdot)$ is the Dirac delta function. By margining out the distributions of $s_n^{(t)}$ and $\theta_n^{(t)}$, one finds that

$$p(x_n^{(t)}) = (1-\lambda)\delta(x_n^{(t)}) + \lambda\,\mathcal{CN}(x_n^{(t)};\zeta,\sigma^2), \tag{4}$$

where $\sigma^2 \triangleq \frac{\alpha\rho}{2-\alpha}$ is the steady-state variance of $\theta_n^{(t)}$. Equation (4) is a Bernoulli-Gaussian or "spike-and-slab" distribution, which is an effective sparsity-promoting prior due to the point-mass at $x_n^{(t)} = 0$. Second, we observe that the amplitude random process, $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^T$, evolves independently from the sparsity pattern random process, $\{\boldsymbol{s}^{(t)}\}_{t=1}^T$. As a result of this modeling choice, there can be significant hidden amplitudes $\theta_n^{(t)}$ associated with inactive coefficients (those for which $s_n^{(t)} = 0$). Consequently, $\theta_n^{(t)}$ should be viewed as the amplitude of $x_n^{(t)}$ *conditioned* on $s_n^{(t)} = 1$. Lastly, we note that higher-order Markov processes and/or more complex coefficient marginals could be considered within the framework we propose, however, to keep development simple, we restrict our attention to first-order Markov processes and Bernoulli-Gaussian marginals until Section V, where we describe an extension of the above signal model that yields Bernoulli-Gaussian-mixture marginals.

III. The DCS-AMP Algorithm

In this section we will describe the DCS-AMP algorithm, which efficiently and accurately estimates the marginal posterior distributions of $\{x_n^{(t)}\}$, $\{\theta_n^{(t)}\}$, and $\{s_n^{(t)}\}$ from the observed measurements $\{\boldsymbol{y}^{(t)}\}_{t=1}^T$, thus enabling both soft estimation and soft support detection. As mentioned in Section I, DCS-AMP can perform either filtering or smoothing.

Factor	Distribution	Functional Form
$g_m^{(t)}\big(\boldsymbol{x}^{(t)}\big)$	$p\big(y_m^{(t)}\big\vert\boldsymbol{x}^{(t)}\big)$	$\mathcal{CN}\big(y_m^{(t)};\boldsymbol{a}_m^{(t)\,T}\boldsymbol{x}^{(t)},\sigma_e^2\big)$
$f_n^{(t)}\big(x_n^{(t)},s_n^{(t)},\theta_n^{(t)}\big)$	$p\big(x_n^{(t)}\big\vert s_n^{(t)},\theta_n^{(t)}\big)$	$\delta\big(x_n^{(t)}-s_n^{(t)}\theta_n^{(t)}\big)$
$h_n^{(1)}\big(s_n^{(1)}\big)$	$p\big(s_n^{(1)}\big)$	$\big(1-\lambda\big)^{1-s_n^{(1)}}\lambda^{s_n^{(1)}}$
$h_n^{(t)}\big(s_n^{(t)},s_n^{(t-1)}\big)$	$p\big(s_n^{(t)}\big\vert s_n^{(t-1)}\big)$	$\begin{cases}(1-p_{10})^{1-s_n^{(t)}}{p_{10}}^{s_n^{(t)}}, & s_n^{(t-1)}=0\\ {p_{01}}^{1-s_n^{(t)}}(1-p_{01})^{s_n^{(t)}}, & s_n^{(t-1)}=1\end{cases}$
$d_n^{(1)}\big(\theta_n^{(1)}\big)$	$p\big(\theta_n^{(1)}\big)$	$\mathcal{CN}\big(\theta_n^{(1)};\zeta,\sigma^2\big)$
$d_n^{(t)}\big(\theta_n^{(t)},\theta_n^{(t-1)}\big)$	$p\big(\theta_n^{(t)}\big\vert\theta_n^{(t-1)}\big)$	$\mathcal{CN}\big(\theta_n^{(t)};(1-\alpha)\theta_n^{(t-1)}+\alpha\zeta,\alpha^2\rho\big)$

TABLE I: The factors, underlying distributions, and functional forms associated with our signal model

The algorithm we develop is designed to exploit the statistical structure inherent in our signal model. By defining $\bar{\boldsymbol{y}}$ to be the collection of all measurements, $\{\boldsymbol{y}^{(t)}\}_{t=1}^T$ (and defining $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{s}}$, and $\bar{\boldsymbol{\theta}}$ similarly), the posterior joint distribution of the signal, support, and amplitude time series, given the measurement time series, can be expressed using Bayes' rule as

$$p(\bar{\boldsymbol{x}},\bar{\boldsymbol{s}},\bar{\boldsymbol{\theta}}|\bar{\boldsymbol{y}})\propto\prod_{t=1}^{T}\left(\prod_{m=1}^{M}p(y_m^{(t)}|\boldsymbol{x}^{(t)})\prod_{n=1}^{N}p(x_n^{(t)}|s_n^{(t)},\theta_n^{(t)})p(s_n^{(t)}|s_n^{(t-1)})p(\theta_n^{(t)}|\theta_n^{(t-1)})\right), \tag{5}$$

where $\propto$ indicates proportionality up to a constant scale factor, $p(s_n^{(1)}|s_n^{(0)})\triangleq p(s_n^{(1)})$, and $p(\theta_n^{(1)}|\theta_n^{(0)})\triangleq p(\theta_n^{(1)})$. By inspecting (5), we see that the posterior joint distribution decomposes into the product of many distributions that only depend on small subsets of variables. A graphical representation of such decompositions is given by the *factor graph*, which is an undirected bipartite graph that connects the pdf "factors" of (5) with the random variables that constitute their arguments [20]. In Table I, we introduce the notation that we will use for the factors of our signal model, showing the correspondence between the factor labels and the underlying distributions they represent, as well as the specific functional form assumed by each factor. The associated factor graph for the posterior joint distribution of (5) is shown in Fig. 1, labeled according to Table I. Filled squares represent factors, while circles represent random variables.

As seen in Fig. 1, all of the variables needed at a given timestep can be visualized as lying in a plane, with successive planes stacked one after another in time. We will refer to these planes as "frames". The temporal correlation of the signal supports is illustrated by the $h_n^{(t)}$ factor nodes that connect the $s_n^{(t)}$ variable nodes between neighboring frames. Likewise, the temporal correlation of the signal amplitudes is expressed by the interconnection of $d_n^{(t)}$ factor nodes and $\theta_n^{(t)}$ variable nodes. For visual clarity, these factor nodes have been omitted from the middle portion of the factor graph, appearing only at indices $n=1$ and $n=N$, but should in fact be present for all indices $n=1,\ldots,N$. Since the measurements

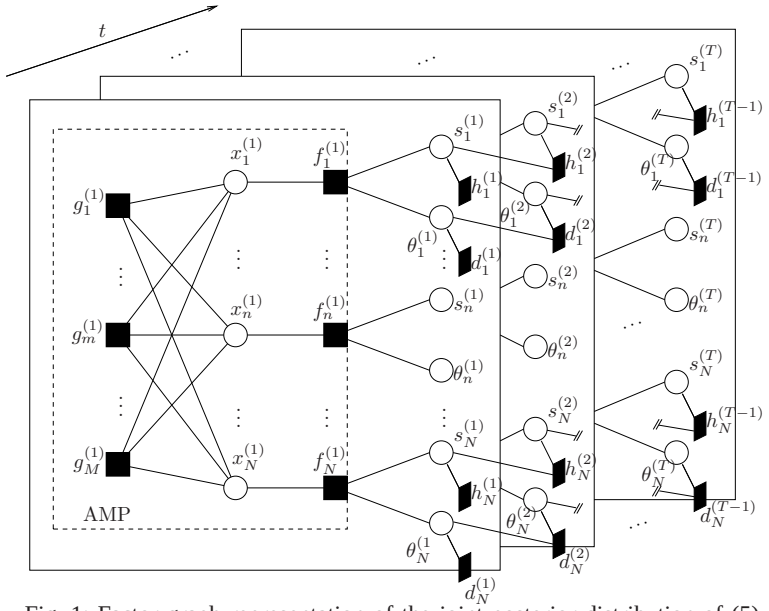

Fig. 1: Factor graph representation of the joint posterior distribution of (5).

$\{y_m^{(t)}\}$ are observed variables, they have been incorporated into the $g_m^{(t)}$ factor nodes.

The algorithm that we develop can be viewed as an approximate implementation of belief propagation (BP) [21], a message passing algorithm for performing inference on factor graphs that describe probabilistic models. When the factor graph is cycle-free, belief propagation is equivalent to the more general sum-product algorithm [20], which is a means of computing the marginal functions that result from summing (or integrating) a multivariate function over all possible input arguments, with one argument held fixed, (i.e., margining out all but one variable). In the context of BP, these marginal functions are the marginal distributions of random variables. Thus, given measurements $\bar{\boldsymbol{y}}$ and the factorization of the posterior joint distribution $p\big(\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}}, \bar{\boldsymbol{\theta}}\big|\bar{\boldsymbol{y}}\big)$, DCS-AMP computes (approximate) posterior marginals of $x_n^{(t)}$, $s_n^{(t)}$, and $\theta_n^{(t)}$. In "filtering" mode, our algorithm would therefore return, e.g., $p\big(x_n^{(t)}\big|\{\boldsymbol{y}^{(t)}\}_{t=1}^{t}\big)$, while in "smoothing" mode it would return $p\big(x_n^{(t)}\big|\{\boldsymbol{y}^{(t)}\}_{t=1}^{T}\big)$. From these marginals, one can compute, e.g., minimum mean-squared error (MMSE) estimates. The factor graph of Fig. 1 contains many short cycles, however, and thus the convergence of loopy BP cannot be guaranteed [20].[3] Despite this, loopy BP has been shown to perform extremely well in a number of different applications, including turbo decoding [26], computer vision [27], and compressive sensing [13]–[15], [17], [28]–[30].

[3]However, it is worth noting that in the past decade much work has been accomplished in identifying specific situations under which loopy BP *is* guaranteed to converge, e.g., [16], [22]–[25].

A. Message scheduling

In loopy factor graphs, there are a number of ways to schedule, or sequence, the messages that are exchanged between nodes. The choice of a schedule can impact not only the rate of convergence of the algorithm, but also the likelihood of convergence as well [31]. We propose a schedule (an evolution of the "turbo" schedule proposed in [15]) for DCS-AMP that is straightforward to implement, suitable for both filtering and smoothing applications, and empirically yields quickly converging estimates under a variety of diverse operating conditions.

Our proposed schedule can be broken down into four distinct steps, which we will refer to using the mnemonics **(into)**, **(within)**, **(out)**, and **(across)**. At a particular timestep t, the **(into)** step involves passing messages that provide current beliefs about the state of the relevant support variables, $\{s_n^{(t)}\}_{n=1}^N$, and amplitude variables, $\{\theta_n^{(t)}\}_{n=1}^N$, laterally *into* the dashed AMP box within frame t. (Recall Fig. 1.) The **(within)** step makes use of these incoming messages, together with the observations available in that frame, $\{y_m^{(t)}\}_{m=1}^M$, to exchange messages *within* the dashed AMP box of frame t, thus generating estimates of the marginal posteriors of the signal variables $\{x_n^{(t)}\}_{n=1}^N$. Using these posterior estimates, the **(out)** step propagates messages *out* of the dashed AMP box, providing updated beliefs about the state of $\{s_n^{(t)}\}_{n=1}^N$ and $\{\ theta a_n^{(t)}\}_{n=1}^N$. Lastly, the **(across)** step involves transmitting messages *across* neighboring frames, using the updated beliefs about $\{s_n^{(t)}\}_{n=1}^N$ and $\{\theta_n^{(t)}\}_{n=1}^N$ to influence the beliefs about $\{s_n^{(t+1)}\}_{n=1}^N$ and $\{\theta_n^{(t+1)}\}_{n=1}^N$ $\big($or $\{s_n^{(t-1)}\}_{n=1}^N$ and $\{\theta_n^{(t-1)}\}_{n=1}^N\big)$.

The procedures for filtering and smoothing both start in the same way. At the initial $t = 1$ frame, steps **(into)**, **(within)** and **(out)** are performed in succession. Next, step **(across)** is performed to pass messages from $\{s_n^{(1)}\}_{n=1}^N$ and $\{\theta_n^{(1)}\}_{n=1}^N$ to $\{s_n^{(2)}\}_{n=1}^N$ and $\{\theta_n^{(2)}\}_{n=1}^N$. Then at frame $t = 2$ the same set of steps are executed, concluding with messages propagating to $\{s_n^{(3)}\}_{n=1}^N$ and $\{\theta_n^{(3)}\}_{n=1}^N$. This process continues until steps **(into)**, **(within)** and **(out)** have been completed at the terminal frame, T. At this point, DCS-AMP has completed what we call a single forward pass. If the objective was to perform filtering, DCS-AMP terminates at this point, since only causal measurements have been used to estimate the marginal posteriors. If instead the objective is to obtain smoothed, non-causal estimates, then information begins to propagate backwards in time, i.e., step **(across)** moves messages from $\{s_n^{(T)}\}_{n=1}^N$ and $\{\theta_n^{(T)}\}_{n=1}^N$ to $\{s_n^{(T-1)}\}_{n=1}^N$ and $\{\theta_n^{(T-1)}\}_{n=1}^N$. Steps **(into)**, **(within)**, **(out)**, and **(across)** are performed at frame $T-1$, with messages bound for frame $T-2$. This continues until the initial frame is reached. At this point DCS-AMP has completed what we term as a single forward/backward pass. Multiple such passes, indexed by the variable k, can be carried out until a convergence criterion is met or a maximum number of passes

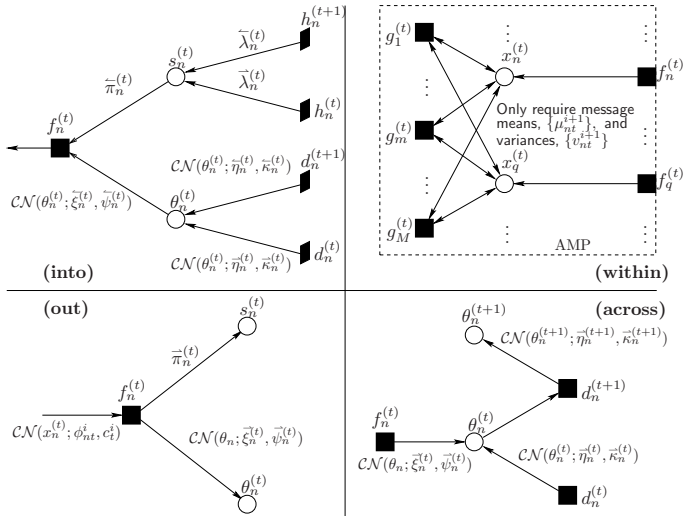

Fig. 2: A summary of the four message passing phases, including message notation and form.

has been performed.

B. Implementing the message passes

We now provide some additional details as to how the above four steps are implemented. To aid our discussion, in Fig. 2 we summarize the form of the messages that pass between the various factor graph nodes, focusing primarily on a single coefficient index n at an intermediate frame t. Directed edges indicate the direction that messages are moving. In the **(across)** phase, we only illustrate the messages involved in a forward pass for the amplitude variables, and leave out a graphic for the corresponding backward pass, as well as graphics for the support variable **(across)** phase. Note that, to be applicable at frame T, the factor node $d_n^{(t+1)}$ and its associated edge should be removed. The figure also introduces the notation that we adopt for the different variables that serve to parameterize the messages. For Bernoulli message pdfs, we show only the non-zero probability, e.g., $\vec{\lambda}_n^{(t)} = \nu_{h_n^{(t)} \to s_n^{(t)}}(s_n^{(t)} = 1)$.

To perform step **(into)**, the messages from the factors $h_n^{(t)}$ and $h_n^{(t+1)}$ to $s_n^{(t)}$ are used to set $\overleftarrow{\pi}_n^{(t)}$, the message from $s_n^{(t)}$ to $f_n^{(t)}$. Likewise, the messages from the factors $d_n^{(t)}$ and $d_n^{(t+1)}$ to $\theta_n^{(t)}$ are used to determine the message from $\theta_n^{(t)}$ to $f_n^{(t)}$. When performing filtering, or the first forward pass of smoothing, no meaningful information should be conveyed from the $h_n^{(t+1)}$ and $d_n^{(t+1)}$ factors. This can be accomplished by initializing $\left(\overleftarrow{\lambda}_n^{(t)}, \overleftarrow{\eta}_n^{(t)}, \overleftarrow{\kappa}_n^{(t)}\right)$ with the values $(\frac{1}{2}, 0, \infty)$.

In step **(within)**, messages must be exchanged between the $\left\{x_n^{(t)}\right\}_{n=1}^N$ and $\left\{g_m^{(t)}\right\}_{m=1}^M$ nodes. When $\boldsymbol{A}^{(t)}$ is not a sparse matrix, this will imply a dense network of connections between these nodes. Consequently, the standard sum-product algorithm would require us to evaluate multi-dimensional integrals of non-

Gaussian messages that grow exponentially in number in both N and M. This approach is clearly infeasible for problems of any appreciable size, and thus we turn to a simplification known as *approximate message passing* (AMP) [14], [28].

At a high-level, AMP can be viewed as a simplification of loopy BP, employing central limit theorem arguments to approximate the sum of many non-Gaussian random variables as a Gaussian. Through a series of principled approximation steps (which become exact for Gaussian $\boldsymbol{A}$ matrices in the large-system limit [16]), AMP produces an iterative thresholding algorithm that requires only $\mathcal{O}(MN)$ operations, dominated by matrix-vector products, to obtain posteriors on the $\left\{x_n^{(t)}\right\}_{n=1}^N$ variable nodes. The specifics of the iterative thresholding algorithm will depend on the signal prior under which AMP is operating [28], but it is assumed that the joint prior decouples into independent (but not necessarily i.i.d.) priors on each coefficient $x_n^{(t)}$.

By viewing $\nu_{f_n^{(t)}\to x_n^{(t)}}(\cdot)$ as a "local prior"[4] for $x_n^{(t)}$, we can readily apply the AMP technique as a means of performing the message passes within the portions of the factor graph enclosed within the dashed boxes of Fig. 1 (only one such box is visible). This local prior is a Bernoulli-Gaussian, namely

$$\nu_{f_n^{(t)}\to x_n^{(t)}}(x_n^{(t)}) = (1-\overleftarrow{\pi}_n^{(t)})\delta(x_n^{(t)}) + \overleftarrow{\pi}_n^{(t)}\mathcal{CN}(x_n^{(t)};\overleftarrow{\xi}_n^{(t)},\overleftarrow{\psi}_n^{(t)}).$$

The appropriate AMP message update equations for this local prior follow a straightforward extension of the derivations outlined in [15], which considered the special case of a zero-mean Bernoulli-Gaussian prior. The specific AMP updates for our model are given by (A4)-(A8) in Table II.

After employing AMP to manage the message passing between the $\left\{x_n^{(t)}\right\}_{n=1}^N$ and $\left\{g_m^{(t)}\right\}_{m=1}^M$ nodes in step **(within)**, messages must be propagated out of the dashed AMP box of frame t (step **(out)**) and either forward or backward in time (step **(across)**). While step **(across)** simply requires a straightforward application of the sum-product message computation rules, step **(out)** imposes several difficulties which we must address. For the remainder of this discussion, we focus on the message $\nu_{f_n^{(t)}\to\theta_n^{(t)}}(\cdot)$.

A routine application of the sum-product rules to the $f_n^{(t)}$-to-$\theta_n^{(t)}$ message would produce the following expression:

$$\nu^{\text{exact}}_{f_n^{(t)}\to\theta_n^{(t)}}(\theta_n^{(t)}) \triangleq (1-\overleftarrow{\pi}_n^{(t)})\mathcal{CN}(0;\phi_{nt}^i,c_t^i) + \overleftarrow{\pi}_n^{(t)}\mathcal{CN}(\theta_n^{(t)};\phi_{nt}^i,c_t^i). \tag{6}$$

Unfortunately, the term $\mathcal{CN}(0;\phi_{nt}^i,c_t^i)$ prevents us from normalizing $\nu^{\text{exact}}_{f_n^{(t)}\to\theta_n^{(t)}}(\theta_n^{(t)})$, as the former is constant with respect to $\theta_n^{(t)}$. Therefore, the distribution on $\theta_n^{(t)}$ represented by (6) is improper. To provide

[4]The AMP algorithm is conventionally run with static, i.i.d. priors for each signal coefficient. When utilized as a sub-block of a larger message passing algorithm on a larger factor graph, the signal priors (from AMP's perspective) will be changing in response to messages from the rest of the factor graph. We refer to these changing AMP priors as *local priors*.

intuition into why this is the case, it is helpful to think of $\nu_{f_n^{(t)}\to\theta_n^{(t)}}(\theta_n^{(t)})$ as a message that conveys information about the value of $\theta_n^{(t)}$ based on the values of $x_n^{(t)}$ and $s_n^{(t)}$. If $s_n^{(t)}=0$, then by (2), $x_n^{(t)}=0$, thus making $\theta_n^{(t)}$ unobservable. The constant term in (6) reflects the uncertainty due to this unobservability through an infinitely broad, uninformative distribution for $\theta_n^{(t)}$.

To avoid an improper pdf, we modify how this message is derived by regarding our assumed signal model, in which $s_n^{(t)}\in\{0,1\}$, as a limiting case of the model with $s_n^{(t)}\in\{\varepsilon,1\}$ as $\varepsilon\to 0$. For any fixed positive ε, the resulting message $\nu_{f_n^{(t)}\to\theta_n^{(t)}}(\cdot)$ is proper, given by

$$\nu^{\text{mod}}_{f_n^{(t)}\to\theta_n^{(t)}}(\theta_n^{(t)}) = (1-\Omega(\overleftarrow{\pi}_n^{(t)}))\ \mathcal{CN}(\theta_n^{(t)};\tfrac{1}{\varepsilon}\phi_{nt}^i,\tfrac{1}{\varepsilon^2}c_t^i)+\Omega(\overleftarrow{\pi}_n^{(t)})\ \mathcal{CN}(\theta_n^{(t)};\phi_{nt}^i,c_t^i), \tag{7}$$

where

$$\Omega(\pi)\triangleq\frac{\varepsilon^2\pi}{(1-\pi)+\varepsilon^2\pi}. \tag{8}$$

The pdf in (7) is that of a binary Gaussian mixture. If we consider $\varepsilon\ll 1$, the first mixture component is extremely broad, while the second is more "informative," with mean ϕ_n^i and variance c_n^i. The relative weight assigned to each component Gaussian is determined by the term $\Omega(\overleftarrow{\pi}_n^{(t)})$. Notice that the limit of this weighting term is the simple indicator function

$$\lim_{\varepsilon\to 0}\Omega(\pi)=\begin{cases}0 & \text{if } 0\le\pi<1,\\ 1 & \text{if } \pi=1.\end{cases} \tag{9}$$

Since we cannot set $\varepsilon=0$, we instead fix a small positive value, e.g., $\varepsilon=10^{-7}$. In this case, (7) could then be used as the outgoing message. However, this presents a further difficulty: propagating a binary Gaussian mixture forward in time would lead to an exponenial growth in the number of mixture components at subsequent timesteps. This difficulty is a familiar one in the context of switched linear dynamical systems (SLDS's) based on conditional Gaussian models, since such models are not closed under marginalization [32]. To avoid the exponential growth in the number of mixture components, we collapse our binary Gaussian mixture to a single Gaussian component, an approach sometimes referred to as a Gaussian sum approximation [33], [34]. This can be justified by the fact that, for $\varepsilon\ll 1$, $\Omega(\cdot)$ behaves nearly like the indicator function in (9), in which case one of the two Gaussian components will typically have negligible mass.

To carry out the Gaussian sum approximation, we propose the following two schemes. The first is to simply choose a threshold τ that is slightly smaller than 1 and, using (9) as a guide, threshold $\overleftarrow{\pi}_n^{(t)}$ to choose between the two Gaussian components of (7). The resultant message is thus

$$\nu_{f_n^{(t)}\to\theta_n^{(t)}}(\theta_n^{(t)})=\mathcal{CN}(\theta_n^{(t)};\vec{\xi}_n^{(t)},\vec{\psi}_n^{(t)}), \tag{10}$$

with $\vec{\xi}_n^{(t)}$ and $\vec{\psi}_n^{(t)}$ chosen according to

$$\left(\vec{\xi}_n^{(t)}, \vec{\psi}_n^{(t)}\right) = \begin{cases} \left(\frac{1}{\varepsilon}\phi_n^i, \frac{1}{\varepsilon^2}c_n^i\right), & \overleftarrow{\pi}_n^{(t)} \leq \tau \\ \left(\phi_n^i, c_n^i\right), & \overleftarrow{\pi}_n^{(t)} > \tau \end{cases}. \tag{11}$$

The second approach is to perform a second-order Taylor series approximation of $-\log \nu^{\text{mod}}_{f_n^{(t)} \to \theta_n^{(t)}}(\theta_n^{(t)})$ with respect to $\theta_n^{(t)}$, which results in a single Gaussian message, as described in [29]. The latter approach has the advantage of being parameter-free. Empirically, we find that this latter approach works well when changes in the support occur infrequently, e.g., $p_{01} < 0.025$, while the former approach is better suited to more dynamic environments.

In Table II we provide a pseudo-code implementation of our proposed DCS-AMP algorithm that gives the explicit message update equations appropriate for performing a single forward pass. The primary computational burden of DCS-AMP is computing the messages passing between the $\{x_n^{(t)}\}$ and $\{g_m^{(t)}\}$ nodes, a task which can be performed efficiently using matrix-vector products involving $\boldsymbol{A}^{(t)}$ and $\boldsymbol{A}^{(t)^{\mathsf{H}}}$. The resulting overall complexity of DCS-AMP is therefore $\mathcal{O}(TMN)$ flops (flops-per-pass) when filtering (smoothing).[5] The storage requirements are $\mathcal{O}(N)$ and $\mathcal{O}(TN)$ complex numbers when filtering and smoothing, respectively.

IV. Learning the signal model parameters

The signal model of Section II is specified by the Markov chain parameters λ, p_{01}, the Gauss-Markov parameters ζ, α, ρ, and the AWGN variance σ_e^2. It is likely that some or all of these parameters will require tuning in order to best match the unknown signal. To this end, we develop an expectation-maximization (EM) [19] algorithm that works together with the message passing procedure described in Section III-A to learn all of the model parameters in an iterative fashion from the data.

The EM algorithm is appealing for two principal reasons. First, the EM algorithm is a well-studied and principled means of parameter estimation. At every EM iteration, the likelihood is guaranteed to increase until convergence to a local maximum occurs [35]. For multimodal likelihoods, local maxima will, in general, not coincide with the global maximum, but a judicious initialization of parameters can help in ensuring the EM algorithm reaches the global maximum [35]. The second appealing feature of the EM algorithm lies in the fact that its expectation step leverages quantities that have already been computed in the process of executing DCS-AMP, making the EM procedure computationally efficient.

[5]When they exist, fast implicit $\boldsymbol{A}^{(t)}$ operators can provide significant computational savings in high-dimensional problems. Implementing a Fourier transform as a fast Fourier transform (FFT) subroutine, for example, would drop DCS-AMP's complexity from $\mathcal{O}(TMN)$ to $\mathcal{O}(TN\log_2 N)$.

% Define soft-thresholding functions:

$$F_{nt}(\phi;c) \triangleq (1+\gamma_{nt}(\phi;c))^{-1}\Big(\frac{\overleftarrow{\psi}_n^{(t)}\phi+\overleftarrow{\xi}_n^{(t)}c}{\overleftarrow{\psi}_n^{(t)}+c}\Big) \tag{D1}$$

$$G_{nt}(\phi;c) \triangleq (1+\gamma_{nt}(\phi;c))^{-1}\Big(\frac{\overleftarrow{\psi}_n^{(t)}c}{\overleftarrow{\psi}_n^{(t)}+c}\Big) + \gamma_{nt}(\phi;c)|F_n(\phi;c)|^2 \tag{D2}$$

$$F'_{nt}(\phi;c) \triangleq \tfrac{\partial}{\partial\phi}F_{nt}(\phi,c) = \tfrac{1}{c}G_{nt}(\phi;c) \tag{D3}$$

$$\gamma_{nt}(\phi;c) \triangleq \Big(\frac{1-\overleftarrow{\pi}_n^{(t)}}{\overleftarrow{\pi}_n^{(t)}}\Big)\Big(\frac{\overleftarrow{\psi}_n^{(t)}+c}{c}\Big) \times \exp\Big(-\Big[\frac{\overleftarrow{\psi}_n^{(t)}|\phi|^2+\overleftarrow{\xi}_n^{(t)*}c\phi+\overleftarrow{\xi}_n^{(t)}c\phi^*-c|\overleftarrow{\xi}_n^{(t)}|^2}{c(\overleftarrow{\psi}_n^{(t)}+c)}\Big]\Big) \tag{D4}$$

% Begin passing messages . . .

for $t = 1,\dots,T$:

% Execute the **(into)** phase . . .

$$\overleftarrow{\pi}_n^{(t)} = \frac{\overrightarrow{\lambda}_n^{(t)}\cdot\overleftarrow{\lambda}_n^{(t)}}{(1-\overrightarrow{\lambda}_n^{(t)})\cdot(1-\overleftarrow{\lambda}_n^{(t)})+\overrightarrow{\lambda}_n^{(t)}\cdot\overleftarrow{\lambda}_n^{(t)}} \quad \forall n \tag{A1}$$

$$\overleftarrow{\psi}_n^{(t)} = \frac{\overrightarrow{\kappa}_n^{(t)}\cdot\overleftarrow{\kappa}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}+\overleftarrow{\kappa}_n^{(t)}} \quad \forall n \tag{A2}$$

$$\overleftarrow{\xi}_n^{(t)} = \overleftarrow{\psi}_n^{(t)}\cdot\Big(\frac{\overrightarrow{\eta}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}}+\frac{\overleftarrow{\eta}_n^{(t)}}{\overleftarrow{\kappa}_n^{(t)}}\Big) \quad \forall n \tag{A3}$$

% Initialize AMP-related variables . . .

$\forall m: z_{mt}^1 = y_m^{(t)}, \forall n: \mu_{nt}^1 = 0,$ and $c_t^1 = 100\cdot\sum_{n=1}^N \psi_n^{(t)}$

% Execute the **(within)** phase using AMP . . .

for $i = 1,\dots,I,$:

$$\phi_{nt}^i = \textstyle\sum_{m=1}^M A_{mn}^{(t)*} z_{mt}^i + \mu_{nt}^i \quad \forall n \tag{A4}$$

$$\mu_{nt}^{i+1} = F_{nt}(\phi_{nt}^i; c_t^i) \quad \forall n \tag{A5}$$

$$v_{nt}^{i+1} = G_{nt}(\phi_{nt}^i; c_t^i) \quad \forall n \tag{A6}$$

$$c_t^{i+1} = \sigma_e^2 + \tfrac{1}{M}\textstyle\sum_{n=1}^N v_{nt}^{i+1} \tag{A7}$$

$$z_{mt}^{i+1} = y_m^{(t)} - \boldsymbol{a}_m^{(t)\mathsf{T}}\boldsymbol{\mu}_t^{i+1} + \tfrac{z_{mt}^i}{M}\textstyle\sum_{n=1}^N F'_{nt}(\phi_{nt}^i; c_t^i) \quad \forall m \tag{A8}$$

end

$$\hat{x}_n^{(t)} = \mu_{nt}^{I+1} \quad \forall n \qquad \text{% Store current estimate of } x_n^{(t)} \tag{A9}$$

% Execute the **(out)** phase . . .

$$\overrightarrow{\pi}_n^{(t)} = \Big(1+\Big(\frac{\overleftarrow{\pi}_n^{(t)}}{1-\overleftarrow{\pi}_n^{(t)}}\Big)\gamma_{nt}(\phi_{nt}^I; c_t^{I+1})\Big)^{-1} \quad \forall n \tag{A10}$$

$$(\overrightarrow{\xi}_n^{(t)}, \overrightarrow{\psi}_n^{(t)}) = \begin{cases} (\phi_n^I/\varepsilon, c_t^{I+1}/\varepsilon^2), & \overleftarrow{\pi}_n^{(t)} \le \tau \\ (\phi_n^I, c_t^{I+1}), & \text{o.w.} \end{cases} \quad \forall n \quad (\varepsilon \ll 1) \tag{A11}$$

% Execute the **(across)** phase from $\theta_n^{(t)}$ to $\theta_n^{(t+1)}$ . . .

$$\overrightarrow{\lambda}_n^{(t+1)} = \frac{p_{10}(1-\overrightarrow{\lambda}_n^{(t)})(1-\overrightarrow{\pi}_n^{(t)})+(1-p_{01})\overrightarrow{\lambda}_n^{(t)}\overrightarrow{\pi}_n^{(t)}}{(1-\overrightarrow{\lambda}_n^{(t)})(1-\overrightarrow{\pi}_n^{(t)})+\overrightarrow{\lambda}_n^{(t)}\overrightarrow{\pi}_n^{(t)}} \quad \forall n \tag{A12}$$

$$\overrightarrow{\eta}_n^{(t+1)} = (1-\alpha)\Big(\frac{\overrightarrow{\kappa}_n^{(t)}\overrightarrow{\psi}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}+\overrightarrow{\psi}_n^{(t)}}\Big)\Big(\frac{\overrightarrow{\eta}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}}+\frac{\overrightarrow{\xi}_n^{(t)}}{\overrightarrow{\psi}_n^{(t)}}\Big)+\alpha\zeta \quad \forall n \tag{A13}$$

$$\overrightarrow{\kappa}_n^{(t+1)} = (1-\alpha)^2\Big(\frac{\overrightarrow{\kappa}_n^{(t)}\overrightarrow{\psi}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}+\overrightarrow{\psi}_n^{(t)}}\Big)+\alpha^2\rho \quad \forall n \tag{A14}$$

end

TABLE II: DCS-AMP steps for filtering mode, or the forward portion of a single forward/backward pass in smoothing mode.

We let $\Gamma \triangleq \{\lambda, p_{0_1}, \zeta, \alpha, \rho, \sigma_e^2\}$ denote the set of all model parameters, and let Γ^k denote the set of parameter estimates at the k^{th} EM iteration. The objective of the EM procedure is to find parameter estimates that maximize the data likelihood $p(\bar{\boldsymbol{y}}|\Gamma)$. Since it is often computationally intractable to perform this maximization, the EM algorithm incorporates additional "hidden" data and iterates between two steps: *(i)* evaluating the conditional expectation of the log likelihood of the hidden data given the observed data, $\bar{\boldsymbol{y}}$, and the current estimates of the parameters, Γ^k, and *(ii)* maximizing this expected log likelihood with respect to the model parameters. For all parameters except the noise variance, σ_e^2, we use $\bar{\boldsymbol{s}}$ and $\bar{\boldsymbol{\theta}}$ as the hidden data, while for σ_e^2 we use $\bar{\boldsymbol{x}}$.

Before running DCS-AMP, the model parameters are initialized using any available prior knowledge. If operating in smoothing mode, DCS-AMP performs an initial forward/backward pass, as described in Section III-A. Upon completing this first pass, estimates of the marginal posterior distributions are available for each of the underlying random variables. Additionally, belief propagation can provide pairwise joint posterior distributions, e.g., $p\big(s_n^{(t)}, s_n^{(t-1)}|\bar{\boldsymbol{y}}\big)$, for any variable nodes connected by a common factor node [36]. With these marginal, and pairwise joint, posterior distributions, it is possible to produce closed-form solutions for performing steps *(i)* and *(ii)* above. We adopt a Gauss-Seidel scheme, performing coordinate-wise maximization, e.g.,

$$\lambda^{k+1} = \underset{\lambda}{\operatorname{argmax}}\, \mathrm{E}_{\bar{\boldsymbol{s}},\bar{\boldsymbol{\theta}}|\bar{\boldsymbol{y}}} \left[\log p(\bar{\boldsymbol{y}}, \bar{\boldsymbol{s}}, \bar{\boldsymbol{\theta}}; \lambda, \Gamma^k \backslash \{\lambda^k\}) \big| \bar{\boldsymbol{y}}; \Gamma^k\right].$$

The EM procedure is performed after each forward/backward pass, leading to a convergent sequence of parameter estimates. If operating in filtering mode, the procedure is similar, however the EM procedure is run after each recovered timestep using only causally available posterior estimates.

In Table III, we provide the EM update equations for each of the parameters of our signal model, assuming DCS-AMP is operating in smoothing mode.

V. Incorporating Additional Structure

In Sections II - IV we described a signal model for the dynamic CS problem and summarized a message passing algorithm for making inferences under this model, while iteratively learning the model parameters via EM. We also hinted that the model could be generalized to incorporate additional, or more complex, forms of structure. In this section we will elaborate on this idea, and illustrate one such generalization.

Recall that, in Section II, we introduced hidden variables $\bar{\boldsymbol{s}}$ and $\bar{\boldsymbol{\theta}}$ in order to characterize the structure in the signal coefficients. An important consequence of introducing these hidden variables was that they

% Define key quantities obtained from AMP-MMV at iteration k:

$$\mathrm{E}\big[s_n^{(t)}\big|\bar{\boldsymbol{y}}\big] = \frac{\big(\overrightarrow{\lambda}_n^{(t)}\overrightarrow{\pi}_n^{(t)}\overleftarrow{\lambda}_n^{(t)}\big)}{\big(\overrightarrow{\lambda}_n^{(t)}\overrightarrow{\pi}_n^{(t)}\overleftarrow{\lambda}_n^{(t)}+(1-\overrightarrow{\lambda}_n^{(t)})(1-\overrightarrow{\pi}_n^{(t)})(1-\overleftarrow{\lambda}_n^{(t)})\big)} \tag{Q1}$$

$$\mathrm{E}\big[s_n^{(t)}s_n^{(t-1)}\big|\bar{\boldsymbol{y}}\big] = p\big(s_n^{(t)}=1, s_n^{(t-1)}=1\big|\bar{\boldsymbol{y}}\big) \tag{Q2}$$

$$\tilde{v}_n^{(t)} \triangleq \mathrm{var}\{\theta_n^{(t)}|\bar{\boldsymbol{y}}\} = \left(\tfrac{1}{\overrightarrow{\kappa}_n^{(t)}} + \tfrac{1}{\overrightarrow{\psi}_n^{(t)}} + \tfrac{1}{\overleftarrow{\kappa}_n^{(t)}}\right)^{-1} \tag{Q3}$$

$$\tilde{\mu}_n^{(t)} \triangleq \mathrm{E}[\theta_n^{(t)}|\bar{\boldsymbol{y}}] = \tilde{v}_n^{(t)} \cdot \left(\tfrac{\overrightarrow{\eta}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}} + \tfrac{\overrightarrow{\xi}_n^{(t)}}{\overrightarrow{\psi}_n^{(t)}} + \tfrac{\overleftarrow{\eta}_n^{(t)}}{\overleftarrow{\kappa}_n^{(t)}}\right) \tag{Q4}$$

$v_n^{(t)} \triangleq \mathrm{var}\big\{x_n^{(t)}\big|\bar{\boldsymbol{y}}\big\}$ % See (A6) of Table II

$\mu_n^{(t)} \triangleq \mathrm{E}\big[x_n^{(t)}\big|\bar{\boldsymbol{y}}\big]$ % See (A5) of Table II

% EM update equations:

$$\lambda^{k+1} = \tfrac{1}{N}\textstyle\sum_{n=1}^N \mathrm{E}\big[s_n^{(1)}\big|\bar{\boldsymbol{y}}\big] \tag{E1}$$

$$p_{01}^{k+1} = \frac{\sum_{t=2}^T\sum_{n=1}^N \mathrm{E}\big[s_n^{(t-1)}\big|\bar{\boldsymbol{y}}\big] - \mathrm{E}\big[s_n^{(t)}s_n^{(t-1)}\big|\bar{\boldsymbol{y}}\big]}{\sum_{t=2}^T\sum_{n=1}^N \mathrm{E}\big[s_n^{(t-1)}\big|\bar{\boldsymbol{y}}\big]} \tag{E2}$$

$$\begin{aligned}\zeta^{k+1} = {}& \left(\tfrac{N(T-1)}{\rho^k} + \tfrac{N}{(\sigma^2)^k}\right)^{-1}\Big(\tfrac{1}{(\sigma^2)^k}\textstyle\sum_{n=1}^N \tilde{\mu}_n^{(1)} \\ & + \textstyle\sum_{t=2}^T\sum_{n=1}^N \tfrac{1}{\alpha^k\rho^k}\big(\tilde{\mu}_n^{(t)} - (1-\alpha^k)\tilde{\mu}_n^{(t-1)}\big)\Big)\end{aligned} \tag{E3}$$

$$\alpha^{k+1} = \tfrac{1}{4N(T-1)}\Big(\mathfrak{b} - \sqrt{\mathfrak{b}^2 + 8N(T-1)\mathfrak{c}}\Big) \tag{E4}$$

where:

$$\begin{aligned}\mathfrak{b} \triangleq {}& \tfrac{2}{\rho^k}\textstyle\sum_{t=2}^T\sum_{n=1}^N \mathfrak{Re}\big\{E[\theta_n^{(t)*}\theta_n^{(t-1)}|\bar{\boldsymbol{y}}]\big\} \\ & -\mathfrak{Re}\{(\tilde{\mu}_n^{(t)} - \tilde{\mu}_n^{(t-1)})^*\zeta^k\} - \tilde{v}_n^{(t-1)} - |\tilde{\mu}_n^{(t-1)}|^2 \\ \mathfrak{c} \triangleq {}& \tfrac{2}{\rho^k}\textstyle\sum_{t=2}^T\sum_{n=1}^N \tilde{v}_n^{(t)} + |\tilde{\mu}_n^{(t)}|^2 + \tilde{v}_n^{(t-1)} + |\tilde{\mu}_n^{(t-1)}|^2 \\ & -2\mathfrak{Re}\big\{E[\theta_n^{(t)*}\theta_n^{(t-1)}|\bar{\boldsymbol{y}}]\big\}\end{aligned}$$

$$\begin{aligned}\rho^{k+1} = {}& \tfrac{1}{(\alpha^k)^2N(T-1)}\textstyle\sum_{t=2}^T\sum_{n=1}^N \tilde{v}_n^{(t)} + |\tilde{\mu}_n^{(t)}|^2 \\ & +(\alpha^k)^2|\zeta^k|^2 - 2(1-\alpha^k)\mathfrak{Re}\big\{E[\theta_n^{(t)*}\theta_n^{(t-1)}|\bar{\boldsymbol{y}}]\big\} \\ & -2\alpha^k\mathfrak{Re}\big\{\tilde{\mu}_n^{(t)*}\zeta^k\big\} + 2\alpha^k(1-\alpha^k)\mathfrak{Re}\big\{\tilde{\mu}_n^{(t-1)*}\zeta^k\big\} \\ & +(1-\alpha^k)(\tilde{v}_n^{(t-1)} + |\tilde{\mu}_n^{(t-1)}|^2)\end{aligned} \tag{E5}$$

$$(\sigma_e^2)^{k+1} = \tfrac{1}{TM}\Big(\textstyle\sum_{t=1}^T \|\boldsymbol{y}^{(t)} - \boldsymbol{A}\boldsymbol{\mu}^{(t)}\|^2 + \mathbf{1}_N^T\boldsymbol{v}^{(t)}\Big) \tag{E6}$$

TABLE III: EM update equations for the signal model parameters of Section II.

made each signal coefficient $x_n^{(t)}$ conditionally independent of the remaining coefficients in $\bar{\boldsymbol{x}}$, given $s_n^{(t)}$ and $\theta_n^{(t)}$. This conditional independence served an important algorithmic purpose since it allowed us to apply the AMP algorithm, which requires independent local priors, within our larger inference procedure.

One way to incorporate additional structure into the signal model of Section II is to generalize our choices of $p(\bar{\boldsymbol{s}})$ and $p(\bar{\boldsymbol{\theta}})$. As an example, suppose that we wish to expand our Bernoulli-Gaussian signal model to one in which signal coefficients are marginally distributed according to a Bernoulli-Gaussian-mixture, i.e.,

$$p(x_n^{(t)}) = \lambda_0^{(t)}\delta(x_n^{(t)}) + \sum_{d=1}^D \lambda_d^{(t)}\mathcal{CN}(x_n^{(t)};\zeta_d,\sigma_d^2),$$

where $\sum_{d=0}^D \lambda_d^{(t)} = 1$. Since we still wish to preserve the slow time-variations in the support and smooth evolution of non-zero amplitudes, a natural choice of hidden variables is $\{\bar{\boldsymbol{s}}, \bar{\boldsymbol{\theta}}_1, \ldots, \bar{\boldsymbol{\theta}}_D\}$, where

$s_n^{(t)} \in \{0, 1, \ldots, D\}$, and $\theta_{d,n}^{(t)} \in \mathbb{C}$, $d = 1, \ldots, D$. The relationship between $x_n^{(t)}$ and the hidden variables then generalizes to:

$$p(x_n^{(t)}|s_n^{(t)}, \theta_{1,n}^{(t)}, \ldots, \theta_{D,n}^{(t)}) = \begin{cases} \delta(x_n^{(t)}), & s_n^{(t)} = 0, \\ \delta(x_n^{(t)} - \theta_{d,n}^{(t)}), & s_n^{(t)} = d \neq 0. \end{cases}$$

To model the slowly changing support, we specify $p(\bar{\boldsymbol{s}})$ using a $(D+1)$-state Markov chain defined by the transition probabilities $p_{0d} \triangleq \Pr\{s_n^{(t)} = 0 | s_n^{(t-1)} = d\}$ and $p_{d0} \triangleq \Pr\{s_n^{(t)} = d | s_n^{(t-1)} = 0\}$, $d = 1, \ldots, D$. For simplicity, we assume that state transitions cannot occur between active mixture components, i.e., $\Pr(s_n^{(t)} = d | s_n^{(t-1)} = e) = 0$ when $d \neq e \neq 0$. For each amplitude time-series we again use independent Gauss-Markov processes to model smooth evolutions in the amplitudes of active signal coefficients, i.e.,

$$\theta_{d,n}^{(t)} = (1 - \alpha_d)\big(\theta_{d,n}^{(t-1)} - \zeta_d\big) + \alpha_d w_{d,n}^{(t)} + \zeta_d,$$

where $w_{d,n}^{(t)} \sim \mathcal{CN}(0, \rho_d)$.

As a consequence of this generalized signal model, a number of the message computations of Section III-B must be modified. For steps **(into)** and **(across)**, it is largely straightforward to extend the computations to account for the additional hidden variables. For step **(within)**, the modifications will affect the AMP thresholding equations defined in (D1) - (D4) of Table II. Details on a Bernoulli-Gaussian-mixture AMP algorithm can be found in [30]. For the **(out)** step, we will encounter difficulties applying standard sum-product update rules to compute the messages $\{\nu_{f_n^{(t)} \to \theta_{d,n}^{(t)}}(\cdot)\}_{d=1}^{D}$. As in the Bernoulli-Gaussian case, we consider a modification of our assumed signal model that incorporates an $\varepsilon \ll 1$ term, and use Taylor series approximations of the resultant messages to collapse a $(D+1)$-ary Gaussian mixture to a single Gaussian. More information on this procedure can be found in [37].

VI. Empirical Study

We now describe the results of an empirical study of dynamic CS.[6] The primary performance metric that we used in all of our experiments, which we refer to as the time-averaged normalized MSE (TNMSE), is defined as

$$\text{TNMSE}(\bar{\boldsymbol{x}}, \hat{\bar{\boldsymbol{x}}}) \triangleq \frac{1}{T} \sum_{t=1}^{T} \frac{\|\boldsymbol{x}^{(t)} - \hat{\boldsymbol{x}}^{(t)}\|_2^2}{\|\boldsymbol{x}^{(t)}\|_2^2},$$

where $\hat{\boldsymbol{x}}^{(t)}$ is an estimate of $\boldsymbol{x}^{(t)}$.

[6]Code for reproducing our results is available at ece.osu.edu/~schniter/DCSturboAMP.

Unless otherwise noted, the following settings were used for DCS-AMP in our experiments. First, DCS-AMP was run as a smoother, with a total of 5 forward/backward passes. The number of inner AMP iterations I for each **(within)** step was $I = 25$, with a possibility for early termination if the change in the estimated signal, $\boldsymbol{\mu}_t^i$, fell below a predefined threshold from one iteration to the next, i.e., $\frac{1}{N}\|\boldsymbol{\mu}_t^i - \boldsymbol{\mu}_t^{i-1}\|_2 < 10^{-5}$. Equation (A9) of Table II was used to produce $\hat{\boldsymbol{x}}^{(t)}$, which corresponds to an MMSE estimate of $\boldsymbol{x}^{(t)}$ under DCS-AMP's estimated posteriors $p(x_n^{(t)}|\bar{\boldsymbol{y}})$. The amplitude approximation parameter ε from (7) was set to $\varepsilon = 10^{-7}$, while the threshold τ from (11) was set to $\tau = 0.99$. In our experiments, we found DCS-AMP's performance to be relatively insensitive to the value of ε provided that $\varepsilon \ll 1$. The choice of τ should be selected to provide a balance between allowing DCS-AMP to track amplitude evolutions on signals with rapidly varying supports and preventing DCS-AMP from prematurely gaining too much confidence in its estimate of the support. We found that the choice $\tau = 0.99$ works well over a broad range of problems. When the estimated transition probability $p_{01} < 0.025$, DCS-AMP automatically switched from the threshold method to the Taylor series method of computing (10), which is advantageous because it is parameter-free.

A. Performance across the sparsity-undersampling plane

Two factors that have a significant effect on the performance of any CS algorithm are the sparsity $|\mathcal{S}^{(t)}|$ of the underlying signal, and the number of measurements M. Consequently, much can be learned about an algorithm by manipulating these factors and observing the resulting change in performance. To this end, we studied DCS-AMP's performance across the sparsity-undersampling plane [38], which is parameterized by two quantities, the *normalized sparsity ratio*, $\beta \triangleq \mathrm{E}[|\mathcal{S}^{(t)}|]/M$, and the *undersampling ratio*, $\delta \triangleq M/N$. For a given (δ, β) pair (with N fixed at 512), sample realizations of $\bar{\boldsymbol{s}}$, $\bar{\boldsymbol{\theta}}$, and $\bar{\boldsymbol{e}}$ were drawn from their respective priors, and elements of a time-varying $\boldsymbol{A}^{(t)}$ were drawn from i.i.d. zero-mean complex circular Gaussians, with all columns subsequently scaled to have unit ℓ_2-norm, thus generating $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$. Given knowledge of the signal model parameters (i.e., without the need for EM learning), a recovery was obtained by executing DCS-AMP in smoothing mode.

As a performance benchmark, we used the support-aware Kalman smoother. In the case of linear dynamical systems with jointly Gaussian signal and observations, the Kalman filter (smoother) is known to provide MSE-optimal causal (non-causal) signal estimates [39]. When the signal is Bernoulli-Gaussian, the Kalman filter/smoother is no longer optimal. However, a lower bound on the achievable MSE can be obtained using the support-aware Kalman filter (SKF) or smoother (SKS). Since the classical state-space formulation of the Kalman filter does not easily yield the support-aware bound, we turn to an

alternative view of Kalman filtering as an instance of message passing on an appropriate factor graph [40]. For this, it suffices to use the factor graph of Fig. 1 with $\{s_n^{(t)}\}$ treated as fixed, known quantities. Following the standard sum-product algorithm rules results in a message passing algorithm in which all messages are Gaussian, and no message approximations are required. Then, by running loopy Gaussian belief propagation until convergence, we are guaranteed that the resultant posterior means constitute the MMSE estimate of $\bar{\boldsymbol{x}}$, in accordance with [22, Claim 5].

Finally, to quantify the improvement obtained by exploiting temporal correlation, signal recovery was also explored using the Bernoulli-Gaussian AMP algorithm (BG-AMP) independently at each timestep (i.e., ignoring temporal structure in the support and amplitudes), accomplished by passing messages only within the dashed boxes of Fig. 1 using $p\big(x_n^{(t)}\big)$ from (4) as AMP's prior.[7]

In Fig. 3, we present three plots from a representative experiment. The leftmost plot shows the TNMSE of the SKS (in dB) across the sparsity-undersampling plane; the center plot shows the TNMSE of DCS-AMP; the rightmost plot shows the TNMSE of BG-AMP. The results shown were averaged over more than 500 independent trials at each (δ, β) pair. For this experiment, signal model parameters were set at $N = 512$, $T = 25$, $p_{01} = 0.05$, $\zeta = 0$, $\alpha = 0.01$, $\sigma^2 = 1$, and a noise variance, σ_e^2, chosen to yield a signal-to-noise ratio (SNR) of 15 dB. (M, λ) were set based on specific (δ, β) pairs, and p_{10} was set so as to keep the expected number of active coefficients constant across time. It is interesting to observe that the performance of both the SKS and DCS-AMP are only weakly dependent on the undersampling ratio δ. In contrast, the structure-agnostic BG-AMP algorithm is strongly affected. This is one of the principal benefits of incorporating temporal structure; it makes it possible to tolerate more substantial amounts of undersampling, particularly when the underlying signal is less sparse.

B. Performance vs p_{01} and α

The temporal correlation of our time-varying sparse signal model is largely dictated by two parameters, the support transition probability p_{01} and the amplitude forgetting factor α. Therefore, it is worth investigating how the performance of DCS-AMP is affected by these two parameters. In an experiment similar to that of Fig. 3, we tracked the performance of DCS-AMP, the SKS, and BG-AMP across a plane of (p_{01}, α) pairs. The active-to-inactive transition probability p_{01} was swept linearly over the range $[0, 0.15]$, while the Gauss-Markov amplitude forgetting factor α was swept logarithmically over the range

[7]Experiments were also run that compared performance against Basis Pursuit Denoising (BPDN) [41] with genie-aided parameter tuning (solved using the SPGL1 solver [42]). However, this was found to yield higher TNMSE than BG-AMP, and at higher computational cost.

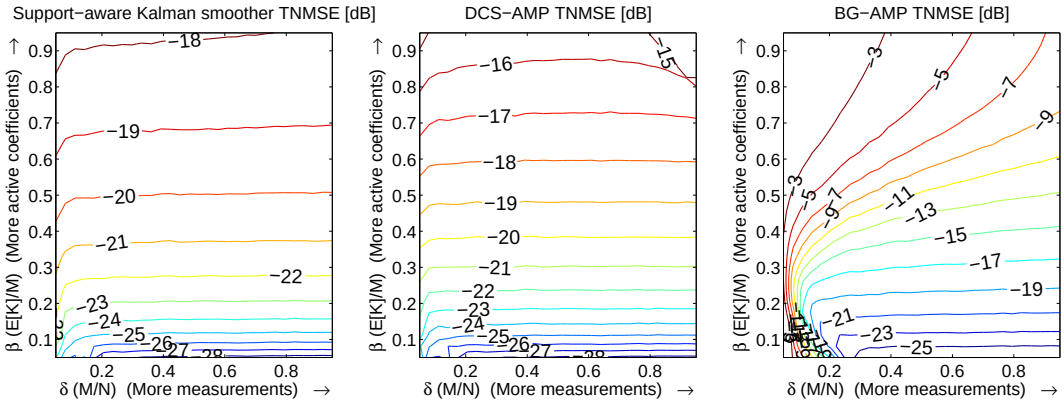

Fig. 3: A plot of the TNMSE (in dB) of the SKS (leftmost), DCS-AMP (center) and BG-AMP (rightmost) across the sparsity-undersampling plane, for temporal correlation parameters $p_{01} = 0.05$ and $\alpha = 0.01$.

$[0.001, 0.95]$. To help interpret the meaning of these parameters, we note that the fraction of the support that is expected to change from one timestep to the next is given by $2\,p_{01}$, and that the Pearson correlation coefficient between temporally adjacent amplitude variables is $1 - \alpha$.

In Fig. 4 we plot the TNMSE (in dB) of the SKS and DCS-AMP as a function of the percentage of the support that changes from one timestep to the next (i.e., $2p_{01} \times 100$) and the logarithmic value of α for a signal model in which $\delta = 1/3$ and $\beta = 0.45$, with remaining parameters set as before. Since BG-AMP is agnostic to temporal correlation, its performance is insensitive to the values of p_{01} and α. Therefore, we do not include a plot of the performance of BG-AMP, but note that it achieved a TNMSE of -11.6 dB across the plane. For both the SKS and DCS-AMP, we see that performance improves with increasing amplitude correlation (moving leftward). This behavior, although intuitive, is in contrast to the relationship between performance and correlation found in the MMV problem [29], [43], primarily due to the fact that the measurement matrix is static for all timesteps in the classical MMV problem, whereas here it varies with time. Since the SKS has perfect knowledge of the support, its performance is only weakly dependent on the rate of support change. From Fig. 4, we see that DCS-AMP performance shows a modest dependence on the rate of support change, but nevertheless is capable of managing rapid temporal changes in support.

C. Recovery of an MRI image sequence

While the above simulations demonstrate the effectiveness of DCS-AMP in recovering signals generated according to our signal model, it remains to be seen whether the signal model itself is suitable for describing practical dynamic CS signals. To address this question, we tested the performance of DCS-AMP on a dynamic MRI experiment first performed in [44]. The experiment consists of recovering

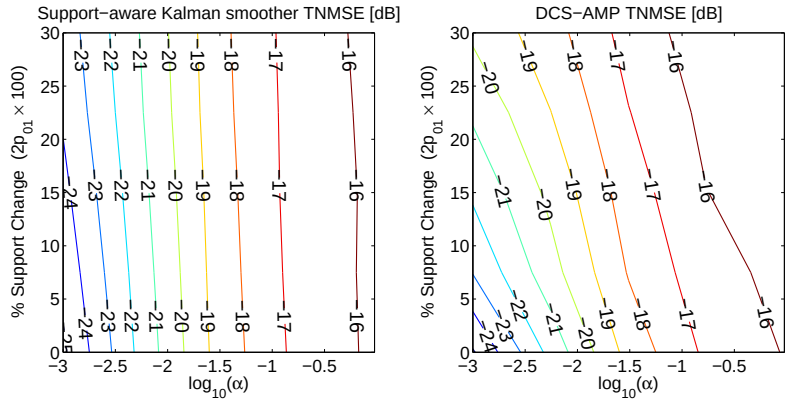

Fig. 4: TNMSE (in dB) of the SKS (leftmost) and DCS-AMP (rightmost) as a function of the model parameters p_{01} and α, for undersampling ratio $\delta = 1/3$ and sparsity ratio $\beta = 0.45$. BG-AMP achieved a TNMSE of -11.6 dB across the plane.

a sequence of 10 MRI images of the larynx, each 256×256 pixels in dimension. (See Fig. 5.) The measurement matrices were never stored explicitly due to the prohibitive sizes involves, but were instead treated as the composition of three linear operations, $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{F}\boldsymbol{W}^{\mathsf{T}}$. The first operation, $\boldsymbol{W}^{\mathsf{T}}$, was the synthesis of the underlying image from a sparsifying 2-D, 2-level Daubechies-4 wavelet transform representation. The second operation, $\boldsymbol{F}$, was a 2-D Fourier transform that yielded the k-space coefficients of the image. The third operation, $\boldsymbol{M}$, was a sub-sampling mask that kept only a fraction of the available k-space data.

Since the image transform coefficients are compressible rather than sparse, the SKF/SKS no longer serves as an appropriate algorithmic benchmark. Instead, we compare performance against Modified-CS [6], as well as timestep-independent Basis Pursuit.[8] As reported in [6], Modified-CS demonstrates that substantial improvements can be obtained over temporally agnostic methods.

Since the statistics of wavelet coefficients at different scales are often highly dissimilar (e.g., the coarsest-scale approximation coefficients are usually much less sparse than those at finer scales, and are also substantially larger in magnitude), we allowed our EM procedure to learn different parameters for different wavelet scales. Using $\mathcal{G}_1$ to denote the indices of the coarsest-scale "approximation" coefficients, and $\mathcal{G}_2$ to denote the finer-scale "wavelet" coefficients, DCS-AMP was initialized with the following parameter choices: $\lambda_{\mathcal{G}_1} = 0.99$, $\lambda_{\mathcal{G}_2} = 0.01$, $p_{01} = 0.01$, $\zeta_{\mathcal{G}_1} = \zeta_{\mathcal{G}_2} = 0$, $\alpha_{\mathcal{G}_1} = \alpha_{\mathcal{G}_2} = 0.05$, $\rho_{\mathcal{G}_1} = 10^5$, $\rho_{\mathcal{G}_2} = 10^4$, and $\sigma_e^2 = 0.01$, and run in filtering mode with $I = 10$ inner AMP iterations.

[8]Modified-CS is available at http://home.engineering.iastate.edu/~luwei/modcs/index.html. Basis Pursuit was solved using the ℓ_1-MAGIC equality-constrained primal-dual solver (chosen since it is used as a subroutine within Modified-CS), available at http://users.ece.gatech.edu/~justin/l1magic/.

Algorithm	TNMSE (dB)	Runtime
Basis Pursuit	-17.22	47 min
Modified-CS	-34.30	7.39 hrs
DCS-AMP (Filter)	**-34.62**	**8.08 sec**

TABLE IV: Performance on dynamic MRI dataset from [44] with increased sampling rate at initial timestep.

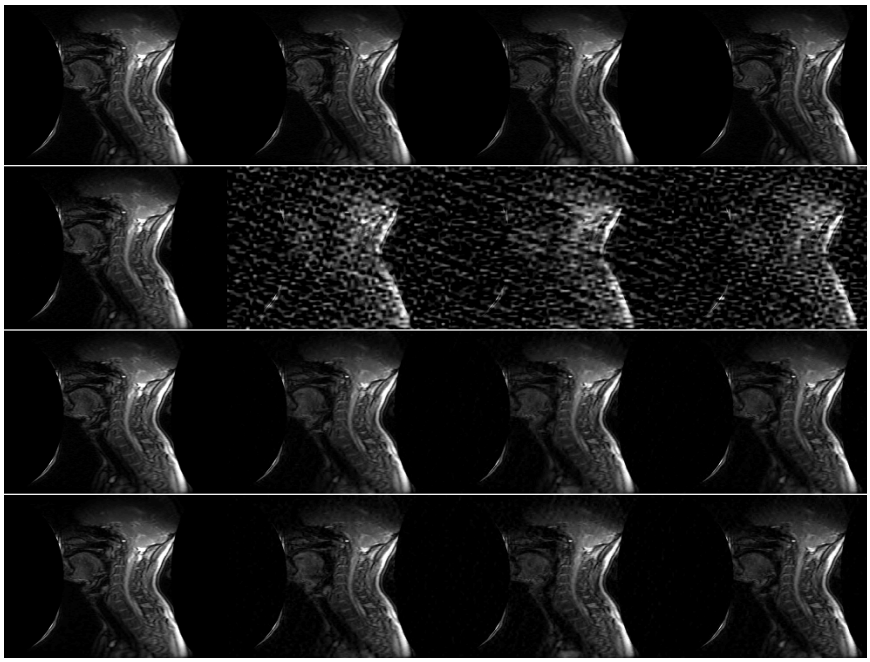

Fig. 5: Frames 1, 2, 5, and 10 of the dynamic MRI image sequence of (from top to bottom): the fully sampled dataset, Basis Pursuit, Modified-CS, and DCS-AMP, with increased sampling rate at initial timestep.

In Table IV we summarize the performance of four different estimators: timestep-independent Basis Pursuit, which performs independent ℓ_1 minimizations at each timestep, Modified-CS, and DCS-AMP (operating in filtering mode). In this experiment, per the setup described in [44], the initial timestep was sampled at 50% of the Nyquist rate, i.e., $M = N/2$, while subsequent timesteps were sampled at 16% of the Nyquist rate. Both Modified-CS and DCS-AMP substantially outperform Basis Pursuit with respect to TNMSE, with DCS-AMP showing a slight advantage over Modified-CS. Despite the similar TNMSE performance, note that DCS-AMP runs in seconds, whereas Modified-CS takes multiple hours. In Fig. 5, we plot the true images along with the recoveries for this experiment, which show severe degradation for Basis Pursuit on all but the initial timestep.

In practice, it may not be possible to acquire an increased number of samples at the initial timestep.

Algorithm	TNMSE (dB)	Runtime
Basis Pursuit	-16.83	47.61 min
Modified-CS	-17.18	7.78 hrs
DCS-AMP (Filter)	**-29.51**	**7.27 sec**

TABLE V: Performance on dynamic MRI dataset from [44] with identical sampling rate at every timestep.

We therefore repeated the experiment while sampling at 16% of the Nyquist rate at every timestep. The results, shown in Table V, show that DCS-AMP's performance degrades by about 5 dB, while Modified-CS suffers a 14 dB reduction, illustrating that, when the estimate of the initial support is poor, Modified-CS struggles to outperform Basis Pursuit.

D. Recovery of a CS audio sequence

In another experiment using real-world data, we used DCS-AMP to recover an audio signal from sub-Nyquist samples. In this case, we employ the Bernoulli-Gaussian-mixture signal model proposed for DCS-AMP in Section V. The audio clip is a 7 second recording of a trumpet solo, and contains a succession of rapid changes in the trumpet's pitch. Such a recording presents a challenge for CS methods, since the signal will be only compressible, and not sparse. The clip, sampled at a rate of 11 kHz, was divided into $T = 54$ non-overlapping segments of length $N = 1500$. Using the discrete cosine transform (DCT) as a sparsifying basis, linear measurements were obtained using a time-invariant i.i.d. Gaussian sensing matrix.

In Fig. 6 we plot the magnitude of the DCT coefficients of the audio signal on a dB scale. Beyond the temporal correlation evident in the plot, it is also interesting to observe that there is a non-trivial amount of frequency correlation (correlation across the index $[n]$), as well as a large dynamic range. We performed recoveries using four techniques: BG-AMP, GM-AMP (a temporally agnostic Bernoulli-Gaussian-mixture AMP algorithm with $D = 4$ Gaussian mixture components), DCS-(BG)-AMP, and DCS-GM-AMP (the Bernoulli-Gaussian-mixture dynamic CS model described in Section V, with $D = 4$). For each algorithm, EM learning of the model parameters was performed using straightforward variations of the procedure described in Section IV, with model parameters initialized automatically using simple heuristics described in [30]. Moreover, unique model parameters were learned at each timestep (with the exception of support transition probabilities). Furthermore, since our model of hidden amplitude evolutions was poorly matched to this audio signal, we fixed $\alpha = 1$.

In Table VI we present the results of applying each algorithm to the audio dataset for three different undersampling rates, δ. For each algorithm, both the TNMSE in dB and the runtime in seconds are pro-

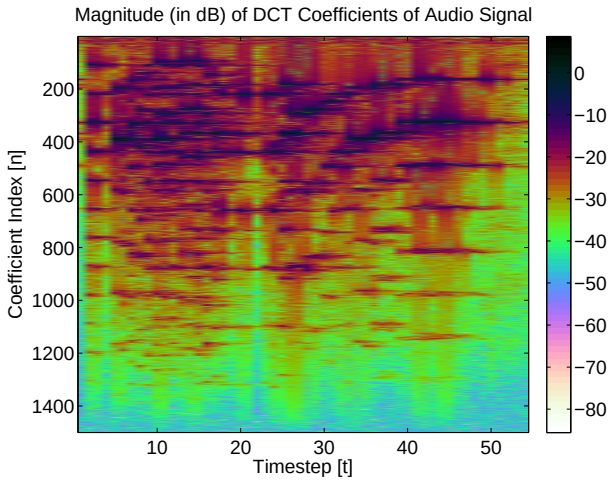

Fig. 6: DCT coefficient magnitudes (in dB) of an audio signal.

		Undersampling Rate		
		$\delta = \frac{1}{2}$	$\delta = \frac{1}{3}$	$\delta = \frac{1}{5}$
Algorithm	BG-AMP	-16.88 (dB) | **9.11 (s)**	-11.67 (dB) | **8.27 (s)**	-8.56 (dB) | **6.63 (s)**
	GM-AMP ($D = 4$)	-17.49 (dB) | 19.36 (s)	-13.74 (dB) | 17.48 (s)	-10.23 (dB) | 15.98 (s)
	DCS-BG-AMP	-19.84 (dB) | 10.20 (s)	-14.33 (dB) | 8.39 (s)	-11.40 (dB) | 6.71 (s)
	DCS-GM-AMP ($D = 4$)	**-21.33 (dB)** | 20.34 (s)	**-16.78 (dB)** | 18.63 (s)	**-12.49 (dB)** | 10.13 (s)

TABLE VI: Performance on audio CS dataset (TNMSE (dB) | Runtime (s)) of two temporally independent algorithms, BG-AMP and GM-AMP, and two temporally structured algorithms, DCS-BG-AMP and DCS-GM-AMP.

vided. Overall, we see that performance improves at each undersampling rate as the signal model becomes more expressive. While GM-AMP outperforms BG-AMP at all undersampling rates, it is surpassed by DCS-BG-AMP and DCS-GM-AMP, with DCS-GM-AMP offering the best TNMSE performance. Indeed, we observe that one can obtain comparable, or even better, performance with an undersampling rate $\delta = \frac{1}{5}$ using DCS-BG-AMP or DCS-GM-AMP, with that obtained using BG-AMP with an undersampling rate $\delta = \frac{1}{3}$.

E. Frequency Estimation

In a final experiment, we compared the performance of DCS-AMP against techniques designed to solve the problem of subspace identification and tracking from partial observations (SITPO) [45], [46], which bears similarities to the dynamic CS problem. In subspace identification, the goal is to learn the low-dimensional subspace occupied by multi-timestep data measured in a high ambient dimension, while in subspace tracking, the goal is to track that subspace as it evolves over time. In the partial observation

setting, the high-dimensional observations are sub-sampled using a mask that varies with time. The dynamic CS problem can be viewed as a special case of SITPO, wherein the time-t subspace is spanned by a subset of the columns of an a priori known matrix $\boldsymbol{A}^{(t)}$. One problem that lies in the intersection of SITPO and dynamic CS is frequency tracking from partial time-domain observations.

For comparison purposes, we replicated the "direction of arrival analysis" experiment described in [46] where the observations at time t take the form

$$\boldsymbol{y}^{(t)} = \boldsymbol{\Phi}^{(t)}\boldsymbol{V}^{(t)}\boldsymbol{a}^{(t)} + \boldsymbol{e}^{(t)}, \quad t = 1, 2, \ldots, T \tag{12}$$

where $\boldsymbol{\Phi}^{(t)} \in \{0,1\}^{M\times N}$ is a selection matrix with non-zero column indices $\mathcal{Q}^{(t)} \subset \{1,\ldots,N\}$, $\boldsymbol{V}^{(t)} \in \mathbb{C}^{N\times K}$ is a Vandermonde matrix of sampled complex sinusoids, i.e.,

$$\boldsymbol{V}^{(t)} \triangleq [\boldsymbol{v}(\omega_1^{(t)}), \ldots, \boldsymbol{v}(\omega_K^{(t)})], \tag{13}$$

with $\boldsymbol{v}(\omega_k^{(t)}) \triangleq [1, e^{j2\pi\omega_k^{(t)}}, \ldots, e^{j2\pi\omega_k^{(t)}(N-1)}]^{\mathsf{T}}$ and $\omega_k^{(t)} \in [0,1)$. $\boldsymbol{a}^{(t)} \in \mathbb{R}^K$ is a vector of instantaneous amplitudes, and $\boldsymbol{e}^{(t)} \in \mathbb{R}^N$ is additive noise with i.i.d. $\mathcal{N}(0,\sigma_e^2)$ elements.[9] Here, $\{\boldsymbol{\Phi}^{(t)}\}_{t=1}^T$ is known, while $\{\boldsymbol{\omega}^{(t)}\}_{t=1}^T$ and $\{\boldsymbol{a}^{(t)}\}_{t=1}^T$ are unknown, and our goal is to estimate them. To assess performance, we report TNMSE in the estimation of the "complete" signal $\{\boldsymbol{V}^{(t)}\boldsymbol{a}^{(t)}\}_{t=1}^T$.

We compared DCS-AMP's performance against two online algorithms designed to solve the SITPO problem: GROUSE [45] and PETRELS [46]. Both GROUSE and PETRELS return time-varying subspace estimates, which were passed to an ESPRIT algorithm to generate time-varying frequency estimates (as in [46]). Finally, time-varying amplitude estimates were computed using least-squares. For DCS-AMP, we constructed $\boldsymbol{A}^{(t)}$ using a $2\times$ column-oversampled DFT matrix, keeping only those rows indexed by $\mathcal{Q}^{(t)}$. DCS-AMP was run in filtering mode for fair comparison with the "online" operation of GROUSE and PETRELS, with $I = 7$ inner AMP iterations.

The results of performing the experiment for three different problem configurations are presented in Table VII, with performance averaged over 100 independent realizations. All three algorithms were given the true value of K. In the first problem setup considered, we see that GROUSE operates the fastest, although its TNMSE performance is noticeably inferior to that of both PETRELS and DCS-AMP, which provide similar TNMSE performance and complexity. In the second problem setup, we reduce the number of measurements, M, from 30 to 10, leaving all other settings fixed. In this regime, both GROUSE and PETRELS are unable to accurately estimate $\{\omega_k^{(t)}\}$, and consequently fail to accurately recover $\boldsymbol{V}^{(t)}\boldsymbol{a}^{(t)}$,

[9]Code for replicating the experiment provided by the authors of [46]. Specific choices regarding $\{\omega_k^{(t)}\}$ and $\{\boldsymbol{a}^{(t)}\}$ were made by the authors of [46], and can be found in the code.

		Problem Setup		
		$N = 256,\ M = 30,\ K = 5$	$N = 256,\ M = 10,\ K = 5$	$N = 1024,\ M = 120,\ K = 20$
Algorithm	GROUSE	-4.52 (dB) | **6.78 (s)**	2.02 (dB) | **6.68 (s)**	-4.51 (dB) | 173.89 (s)
	PETRELS	**-15.62** (dB) | 29.51 (s)	0.50 (dB) | 14.93 (s)	-7.98 (dB) | 381.10 (s)
	DCS-AMP	-15.46 (dB) | 34.49 (s)	**-10.85 (dB)** | 28.42 (s)	**-12.79 (dB)** | **138.07 (s)**

TABLE VII: Average performance on synthetic frequency estimation experiment (TNMSE (dB) | Runtime (s)) of GROUSE, PETRELS, and DCS-AMP. In all cases, $T = 4000$, $\sigma_e^2 = 10^{-6}$.

in contrast to DCS-AMP. In the third problem setup, we increased the problem dimensions from the first problem setup by a factor of 4 to understand how the complexity of each approach scales with problem size. Interestingly, DCS-AMP, which was the slowest at smaller problem dimensions, becomes the fastest (and most accurate) in the higher-dimensional setting, scaling much better than either GROUSE or PETRELS.

VII. Conclusion

In this work we proposed DCS-AMP, a novel approach to dynamic CS. Our technique merges ideas from the fields of belief propagation and switched linear dynamical systems, together with a computationally efficient inference method known as AMP. Moreover, we proposed an EM approach that learns all model parameters automatically from the data. In numerical experiments on synthetic data, DCS-AMP performed within 3 dB of the support-aware Kalman smoother bound across the sparsity-undersampling plane. Repeating the dynamic MRI experiment from [44], DCS-AMP slightly outperformed Modified-CS in MSE, but required less than 10 seconds to run, in comparison to more than 7 hours for Modified-CS. For the compressive sensing of audio, we demonstrated significant gains from the exploitation of temporal structure and Gaussian-mixture learning of the signal prior. Lastly, we found that DCS-AMP can outperform recent approaches to Subspace Identification and Tracking from Partial Observations (SITPO) when the underlying problem can be well-represented through a dynamic CS model.

References

[1] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Asilomar Conf. on Signals, Systems and Computers 2010*, (Pacific Grove, CA), Nov. 2010.

[2] N. Vaswani, "Kalman filtered compressed sensing," in *IEEE Int'l Conf. on Image Processing (ICIP) 2008*, pp. 893 –896, 12-15 2008.

[3] M. Salman Asif, D. Reddy, P. Boufounos, and A. Veeraraghavan, "Streaming compressive sensing for high-speed periodic videos," in *Int'l Conf. on Image Processing (ICIP) 2010*, (Hong Kong), Sept. 2010.

[4] W. Li and J. C. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE J. Oceanic Engr.*, vol. 32, pp. 927–939, Oct. 2007.

[5] D. Angelosante, G. B. Giannakis, and E. Grossi, "Compressed sensing of time-varying signals," in *Int'l Conf. on Digital Signal Processing 2009*, pp. 1–8, July 2009.

[6] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, pp. 4595–4607, Sept. 2010.

[7] D. Angelosante, S. Roumeliotis, and G. Giannakis, "Lasso-Kalman smoother for tracking sparse signals," in *Asilomar Conf. on Signals, Systems and Computers 2009*, (Pacific Grove, CA), pp. 181 – 185, Nov. 2009.

[8] W. Dai, D. Sejdinović, and O. Milenkovic, "Gaussian dynamic compressive sensing," in *Int'l Conf. on Sampling Theory and Appl. (SampTA)*, (Singapore), May 2011.

[9] D. Sejdinović, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *48th Allerton Conf. Comm., Control, & Comp.*, (Urbana, IL), pp. 1730–1736, Nov. 2010.

[10] B. Shahrasbi, A. Talari, and N. Rahnavard, "TC-CSBP: Compressive sensing for time-correlated data based on belief propagation," in *Conf. on Inform. Sci. and Syst. (CISS)*, (Baltimore, MD), pp. 1–6, Mar. 2011.

[11] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2nd ed., 2004.

[12] B. J. Frey and D. J. C. MacKay, "A revolution: Belief propagation in graphs with cycles," *Neural Information Processing Systems (NIPS)*, vol. 10, pp. 479–485, 1998.

[13] 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.

[14] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," in *Proceedings of the National Academy of Sciences*, vol. 106, pp. 18914–18919, Nov. 2009.

[15] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Conf. on Information Sciences and Systems (CISS)*, pp. 1 –6, Mar. 2010.

[16] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, pp. 764–785, Feb. 2011.

[17] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process. [to appear]*, 2012.

[18] P. Schniter, "A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels," *IEEE J. Select. Topics in Signal Process.*, vol. 5, pp. 1462–1474, Dec. 2011.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

[20] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[22] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, pp. 2173–2200, Oct. 2001.

[23] S. C. Tatikonda and M. I. Jordan, *Loopy belief propagation and Gibbs measures*, pp. 493–500. Proc. 18th Conf. Uncertainty in Artificial Intelligence (UAI), San Mateo, CA: Morgan Kaufmann, 2002.

[24] T. Heskes, "On the uniqueness of belief propagation fixed points," *Neural Comput.*, vol. 16, no. 11, pp. 2379–2413, 2004.

[25] A. T. Ihler, J. W. Fisher III, and A. S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *J. Mach. Learn. Res.*, vol. 6, pp. 905–936, 2005.

[26] R. J. McEliece, D. J. C. MacKay, and J. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Select. Areas Comm.*, vol. 16, pp. 140–152, Feb. 1998.

[27] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int'l. J. Comp. Vision*, vol. 40, pp. 25–47, Oct. 2000.

[28] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proc. of Information Theory Workshop*, Jan. 2010.

[29] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem." arXiv:1111.5272 [cs.IT], Nov. 2011.

[30] J. P. Vila and P. Schniter, "Expectation-Maximization Gaussian-mixture approximate message passing," in *Proc. Conf. on Information Sciences and Systems*, (Princeton, NJ), Mar. 2012.

[31] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: Informed scheduling for asynchronous message passing," in *Proc. 22nd Conf. Uncertainty Artificial Intelligence (UAI)*, 2006.

[32] O. Zoeter and T. Heskes, "Change point problems in linear dynamical systems," *J. Mach. Learn. Res.*, vol. 6, pp. 1999–2026, Dec. 2005.

[33] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Auto. Control*, vol. 17, pp. 439–448, Aug. 1972.

[34] D. Barber and A. T. Cemgil, "Graphical models for time series," *IEEE Signal Process. Mag.*, vol. 27, pp. 18–28, Nov. 2010.

[35] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, pp. 47–60, Nov. 1996.

[36] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[37] J. Ziniel, *Message Passing Approaches to Compressive Inference Under Structured Signal Priors*. PhD thesis, The Ohio State University. In preparation.

[38] D. L. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Phil. Trans. R. Soc. A*, vol. 367, pp. 4273–4293, 2009.

[39] R. L. Eubank, *A Kalman Filter Primer*. Boca Raton, FL: Chapman & Hall/CRC, 2006.

[40] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[41] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[42] E. van den Berg and M. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Scientific Comp.*, vol. 31, pp. 890–912, Nov. 2008.

[43] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using Sparse Bayesian Learning," *IEEE J. Selected Topics Signal Process.*, vol. 5, pp. 912–926, Sept. 2011.

[44] W. Lu and N. Vaswani, "Modified compressive sensing for real-time dynamic MR imaging," in *Image Processing (ICIP), 2009 IEEE Int'l Conf. on*, pp. 3045 –3048, Nov. 2009.

[45] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Allerton Conf. on Comm., Control, and Comp.*, pp. 704 –711, Oct. 2010.

[46] Y. Chi, Y. Eldar, and R. Calderbank, "PETRELS: Subspace estimation and tracking from partial observations," in *Int'l Conf. Acoustics, Speech, & Signal Process. (ICASSP)*, (Kyoto, Japan), Mar. 2012.