

# From Denoising to Compressed Sensing

Christopher A. Metzler, Arian Maleki, and Richard G. Baraniuk

**Abstract**—A denoising algorithm seeks to remove noise, errors, or perturbations from a signal. Extensive research has been devoted to this arena over the last several decades, and as a result, today's denoisers can effectively remove large amounts of additive white Gaussian noise. A compressed sensing (CS) reconstruction algorithm seeks to recover a structured signal acquired using a small number of randomized measurements. Typical CS reconstruction algorithms can be cast as iteratively estimating a signal from a perturbed observation. This paper answers a natural question: How can one effectively employ a generic denoiser in a CS reconstruction algorithm? In response, we develop an extension of the approximate message passing (AMP) framework, called Denoising-based AMP (D-AMP), that can integrate a wide class of denoisers within its iterations. We demonstrate that, when used with a high performance denoiser for natural images, D-AMP offers state-of-the-art CS recovery performance while operating tens of times faster than competing methods. We explain the exceptional performance of D-AMP by analyzing some of its theoretical features. A key element in D-AMP is the use of an appropriate Onsager correction term in its iterations, which coerces the signal perturbation at each iteration to be very close to the white Gaussian noise that denoisers are typically designed to remove.

**Index Terms**—Compressed Sensing, Denoiser, Approximate Message Passing, Onsager Correction

## I. INTRODUCTION

### A. Compressed sensing

The fundamental challenge faced by a compressed sensing (CS) reconstruction algorithm is to reconstruct a high-dimensional signal from a small number of measurements. The process of taking compressive measurements can be thought of as a linear mapping of a length  $n$  signal vector  $x_o$  to a length  $m$ ,  $m \ll n$ , measurement vector  $y$ . Because this process is linear, it can be modeled by a measurement matrix  $\Phi \in \mathbb{C}^{m \times n}$ . The matrix  $\Phi$  can take on a variety of physical interpretations: In a compressively sampled MRI,  $\Phi$  might be sampled rows of an  $n \times n$  Fourier matrix [1], [2]. In a single pixel camera,  $\Phi$  might be a sequence of 1s and 0s representing the modulation of a micromirror array [3].

Oftentimes a signal  $x_o$  is sparse (or approximately sparse) in some transform domain, i.e.,  $x_o = \Psi u$  with sparse  $u$ , where  $\Psi$  represents the inverse transform matrix. In this case we lump the measurement and transformation into a single measurement matrix  $\mathbf{A} = \Phi\Psi$ . When a sparsifying basis is

not used  $\mathbf{A} = \Phi$ . Future references to the measurement matrix refer to  $\mathbf{A}$ .

The compressed sensing reconstruction problem is to determine which signal  $x_o$  produced  $y$  when sampled according to  $y = \mathbf{A}x_o + w$  where  $w$  represents measurement noise. Because  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $m \ll n$ , the problem is severely under-determined.<sup>1</sup> Therefore to recover  $x_o$  one first assumes that  $x_o$  possesses a certain structure and then searches, among all the vectors  $x$  that satisfy  $y \approx \mathbf{A}x$ , for one that also exhibits the given structure. In case of sparse  $x_o$ , one recovery method is to solve the convex problem

$$\underset{x}{\text{minimize}} \|x\|_1 \quad \text{subject to} \quad \|y - \mathbf{A}x\|_2^2 \leq \lambda, \quad (1)$$

which is known formally as basis pursuit denoising (BPDN). It was first shown in [4], [5] that if  $x_o$  is sufficiently sparse and  $\mathbf{A}$  satisfies certain properties, then (1) can accurately recover  $x_o$ .

The initial work in CS solved (1) using convex programming methods. However, when dealing with large signals, such as images, these convex programs are extremely computationally demanding. Therefore, lower cost iterative algorithms were developed; including matching pursuit [6], orthogonal matching pursuit [7], iterative hard-thresholding [8], compressive sampling matching pursuit [9], approximate message passing [10], and iterative soft-thresholding [11]–[16], to name just a few. See [17], [18] for a complete set of references.

Iterative thresholding (IT) algorithms generally take the form

$$\begin{aligned} x^{t+1} &= \eta_\tau(\mathbf{A}^* z^t + x^t), \\ z^t &= y - \mathbf{A}x^t, \end{aligned} \quad (2)$$

where  $\eta_\tau(y)$  is a shrinkage/thresholding non-linearity,  $x^t$  is the estimate of  $x_o$  at iteration  $t$ , and  $z^t$  denotes the estimate of the residual  $y - \mathbf{A}x_o$  at iteration  $t$ . When  $\eta_\tau(y) = (|y| - \tau)_+ \text{sign}(y)$  the algorithm is known as iterative soft-thresholding (IST).

AMP extends iterative soft-thresholding by adding an extra term to the residual known as the *Onsager correction term*:

$$\begin{aligned} x^{t+1} &= \eta_\tau(\mathbf{A}^* z^t + x^t), \\ z^t &= y - \mathbf{A}x^t + \frac{1}{\delta} z^{t-1} \langle \eta'_\tau(\mathbf{A}^* z^{t-1} + x^{t-1}) \rangle. \end{aligned} \quad (3)$$

Here,  $\delta = m/n$  is a measure of the under-determinacy of the problem,  $\langle \cdot \rangle$  denotes the average of a vector, and  $\frac{1}{\delta} \langle \eta'_\tau(\mathbf{A}^* z^{t-1} + x^{t-1}) \rangle$ , where  $\eta'_\tau$  represents the derivative of  $\eta_\tau$ , is the Onsager correction term. The role of this term is illustrated in Figure 1. This figure compares the QQplot<sup>2</sup> of  $x^t + \mathbf{A}^* z^t - x_o$  for IST and AMP. We call this quantity

<sup>1</sup>Note that for notational simplicity in our current derivations and algorithms we restrict  $\mathbf{A}$  to be in  $\mathbb{R}^{m \times n}$ . However, an extension to  $\mathbb{C}^{m \times n}$  is also possible.

<sup>2</sup>A QQplot is a visual inspection tool for checking the Gaussianity of the data. In a QQplot, deviation from a straight line is an evidence of non-Gaussianity.

C. Metzler and R. Baraniuk are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77023 USA (e-mail: chris.metzler@rice.edu and richb@rice.edu).

A. Maleki is with the Department of Statistics, Columbia University, New York, NY 10023 USA (e-mail: arian@stat.columbia.edu).

The work of C. Metzler supported by the NSF GRF Program and the DoD NDSEG Program. The work of C. Metzler and R. Baraniuk was supported in part and by the grants NSF CCF1527501, AFOSR FA9550-14-1-0088, ARO W911NF-15-1-0316, ONR N00014-12-1-0579, and DoD HM04761510007. The work of A. Maleki was supported by the grant NSF CCF-1420328.

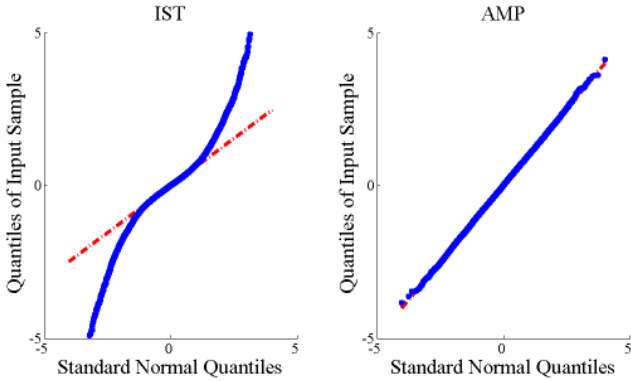


Fig. 1. QQplot comparing the distributions of the effective noise of the IST and AMP algorithms at iteration 5 while reconstructing a 50% sampled Barbara test image. Notice the heavy tailed distribution of IST. AMP remains Gaussian because of the Onsager correction term.

the *effective noise* of the algorithm at iteration  $t$ . As is clear from the figure, the QQplot of the effective noise in AMP is a straight line. This means that the noise is approximately Gaussian. This important feature enables the accurate analysis of the algorithm [10], [19], the optimal tuning of the parameters [20], and leads to the linear convergence of  $x^t$  to the final solution [21]. We will employ this important feature of AMP in our work as well.

### B. Main contributions

A sparsity model is accurate for many signals and has been the focus of the majority of CS research. Unfortunately, sparsity-based methods are less appropriate for many imaging applications. The reason for this failure is that natural images do not have an exactly sparse representation in any known basis (DCT, wavelet, curvelet, etc.). Figure 2 shows the wavelet coefficients of the classic signal processing image Barbara. The majority of the coefficients are non-zero and many are far from zero. As a result, algorithms that seek only wavelet-sparsity fail to recover the signal.

In response to this failure, researchers have considered more elaborate structures for CS recovery. These include minimal total variation [1], [22], block sparsity [23], wavelet tree sparsity [24], [25], hidden Markov mixture models [26]–[28], non-local self-similarity [29]–[31], and simple representations in adaptive bases [32], [33]. Many of these approaches have led to significant improvements in imaging tasks.

In this paper, we take a complementary approach to enhancing the performance of CS recovery of non-sparse signals [34]. Rather than focusing on developing new signal models, we demonstrate how the existing rich literature on *signal denoising* can be leveraged for enhanced CS recovery.<sup>3</sup> The idea is simple: Signal denoising algorithms (whether based on an explicit or implicit model) have been developed and

<sup>3</sup>In this paper, denoising refers to any algorithm that receives  $x_o + \sigma z$ , where  $\sigma z \sim N(0, \sigma^2 I)$  denotes the noise, as its input and returns an estimate of  $x_o$  as its output. Refer to Sections III-B and VII-A for more information on denoisers.

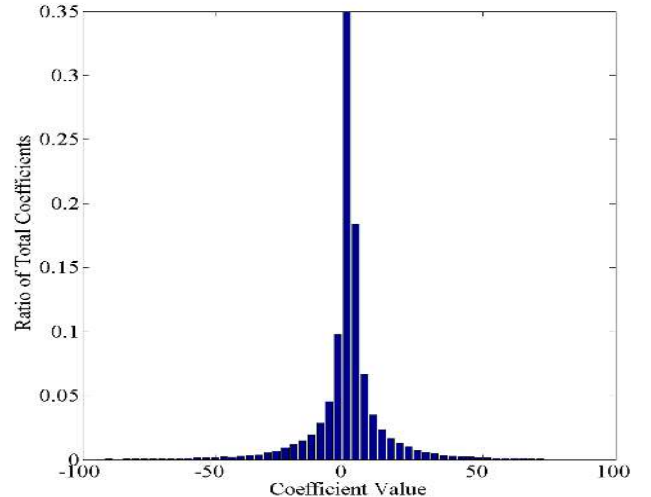


Fig. 2. Histogram of the Daubechies 4 wavelet coefficients of the Barbara test image. Notice the non-sparse distribution of the coefficients. Sparsity-based compressed sensing algorithms fail because of this distribution.

optimized for decades. Hence, any CS recovery scheme that employs such denoising algorithms should be able to capture complicated structures that have heretofore not been captured by existing CS recovery schemes.

The approximate message passing algorithm (AMP) [21], [35] presents a natural way to employ denoising algorithms for CS recovery. We call the AMP that employs denoiser  $D$  D-AMP. D-AMP assumes that  $x_o$  belongs to a class of signals  $C \subset \mathbb{R}^n$ , such as the class of natural images of a certain size, for which a family of denoisers  $\{D_\sigma : \sigma > 0\}$  exists. Each denoiser  $D_\sigma$  can be applied to  $x_o + \sigma z$  with  $z \sim N(0, I)$  and will return an estimate of  $x_o$  that is hopefully closer to  $x_o$  than  $x_o + \sigma z$ . These denoisers may employ simple structures such as sparsity or much more complicated structures, which we will discuss in Section VII-A. In this paper we treat each denoiser as a black box; it receives a signal plus Gaussian noise and returns an estimate of  $x_o$ . Hence, we do not assume any knowledge of the signal structure/information the denoising algorithm is employing to achieve its goal. This makes our derivations applicable to a wide variety of signal classes and a wide variety of denoisers.

D-AMP has several advantages over existing CS recovery algorithms: (i) It can be easily applied to many different signal classes. (ii) It outperforms existing algorithms and is extremely robust to measurement noise (our simulation results are summarized in Section VII). (iii) It comes with an analysis framework that not only characterizes its fundamental limits, but also suggests how we can best use the framework in practice.

D-AMP employs a denoiser in the following iteration:

$$\begin{aligned}
 x^{t+1} &= D_{\hat{\sigma}^t}(x^t + \mathbf{A}^* z^t), \\
 z^t &= y - \mathbf{A}x^t + z^{t-1} \text{div} D_{\hat{\sigma}^{t-1}}(x^{t-1} + \mathbf{A}^* z^{t-1})/m, \\
 (\hat{\sigma}^t)^2 &= \frac{\|z^t\|_2^2}{m}.
 \end{aligned} \tag{4}$$

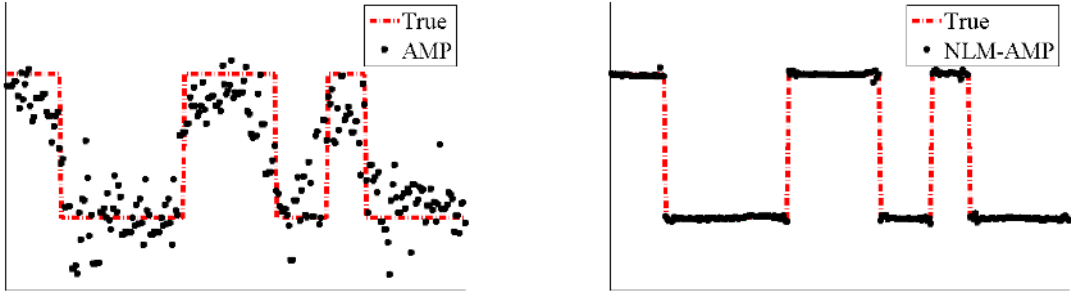


Fig. 3. Reconstructions of a piecewise constant signal that was sampled at a rate of  $\delta = 1/3$ . Notice that NLM-AMP successfully reconstructs the piecewise constant signal whereas AMP, which is based on wavelet thresholding, does not.

Here,  $x^t$  is the estimate of  $x_o$  at iteration  $t$  and  $z^t$  is an estimate of the residual. As we will show later,  $x^t + \mathbf{A}^*z^t$  can be written as  $x_o + v^t$ , where  $v^t$  can be considered as i.i.d. Gaussian noise.<sup>4</sup>  $\hat{\sigma}^t$  is an estimate of the standard deviation of that noise.  $\text{div}D_{\hat{\sigma}^{t-1}}$  denotes the divergence of the denoiser.<sup>5</sup> The term  $z^{t-1}\text{div}D_{\hat{\sigma}^{t-1}}(x^{t-1} + \mathbf{A}^*z^{t-1})/m$  is the Onsager correction term. We will show later that this term has a major impact on the performance of the algorithm. The explicit calculation of this term is not always straightforward: many popular denoisers do not have explicit formulations. However, we will show that it may be approximately calculated without requiring the explicit form of the denoiser.

D-AMP applies an existing denoising algorithm to vectors that are generated from compressive measurements. The intuition is that at every iteration D-AMP obtains a better estimate of  $x_o$  and that this sequence of estimates eventually converges to  $x_o$ .

To predict the performance of D-AMP, we will employ a novel state evolution framework to theoretically track the standard deviation of the noise,  $\hat{\sigma}_t$  at each iteration of D-AMP. Our framework extends and validates the state evolution framework proposed in [10], [21]. Through extensive simulations we show that in high-dimensional settings (for the subset of denoisers that we consider in this paper) our state evolution predicts the mean square error (MSE) of D-AMP accurately. Based on the state evolution we characterize the performance of D-AMP and connect the number of measurements D-AMP requires to the performance of the denoiser. We also employ the state evolution to address practical concerns such as the tuning of the parameters of denoisers and the sensitivity of the algorithm to measurement noise. Furthermore, we use the state evolution to explore the optimality of D-AMP. We postpone a detailed discussion to Section III.

<sup>4</sup>This conjecture has been validated empirically elsewhere [21], [35], [36] for simpler denoisers. For known  $\sigma_t$  the conjecture has been proven for scalar denoisers in [37]. By combining the proof of [37] with the proof technique developed in [38] we can prove the above conjecture for the scalar denoisers. Since scalar denoisers are not of our main concern in this paper, we do not include a proof here. We will present empirical evidence that it holds for many of the state-of-the-art image denoising algorithms.

<sup>5</sup>In the context of this work the divergence  $\text{div}D(x)$  is simply the sum of the partial derivatives with respect to each element of  $x$ , i.e.,  $\text{div}D(x) = \sum_{i=1}^n \frac{\partial D(x)}{\partial x_i}$ , where  $x_i$  is the  $i^{\text{th}}$  element of  $x$ .

Figure 3 compares the performance of the original AMP (which employs sparsity in the wavelet domain) with that of D-AMP based on the non-local means denoising algorithms [39], called NLM-AMP here. Since NLM is a better denoiser than wavelet thresholding for piecewise constant functions, NLM-AMP dramatically outperforms the original AMP. The details of our simulations are given in Section VII-D.

### C. Related work

1) *Approximate message passing and extensions:* In the last five years, message passing and approximate message passing algorithms have been the subject of extensive research in the field of compressed sensing [10], [19], [21], [28], [36]–[38], [40]–[53]. Most previously published papers consider a Bayesian framework in which a signal prior  $p_x$  is defined on the class of signals  $C$  to which  $x_o$  belongs. Message passing algorithms are then considered as heuristic approaches of calculating the posterior mean,  $\mathbb{E}(x_o | y, A)$ . Message passing has been simplified to approximate message passing (AMP) by employing the high dimensionality of the data [10]. The state evolution framework has been proposed as a way of analyzing the AMP algorithm [10]. The main difference between this line of work and our work is that we do not assume any signal prior on the signal space  $C$ . This distinction introduces a difference between the state evolution framework we develop in this paper and the ones that have been developed elsewhere. We will highlight the connection between these two different state evolutions in Section IV.

Note that in the development of D-AMP we are not concerned about whether the algorithm is approximating a posterior distribution for a certain prior or not. Nor are we concerned about whether or not the denoisers used within D-AMP's iterations are tied to any prior. Instead, we rely on only one important feature of AMP—that  $x^t + \mathbf{A}^*z^t - x_o$  behaves similar to i.i.d. Gaussian noise. Our analysis is based on this assumption. We validate this assumption with extensive simulations that are presented in Section VII-C.

Donoho et al. [35] also extended the AMP framework based upon the fact that  $x^t + \mathbf{A}^*z^t - x_o$  behaves similar to i.i.d. Gaussian noise. In their framework the denoiser  $D$  can be any scale-invariant function. There are several major differences between our work and theirs: (i) We do not impose

scale-invariance on the denoiser, because this assumption does not hold for many practical denoisers. (ii) We present a far broader validation of our method and state evolution: The empirical validation [35] presented is concerned with very specific simple denoisers and has remained at the level of maximin phase transition [21]. Likewise, the state evolution they employed in their empirical validation is based on the Bayesian framework described above and the validations are restricted to simple distributions. In this paper we consider a deterministic version of the state evolution and for the first time present evidence that such a state evolution can in fact predict the performance of D-AMP. Note that the evidence we present goes far beyond the match in the maximin phase transition. This development is important because the maximin framework employed in [35] is not useful in most practical applications that deal with naturally occurring signals. (iii) We present a signal-dependent parameter tuning strategy for AMP and show that our deterministic state evolution can cope with those situations as well. (iv) We show how practical denoisers whose explicit functional form is not given can be employed in AMP. (v) We investigate the optimality of D-AMP as a means to employ different denoisers in the AMP algorithm.

While writing this paper, we became aware of another relevant paper about extensions to the AMP algorithm [54]. In this work, the authors employ AMP with scalar denoisers that are better adapted to the statistics of natural images. By doing so, they have obtained a major improvement over existing algorithms. In this paper, we consider a much broader class of denoisers. We not only show how the AMP algorithm can be adapted to such denoisers; we also explore the theoretical properties of our recovery algorithms.

2) *Model-based CS imaging*: Many researchers have noticed the weakness of sparsity-based methods for imaging applications and have therefore explored the use of more complicated signal models. These models can be enforced explicitly, by constraining the solution space, or implicitly, by using penalty functionals to encourage solutions of a certain form.

Initially these model-based methods were restricted to simple concepts like minimal total variation [22] and block sparsity [23], but they have since been extended to structures such as wavelet trees [24], [25] and mixture models [26]–[28]. Furthermore, some researchers have employed more complicated signal models through non-local regularization [29]–[31] and the use of adaptive over-complete dictionaries [32], [33]. A non-local regularization method, NLR-CS [31], represents the current state-of-the-art in CS recovery. Through the use of denoisers, rather than explicit models or penalty functionals, our algorithm outperforms these methods on standard test images.

An additional reconstruction algorithm does not fit into any of the above categories but in many ways relates closely to our own. Egiazarian et al. developed a denoising-based CS recovery algorithm [55] that uses the same research group’s BM3D denoising algorithm [56] to impose a non-parametric model on the reconstructed signal. This method solves the CS problem when the measurement matrix is a subsampled

DFT matrix. The method iteratively adds noise to the missing part of the spectra and then applies BM3D to the result. In [31] it was shown that the BM3D-based algorithm performed considerably worse than NLR-CS. Therefore it is not tested here.

Finally we should emphasize another major difference between our work and other approaches designed for imaging applications. D-AMP comes with an accurate analysis that explains the behavior of the algorithm, its optimality properties, and its limitations. Such an accurate analysis does not exist for other methods.

#### D. Structure of the paper

The remainder of this paper is structured as follows: Section II introduces our D-AMP algorithm and some of its main features. Section III is devoted to the theoretical analysis of D-AMP and its optimality properties. Section IV establishes a connection between our state evolution and existing state evolutions. Section V explains two different approaches to calculating the Onsager correction term. Section VI explains how to smooth poorly behaved denoisers so that they can be used within our framework. Section VII summarizes our main simulation results: it provides evidence on the validity of our state evolution framework; it provides a detailed guideline on setting and tuning of different parameters of the algorithms; it compares the performance of our D-AMP algorithm with the state-of-the-art algorithms in compressive imaging.

## II. DENOISING-BASED APPROXIMATE MESSAGE PASSING

Consider a family of denoising algorithms  $D_\sigma$  for a class of signals  $\mathcal{C}$ . Our goal is to employ these denoisers to obtain a good estimate of  $x_o \in \mathcal{C}$  from  $y = \mathbf{A}x_o + w$ , where  $w \sim N(0, \sigma_w^2 I)$ . We start with the following approach that is inspired by the iterative hard-thresholding algorithm [8] and its extensions for block-based compressive imaging [57]–[59]. To better understand this approach, consider the noiseless setting in which  $y = \mathbf{A}x_o$ , and assume that the denoiser is a projection onto  $C$ . The affine subspace defined by  $y = \mathbf{A}x$  and the set  $C$  are illustrated in Figure 4. We assume that the point  $x_o$  is the unique point in the intersection of  $y = \mathbf{A}x$  and  $C$ .

We know that the solution lies in the affine subspace  $\{x|y = \mathbf{A}x\}$ . Therefore, starting from  $x^0 = 0$ , we move in the direction that is orthogonal to the subspace, i.e.,  $\mathbf{A}^*y$ .  $\mathbf{A}^*y$  is closer to the subspace however it is not necessarily close to  $C$ . Hence, we employ denoising (or projection in the figure) to obtain an estimate that satisfies the structure of our signal class  $C$ . After these two steps we obtain  $D(\mathbf{A}^*y)$ . As is also clear in the figure, by repeating these two steps, i.e., moving in the direction of the gradient and then projecting onto  $C$ , our estimate may eventually converge to the correct solution  $x_o$ . This leads us to the following iterative algorithm:<sup>6</sup>

$$\begin{aligned} x^{t+1} &= D_{\hat{\sigma}}(\mathbf{A}^*z^t + x^t), \\ z^t &= y - \mathbf{A}x^t. \end{aligned} \quad (5)$$

<sup>6</sup>Note that if  $D$  is a projection operator onto  $C$  and  $C$  is a convex set, then this algorithm is known as projected gradient descent and is known to converge to the correct answer  $x_o$ .

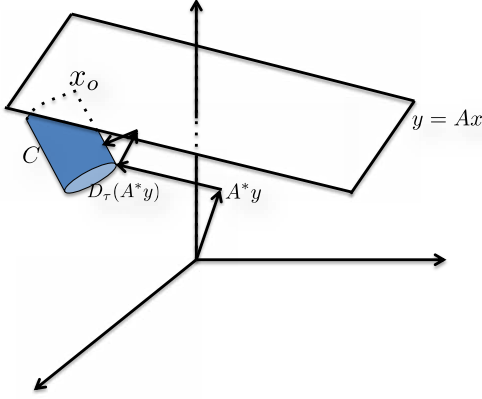


Fig. 4. Reconstruction behavior of denoising-based iterative thresholding algorithm.

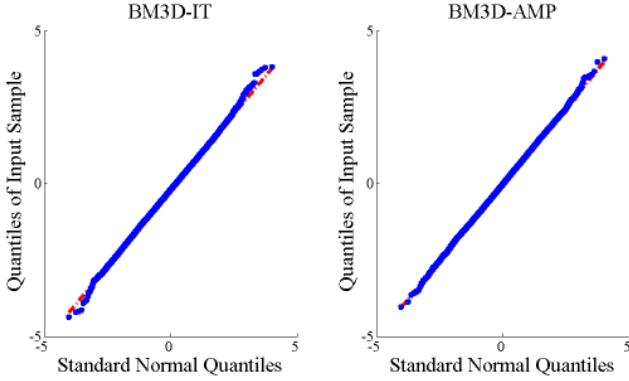


Fig. 5. QQplot comparing the distribution of the effective noise of D-IT and D-AMP at iteration 5 while reconstructing a 50% sampled Barbara test image. Notice the highly Gaussian distribution of D-AMP and the slight deviation from Gaussianity at the ends of the D-IT QQplot. This difference is due to D-AMP's use of an Onsager correction term. The denoiser that is employed in these simulations is BM3D. BM3D will be reviewed in Section VII-A.

For ease of notation, we have introduced the vector of estimated residual as  $z^t$ . We call this algorithm denoising-based iterative thresholding (D-IT). Note that if we replace  $D$  (that was assumed to be projection onto set  $C$  in Figure 4) with a denoising algorithm we implicitly assume that  $x^t + \mathbf{A}^*z^t$  can be modeled as  $x_o + v^t$ , where  $v^t \sim N(0, (\sigma^t)^2 I)$  and is independent of  $x_o$ . Hence, by applying a denoiser we obtain a signal that is closer to  $x_o$ . Unfortunately, as is shown in Figure 5(a) (we will show stronger evidence in Section III-C), this assumption is not true for D-IT. This is the same phenomenon that we observed in Section I-A for iterative soft-thresholding.

Our proposed solution to avoid the non-Gaussianity of the noise in the case of the iterative thresholding algorithms was to employ message passing/approximate message passing. Following the same path, we propose the following message

passing algorithm:

$$\begin{aligned} x_{\cdot \rightarrow a}^t &= D_{\hat{\sigma}^t} \left( \begin{bmatrix} \sum_{b \neq a} \mathbf{A}_{b1} z_{b \rightarrow 1}^t \\ \sum_{b \neq a} \mathbf{A}_{b2} z_{b \rightarrow 2}^t \\ \vdots \\ \sum_{b \neq a} \mathbf{A}_{bn} z_{b \rightarrow n}^t \end{bmatrix} \right), \\ z_{a \rightarrow i}^t &= y_a - \sum_{j \neq i} \mathbf{A}_{aj} x_{j \rightarrow a}^t. \end{aligned} \quad (6)$$

Here,  $x_{\cdot \rightarrow a}^t = [x_{1 \rightarrow a}^t, x_{2 \rightarrow a}^t, \dots, x_{n \rightarrow a}^t]^T$  provides an estimate of  $x_o$ .  $\hat{\sigma}^t$  denotes the standard deviation of the vector

$$v_{\cdot \rightarrow a}^t = \left( \begin{bmatrix} \sum_{b \neq a} \mathbf{A}_{b1} z_{b \rightarrow 1}^t \\ \sum_{b \neq a} \mathbf{A}_{b2} z_{b \rightarrow 2}^t \\ \vdots \\ \sum_{b \neq a} \mathbf{A}_{bn} z_{b \rightarrow n}^t \end{bmatrix} \right) - x_o.$$

Our empirical findings, summarized in Section VII-C, show that  $v_{\cdot \rightarrow a}^t$  closely resembles i.i.d. Gaussian noise in high-dimensional settings (both  $m$  and  $n$  are large). This result has been rigorously proved for a class of scalar denoisers and can also be proved for a class of block-wise denoisers [35], [36].<sup>7</sup>

Despite their advantage in avoiding the non-Gaussianity of the effective noise vector  $v$ , message passing algorithms have  $m$  (number of measurements) different estimates of  $x_o$ ; each  $x_{\cdot \rightarrow a}^t$  is an estimate of  $x_o$ . Similarly, they have  $n$  different estimates of the residual  $y - \mathbf{A}x_o$ . The update of all these messages is computationally demanding. Fortunately, if the problem is high dimensional, we can approximate a message passing algorithm's iterations and obtain the denoising-based approximate message passing algorithm (D-AMP):

$$\begin{aligned} x^{t+1} &= D_{\hat{\sigma}^t}(x^t + \mathbf{A}^*z^t), \\ z^t &= y - \mathbf{A}x^t + z^{t-1} \frac{\text{div} D_{\hat{\sigma}^{t-1}}(x^{t-1} + \mathbf{A}^*z^{t-1})}{m}. \end{aligned} \quad (7)$$

The only difference between D-AMP and D-IT is again in the Onsager correction term  $z^{t-1} \text{div} D_{\hat{\sigma}^{t-1}}(x^{t-1} + \mathbf{A}^*z^{t-1})/m$ . The derivation of D-AMP from the Denoising-based Message Passing (D-MP) algorithm is similar to the derivation of AMP from Message Passing (MP) which can be found in Chapter 5 of [21]. Similar to D-IT, D-AMP relies on the assumption that the effective noise  $v^t = x^t + \mathbf{A}^*z^t - x_o$  resembles i.i.d. Gaussian noise (independent of the signal  $x_o$ ) at every iteration. Our empirical findings confirm this assumption: Figure 5(b) displays the effective noise at iteration 5 of D-AMP with BM3D denoising, which will be briefly explained in Section VII-A (we call this algorithm BM3D-AMP). Notice the clearly Gaussian distribution. Based on this observation, and stronger evidence that we will provide in Section VII-C, we conjecture that  $v^t$  indeed behaves as additive white Gaussian noise for high dimensional problems. The proof of this property is left for future work. In this paper

<sup>7</sup>There are some subtle differences between our claim regarding the Gaussianity of  $v_{\cdot \rightarrow a}^t$  and the claims presented in other works. Our claim is made in a deterministic setting, while in existing works the Gaussianity claim is made in regards to stochastic settings. This point will be clarified in Section IV.

we not only provide strong empirical evidence to support our conjecture, but also explore its theoretical implications.

### III. THEORETICAL ANALYSIS OF D-AMP

The main objective of this section is to characterize the theoretical properties of the D-AMP framework. In this section (and also in our simulations) we start the algorithm with  $x^0 = 0$  and  $z^0 = y$ . Our analysis is under the high dimensional setting:  $n, m$  are very large while  $m/n = \delta < 1$  is a fixed number.  $\delta$  is called the under-determinacy of the system of equations.

#### A. Notation

We use boldfaced capital letters such as  $\mathbf{A}$  for matrices. For a matrix  $\mathbf{A}$ ;  $\mathbf{A}_i$ ,  $\mathbf{A}_{i,j}$ , and  $\mathbf{A}^*$  denote its  $i^{\text{th}}$  column,  $i,j^{\text{th}}$  element, and its transpose, respectively. Small letters such as  $x$  are reserved for vectors and scalars. For a vector  $x$ ,  $x_i$  and  $\|x\|_p$  denote the  $i^{\text{th}}$  element of the vector and its  $p$ -norm, respectively. The notations  $\mathbb{E}$  and  $\mathbb{P}$  denote the expected value of a random variable (or a random vector) and probability of an event, respectively. If the expected value is with respect to two random variables  $X$  and  $Z$ , then  $\mathbb{E}_X$  (or  $\mathbb{E}_Z$ ) denotes the expectation with respect to  $X$  (or  $Z$ ) and  $\mathbb{E}$  denotes the expected value with respect to both  $X$  and  $Z$ .

#### B. Denoiser properties

The role of a denoiser is to estimate a signal  $x_o$  belonging to a class of signals  $C \subset \mathbb{R}^n$  from noisy observations,  $x_o + \sigma\epsilon$ , where  $\epsilon \sim N(0, I)$ , and  $\sigma > 0$  denotes the standard deviation of the noise. We let  $D_\sigma$  denote a family of denoisers indexed with the standard deviation of the noise. At every value of  $\sigma$ ,  $D_\sigma$  takes  $x_o + \sigma\epsilon$  as the input and returns an estimate of  $x_o$ .

To analyze D-AMP, we require the denoiser family to be (near) proper, monotone, and Lipschitz continuous (proper and monotone are defined below). Because most denoisers easily satisfy these first two properties, and can be modified to satisfy the third (see Section VI), the requirements do not overly restrict our analysis.

**Definition 1.**  $D_\sigma$  is called a proper family of denoisers of level  $\kappa$  ( $\kappa \in (0, 1)$ ) for the class of signals  $C$  if

$$\sup_{x_o \in C} \frac{\mathbb{E}\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n} \leq \kappa\sigma^2, \quad (8)$$

for every  $\sigma > 0$ . Note that the expectation is with respect to  $\epsilon \sim N(0, I)$ .

To clarify the above definition, we consider the following examples:

**Example 1.** Let  $C$  denote a  $k$ -dimensional subspace of  $\mathbb{R}^n$  ( $k < n$ ). Also, let  $D_\sigma(y)$  be the projection of  $y$  onto subspace  $C$  denoted by  $P_C(y)$ . Then,

$$\frac{\mathbb{E}\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n} = \frac{k}{n}\sigma^2,$$

for every  $x_o \in C$  and every  $\sigma^2$ . Hence, this family of denoisers is proper of level  $k/n$ .

*Proof.* First note that since the projection onto a subspace is a linear operator and since  $P_C(x_o) = x_o$  we have

$$\mathbb{E}\|P_C(x_o + \sigma\epsilon) - x_o\|_2^2 = \mathbb{E}\|x_o + \sigma P_C(\epsilon) - x_o\|_2^2 = \sigma^2 \mathbb{E}\|P_C(\epsilon)\|_2^2.$$

Also note that since  $P_C^2 = P_C$ , all the eigenvalues of  $P_C$  are either zero or one. Furthermore, since the null space of  $P_C$  is  $n - k$  dimensional, the rank of  $P_C$  is  $k$ . Hence,  $P_C$  has  $k$  eigenvalues equal to 1 and the rest are zero. Hence  $\|P_C(\epsilon)\|_2^2$  follows a  $\chi^2$  distribution with  $k$  degrees of freedom and  $\mathbb{E}\|P_C(x_o + \sigma\epsilon) - x_o\|_2^2 = k\sigma^2$ .  $\square$

Next we consider a slightly more complicated example that has been popular in signal processing for the last twenty-five years. Let  $\Gamma_k$  denote the set of  $k$ -sparse vectors.

**Example 2.** Let  $\eta(y; \tau\sigma) = (|y| - \tau\sigma)_+ \text{sign}(y)$  denote the family of soft-thresholding denoisers. Then

$$\begin{aligned} & \sup_{x_o \in \Gamma_k} \frac{\mathbb{E}\|\eta(x_o + \sigma\epsilon; \tau\sigma) - x_o\|_2^2}{n} \\ &= \left[ \frac{(1 + \tau^2)k}{n} + \frac{n - k}{n} \mathbb{E}(\eta(\epsilon_1; \tau))^2 \right] \sigma^2. \end{aligned}$$

Similar results can be found in other papers including [19]. But since the proof is short and the result is slightly different from similar existing results, we mention the proof here.

*Proof.* For notational simplicity we assume that the first  $k$  coordinates of  $x_o$  are non-zero and the rest are equal to zero.

$$\begin{aligned} & \frac{\mathbb{E}\|\eta(x_o + \sigma\epsilon; \tau\sigma) - x_o\|_2^2}{n\sigma^2} \\ &= \frac{\sum_{i=1}^k \mathbb{E}(\eta(x_{o,i} + \sigma\epsilon_i; \tau\sigma) - x_{o,i})^2}{n\sigma^2} + \frac{n - k}{n\sigma^2} \mathbb{E}(\eta(\sigma\epsilon_n; \tau\sigma))^2 \\ &= \frac{\sum_{i=1}^k \mathbb{E}\left(\eta\left(\frac{x_{o,i}}{\sigma} + \epsilon_i; \tau\right) - \frac{x_{o,i}}{\sigma}\right)^2}{n} + \frac{n - k}{n} \mathbb{E}(\eta(\epsilon_n; \tau))^2. \end{aligned} \quad (9)$$

Note that  $\mathbb{E}\left(\eta\left(\frac{x_{o,i}}{\sigma} + \epsilon_i; \tau\right) - \frac{x_{o,i}}{\sigma}\right)^2$  is an increasing function of  $\frac{x_{o,i}}{\sigma}$  [60]. Therefore, it is straightforward to see that

$$\begin{aligned} & \mathbb{E}\left(\eta\left(\frac{x_{o,i}}{\sigma} + \epsilon_i; \tau\right) - \frac{x_{o,i}}{\sigma}\right)^2 \\ & \leq \lim_{x_{o,i} \rightarrow \infty} \mathbb{E}\left(\eta\left(\frac{x_{o,i}}{\sigma} + \epsilon_i; \tau\right) - \frac{x_{o,i}}{\sigma}\right)^2 = 1 + \tau^2 \end{aligned} \quad (10)$$

where the last step swaps the lim and  $\mathbb{E}$  (by the dominated convergence theorem). We obtain the desired result by combining (9) and (10).  $\square$

Note that the optimal threshold  $\tau$  to use within soft-thresholding depends on the sparsity  $k/n$  of the signal being denoised. One can optimize the parameter  $\tau$  for every value of  $k/n$  and obtain an optimized family of denoisers. Figure 6 displays the level  $\kappa$  of the optimized soft-thresholding in terms of  $k/n$ . Note that for sparse signals ( $k/n$  small) soft-thresholding is an effective denoiser and thus  $\kappa$  is small.

The previous denoisers both utilized prior knowledge about the structure of the signal (its dimensionality and its sparsity) in order to denoise  $x_o$ . When nothing is known about  $x_o$  a proper denoiser might be too much to ask for. For instance, consider the maximum likelihood estimator.



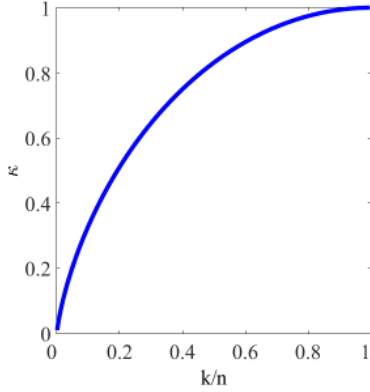


Fig. 6. The level  $\kappa$  of optimal soft-thresholding method as a function of normalized sparsity  $k/n$ . For sparse signals, soft-thresholding is a high performance denoiser.

**Example 3.** If  $D_\sigma(x_o + \sigma\epsilon)$  is the maximum likelihood estimate of  $x_o$  from  $x_o + \sigma\epsilon$ , then

$$\frac{\mathbb{E}\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n\sigma^2} = 1.$$

So, this family of denoisers are not proper of level  $\kappa$  for any  $\kappa < 1$ . The proof of this statement is straightforward and hence is skipped here. It has been shown that (Chapter 5 of [61]) for any denoiser  $\tilde{D}_\sigma$  we have

$$\sup_{x_o \in \mathbb{R}^n} \frac{\mathbb{E}\|\tilde{D}_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n\sigma^2} = 1.$$

In this example the class of signals we have considered is generic and hence the denoiser cannot employ any specific structure in  $x_o$ .

There are occasions when we want to deal with denoisers that are not proper because of an error/bias term that is independent of the noise level. To deal with scenarios such as these, we introduce the definition *near proper*.

**Definition 2.**  $D_\sigma$  is called a *near proper family of denoisers of levels  $\kappa$  ( $\kappa \in (0, 1)$ ) and  $B$  ( $B \in \mathbb{R}_+$ ) for the class of signals  $C$  if*

$$\sup_{x_o \in C} \frac{\mathbb{E}\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n} \leq \kappa\sigma^2 + B, \quad (11)$$

for every  $\sigma > 0$ . Note that the expectation is with respect to  $\epsilon \sim N(0, I)$ .

As in Definition 1, the constants  $\kappa$  and  $B$  determine the quality of the denoiser family. Better denoisers have smaller constants.

**Example 4.** Let  $C_p = \{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$  for some  $0 < p \leq 1$ .<sup>8</sup> For a fixed  $k$ , let  $D_\sigma$  denote a denoiser that, through oracle information, knows the indices of the  $k$  largest elements of  $x$  and projects the noisy observation  $x_o + \sigma\epsilon$  onto those coordinates. Then

$$\sup_{x_o \in C_p} \frac{\mathbb{E}\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n} \leq \frac{k}{n}\sigma^2 + \frac{k^{1-2/p}}{n(2/p-1)},$$

<sup>8</sup>For every  $0 < p \leq 1$ ,  $\|x\|_p^p = \sum_{i=1}^n |x_i|^p$ .

for every  $x_o \in C_p$  and every  $\sigma^2$ . Hence, this family of denoisers is near proper with  $\kappa = \frac{k}{n}$  and  $B = \frac{(k+1)^{1-2/p}}{n(2/p-1)}$ .

*Proof.* Let  $\Lambda$  denote the set of indices of the  $k$ -largest coefficients of  $x_o$ . For a vector  $x$ , define  $x_\Lambda$  in the following way:  $x_{\Lambda,i} = x_i$  if  $i \in \Lambda$  and otherwise  $x_{\Lambda,i} = 0$ . Note that  $x_{o,\Lambda}$  is the best  $k$ -term approximation of  $x_o$ . We have

$$\begin{aligned} \mathbb{E}\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 &= \mathbb{E}\|x_{o,\Lambda} + \sigma\epsilon_\Lambda - x_o\|_2^2 \\ &= \|x_{o,\Lambda} - x_o\|_2^2 + \sigma^2\mathbb{E}\|\epsilon_\Lambda\|_2^2. \end{aligned} \quad (12)$$

Following the same logic as used in Example 1 we see that

$$\mathbb{E}\|\epsilon_\Lambda\|_2^2 = k. \quad (13)$$

The term  $\|x_{o,\Lambda} - x_o\|_2^2$  above is simply the squared  $\ell_2$ -norm of the smallest  $n - k$  values of  $x_o$ . Below we obtain an upper bound for this quantity. Note that since  $x_o \in C_p$ , we have

$$\sum_{i=1}^n |x_{o,i}|^p \leq 1. \quad (14)$$

Let  $x_{o,(j)}$  denote the  $j^{\text{th}}$  largest element in absolute value of  $x_o$ . It is clear that  $|x_{o,(1)}| \geq |x_{o,(2)}|, \dots, |x_{o,(j-1)}| \geq |x_{o,(j)}|$ . Combining this fact with (14) we obtain  $j|x_{o,(j)}|^p \leq 1$ , which in turn implies  $|x_{o,(j)}| \leq j^{-1/p}$ . Returning to (12), we see that

$$\begin{aligned} \|x_{o,\Lambda} - x_o\|_2^2 &\leq \sum_{j=k+1}^n |x_{o,(j)}|^2 \leq \sum_{j=k+1}^n j^{-2/p} \\ &\leq \int_k^\infty \gamma^{-2/p} d\gamma = \frac{\gamma^{1-2/p}}{(1-2/p)} \Big|_k^\infty = \frac{(k)^{1-2/p}}{(2/p-1)}. \end{aligned} \quad (15)$$

Substituting (13) and (15) into (12) gives the desired result.  $\square$

In subsequent sections we assume our signal belongs to a class  $C$  for which we have a proper or near proper family of denoisers  $D_\sigma$ . The class and denoiser can be very general. For instance, we may assume  $C$  to be the class of natural images and  $D_\sigma$  to denote the BM3D algorithm<sup>9</sup> at different noise levels [56].

**Definition 3.** We call a denoiser *monotone* if for every  $x_o$  its risk function

$$R(\sigma^2, x_o) = \frac{\mathbb{E}(\|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2)}{n},$$

is a non-decreasing function of  $\sigma^2$ .

We make a few remarks regarding monotone denoisers.

**Remark 1.** *Monotonicity is a natural property to expect from denoisers. Many standard denoisers such as soft-thresholding and group soft-thresholding are monotone if we optimize over the threshold parameter. See Lemma 4.4 in [20] for more information.*

**Remark 2.** *If a family of denoisers  $D_\sigma$  is not monotone, then it is straightforward to construct a new denoiser that*

<sup>9</sup>We will review this algorithm briefly in Section VII-A

outperforms  $D_\sigma$ . Here is a simple proof. Suppose that for  $\sigma_1 < \sigma_2$  we have

$$R(\sigma_1^2, x_o) > R(\sigma_2^2, x_o).$$

Then construct a new denoiser for noise level  $\sigma_1$  in the following way:

$$\tilde{D}_{\sigma_1}(y) = \mathbb{E}_{\tilde{\epsilon}} D_{\sigma_2} \left( y + \sqrt{\sigma_2^2 - \sigma_1^2} \tilde{\epsilon} \right),$$

where  $\tilde{\epsilon} \sim N(0, I)$  is independent of  $y$  and  $\mathbb{E}_{\tilde{\epsilon}}(y + \sqrt{\sigma_2^2 - \sigma_1^2} \tilde{\epsilon})$  denotes the expected value with respect to  $\tilde{\epsilon}$ . Let  $\tilde{\sigma}_2 = \sqrt{\sigma_2^2 - \sigma_1^2}$ . A simple application of Jensen's inequality shows that

$$\begin{aligned} & \frac{\mathbb{E}_{\epsilon}(\|\tilde{D}_{\sigma_1}(x_o + \sigma_1 \epsilon) - x_o\|_2^2)}{n} \\ &= \frac{\mathbb{E}_{\epsilon}(\|\mathbb{E}_{\tilde{\epsilon}} D_{\sigma_2}(x_o + \sigma_1 \epsilon + \tilde{\sigma}_2 \tilde{\epsilon}) - x_o\|_2^2)}{n} \\ &\leq \frac{\mathbb{E}_{\epsilon, \tilde{\epsilon}}(\|D_{\sigma_2}(x_o + \sigma_1 \epsilon + \tilde{\sigma}_2 \tilde{\epsilon}) - x_o\|_2^2)}{n}. \end{aligned}$$

Note that since  $\tilde{\epsilon}$  and  $\epsilon$  are independent  $\frac{\mathbb{E}_{\epsilon, \tilde{\epsilon}}(\|D_{\sigma_2}(x_o + \sigma_1 \epsilon + \tilde{\sigma}_2 \tilde{\epsilon}) - x_o\|_2^2)}{n} = R(\sigma_2^2, x_o)$ . Therefore,  $\tilde{D}$  improves  $D$  and does not violate the monotone property. Therefore, as is clear from this statement, non-monotone denoisers are not desirable in general since we can easily improve them.

In the rest of the paper we consider only monotone denoisers.

### C. State evolution

A key ingredient in our analysis of D-AMP is the state evolution; a series of equations that predict the intermediate MSE of AMP algorithms at each iteration. Here we introduce a new ‘‘deterministic’’ state-evolution to predict the performance of D-AMP. Starting from  $\theta^0 = \frac{\|x_o\|_2^2}{n}$  the state evolution generates a sequence of numbers through the following iterations:

$$\theta^{t+1}(x_o, \delta, \sigma_w^2) = \frac{1}{n} \mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t \epsilon) - x_o\|_2^2, \quad (16)$$

where  $(\sigma^t)^2 = \frac{\theta^t}{\delta}(x_o, \delta, \sigma_w^2) + \sigma_w^2$  and the expectation is with respect to  $\epsilon \sim N(0, I)$ . Note that our notation  $\theta^{t+1}(x_o, \delta, \sigma_w^2)$  is set to emphasize that  $\theta^t$  may depend on the signal  $x_o$ , the under-determinacy  $\delta$ , and the measurement noise. Consider the iterations of D-AMP and let  $x^t$  denote its estimate at iteration  $t$ . Our empirical findings show that the MSE of D-AMP is predicted accurately by the state evolution. We formally state our finding.

**Finding 1.** *If the D-AMP algorithm starts from  $x^0 = 0$ , then for large values of  $m$  and  $n$ , state evolution predicts the mean square error of D-AMP, i.e.,*

$$\theta^t(x_o, \delta, \sigma_w^2) \approx \frac{1}{n} \|x^t - x_o\|_2^2.$$

Based on extensive simulations, we believe that this finding is true if the following properties are satisfied: (i) The elements of the matrix  $A$  are i.i.d. Gaussian (or subGaussian) with mean zero and standard deviation  $1/m$ . (ii) The noise  $w$  is also i.i.d.

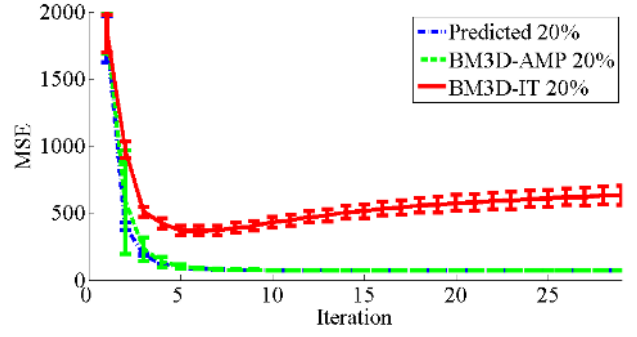


Fig. 7. The MSE of the intermediate estimate versus the iteration count for BM3D-AMP and BM3D-IT alongside their predicted state evolution. Notice that BM3D-AMP is well predicted by the state evolution whereas BM3D-IT is not.

Gaussian. (iii) The denoiser  $D$  is Lipschitz continuous.<sup>10</sup> In all our simulations the elements of  $A$  are i.i.d. Gaussian. The same is true for the elements of  $w$ .

Figure 7 compares the state evolution predictions of D-AMP (based on the BM3D denoising algorithm [56]) with the empirical performance of D-AMP and D-IT. As is clear from this figure, the state evolution is accurate for D-AMP but not for D-IT. We have checked the validity of the above finding for the following denoising algorithms: (i) BM3D [56], (ii) BLS-GSM [62], (iii) Non-local means [39], (iv) AMP with soft-wavelet-thresholding [10], [63]. We report some of our simulations on this phenomenon in Section VII-C. We have posted our code online<sup>11</sup> to enable other researchers to verify our findings in more general settings and explore the validity of this conjecture on a wider range of denoisers.

In the following sections we assume that the state evolution is accurate for D-AMP and derive some of the main features of D-AMP based on this assumption.

### D. Analysis of D-AMP in the absence of measurement noise

In this section we consider the noiseless setting  $\sigma_w^2 = 0$  and characterize the number of measurements D-AMP requires (under the validity of the state evolution framework) to recover the signal  $x_o$  exactly. We consider monotone denoisers, as defined in section III-B. Consider the state evolution equation under the noiseless setting  $\sigma_w^2 = 0$ :

$$\theta^{t+1}(x_o, \delta, 0) = \frac{1}{n} \mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t \epsilon) - x_o\|_2^2,$$

where  $(\sigma^t)^2 = \frac{\theta^t(x_o, \delta, 0)}{\delta}$ . Starting with  $\theta^0(x_o, \delta, 0) = \frac{\|x_o\|_2^2}{n}$ , depending on the value of  $\delta$  there are two conceivable scenarios for the state evolution equation:

<sup>10</sup>A denoiser is said to be  $L$ -Lipschitz continuous if for every  $x_1, x_2 \in C$  we have  $\|D(x_1) - D(x_2)\|_2^2 \leq L \|x_1 - x_2\|_2^2$ . Many advanced image denoisers have no closed form expression, thus it is very hard to verify whether or not they are Lipschitz continuous. That said, every advanced denoisers we tested was found to closely follow our state evolution equations (Finding 1), suggesting they are in fact Lipschitz. In Section VI we show examples in which Lipschitz continuity is violated and propose a simple approach for dealing with discontinuous denoisers.

<sup>11</sup><http://dsp.rice.edu/software/DAMP-toolbox>



- (i)  $\theta^t(x_o, \delta, 0) \rightarrow 0$  as  $t \rightarrow \infty$ .
- (ii)  $\theta^t(x_o, \delta, 0) \rightarrow 0$  as  $t \rightarrow \infty$ .

$\theta^t(x_o, \delta, 0) \rightarrow 0$  implies the success of D-AMP algorithm, while  $\theta^t(x_o, \delta, 0) \rightarrow 0$  implies its failure in recovering  $x_o$ . The main goal of this section is to study the success and failure regions.

**Lemma 1.** *For monotone denoisers, if for  $\delta_0$ ,  $\theta^t(x_o, \delta_0, 0) \rightarrow 0$ , then for any  $\delta > \delta_0$ ,  $\theta^t(x_o, \delta, 0) \rightarrow 0$  as well.*

*Proof.* Define  $(\sigma^t)^2 = \frac{\theta^t(x_o, \delta, \sigma_w^2)}{\delta}$ . Clearly, since  $\theta^t(x_o, \delta_0, \sigma_w^2) \rightarrow 0$  so does  $\sigma^t$ . Our first claim is that for every  $\sigma^2 < \frac{\|x_o\|_2^2}{n\delta_0} = (\sigma^0)^2$  (this is where D-AMP is initialized) we have

$$\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 < \sigma^2, \quad \forall \sigma^2 > 0.$$

Suppose that this is not true and define

$$\sigma_*^2 = \sup_{\sigma^2 \leq \frac{\|x_o\|_2^2}{n\delta_0}} \{\sigma^2 : \frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \geq \sigma^2\}.$$

We claim that if  $\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma^0\epsilon) - x_o\|_2^2 < (\sigma^0)^2$ , then  $\sigma^t \rightarrow \sigma_*$  as  $t \rightarrow \infty$ . First, it is straightforward to see that  $\frac{1}{n\delta_0} \mathbb{E} \|D_{\sigma_*}(x_o + \sigma_*\epsilon) - x_o\|_2^2 = \sigma_*^2$ . For  $\sigma > \sigma_*$  we know that

$$\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 < \sigma^2.$$

By using the monotonicity of the denoiser we have for every  $\sigma \geq \sigma_*$

$$\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \geq \frac{1}{n\delta_0} \mathbb{E} \|D_{\sigma_*}(x_o + \sigma_*\epsilon) - x_o\|_2^2 = \sigma_*^2.$$

This (through simple induction) implies that for every  $t$ ,

$$(\sigma^t)^2 \geq (\sigma_*)^2.$$

Furthermore according to the definition of  $\sigma_*^2$  and the fact that  $\sigma^t > \sigma_*$ , we have

$$(\sigma^{t+1})^2 = \frac{1}{n\delta_0} \mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t\epsilon) - x_o\|_2^2 < (\sigma^t)^2.$$

Therefore,  $\sigma^{t+1}$  is a decreasing sequence with lower bound  $\sigma_*$ . Hence,  $\sigma^t$  converges to  $\sigma^\infty \geq \sigma_*$ . The last step is to show that  $\sigma^\infty = \sigma_*$ . If this is not the case, then  $\sigma^\infty > \sigma_*$ . But according to the definition of  $\sigma_*$  and the supposition that  $\sigma^\infty > \sigma_*$ , we have

$$\frac{1}{n\delta_0} \mathbb{E} \|D_{\sigma^\infty}(x_o + \sigma^\infty\epsilon) - x_o\|_2^2 < (\sigma^\infty)^2,$$

which is a contradiction to  $\sigma^\infty$  being a fixed point. Hence  $\sigma^\infty = \sigma_*$ . Since  $\sigma^\infty = 0$ , we conclude that  $\sigma_* = 0$  and we have

$$\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 < \sigma^2, \quad \forall \sigma^2 > 0.$$

Since,  $\delta > \delta_0$  we can conclude that

$$\frac{1}{n\delta} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 < \sigma^2, \quad \forall \sigma^2 > 0.$$

Hence the only fixed point of this equation is also at zero and hence  $\theta^t(x_o, \delta, 0) \rightarrow 0$ . Note that all the above argu-

ment is based on the assumption that  $\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma^0\epsilon) - x_o\|_2^2 < (\sigma^0)^2$ . What if this assumption is violated? Using similar argument it is straightforward to show that if the  $\frac{1}{n\delta_0} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2$  has a fixed point above  $(\sigma^0)^2$ , then the algorithm converges to the closest fixed point above  $\sigma_0$ , which is a contradiction again. Also, if the algorithm does not have any fixed point above  $\sigma^0$ , then it will diverge to infinity, which is again a contradiction.  $\square$

Note that for very small values of  $\delta$ , it is straightforward to see that  $\theta^t(x_o, \delta, 0) \rightarrow 0$  as  $t \rightarrow \infty$ . If we combine this result with Lemma 1 we conclude the following simple result: For small values of  $\delta$  D-AMP fails in recovering  $x_o$ . As  $\delta$  increases, after a certain value of  $\delta$  D-AMP will successfully recover  $x_o$  from its undersampled measurements. Define

$$\delta^*(x_o) = \inf_{\delta \in (0,1)} \{\delta : \theta^t(x_o, \delta, 0) \rightarrow 0 \text{ as } t \rightarrow \infty\}.$$

$\delta^*(x_o)$  denotes the minimum number of measurements required for the successful recovery of  $x_o$ . Our goal is to characterize  $\delta^*(x_o)$  in terms of the performance (we will clarify what we mean by performance) of the denoising algorithm. However, since the number of measurements  $\delta^*(x_o)$  depends on the signal  $x_o$ , a more natural question in the design of a system is the following: How many measurements does D-AMP require to recover every signal  $x_o \in C$ ? The following result addresses this question.

**Proposition 1.** *Suppose that for signal class  $C$  the denoiser  $D_\sigma$  is proper at level  $\kappa$ . Then*

$$\sup_{x_o \in C} \delta^*(x_o) \leq \kappa.$$

*Proof.* The proof of this proposition is a simple application of the state evolution equation. Similar to the proof of Lemma 1 define

$$(\sigma^t(x_o, \delta, \sigma_w^2))^2 = \frac{\theta^t(x_o, \delta, \sigma_w^2)}{\delta}.$$

Also for notational simplicity we use the notation  $\sigma^t$  instead of  $\sigma^t(x_o, \delta, 0)$  in the equation below. According to state evolution we have

$$\begin{aligned} (\sigma^{t+1})^2 &= \frac{1}{n\delta} \mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t\epsilon) - x_o\|_2^2 \\ &= \frac{(\sigma^t)^2}{n\delta(\sigma^t)^2} \mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t\epsilon) - x_o\|_2^2 \\ &\leq \frac{(\sigma^t)^2}{\delta} \sup_{x_o \in C} \frac{\mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t\epsilon) - x_o\|_2^2}{n(\sigma^t)^2} \\ &\leq \frac{\kappa(\sigma^t)^2}{\delta}. \end{aligned} \tag{17}$$

It is straightforward to see that

$$(\sigma^t(x_o, \delta, 0))^2 \leq \left(\frac{\kappa}{\delta}\right)^t (\sigma^0(x_o, \delta, 0))^2.$$

Hence, if  $\delta > \kappa$ , then  $(\sigma^t(x_o, \delta, 0))^2 \rightarrow 0$  as  $t \rightarrow \infty$ .  $\square$

We can apply Proposition 1 to the examples of Section III-B and derive some well-known results, such as the phase transition of AMP with the soft-threshold denoiser [10].

If our denoiser is only nearly proper, perfect recovery may not be possible. However, we can use the same technique to bound the recovery error of D-AMP.

**Lemma 2.** *Let  $D_\sigma$  denote a near proper family of denoisers with levels  $\kappa$  and  $B$ , as defined in Definition 2. Then, if  $\delta > \kappa$ , the error of D-AMP is upper bounded by*

$$\lim_{t \rightarrow \infty} (\sigma^t(x_o, \delta, 0))^2 \leq \frac{B}{\delta - \kappa}.$$

*Proof.* The proof of this result is much like the one used for proper denoisers. Again define  $\sigma^t(x_o, \delta, \sigma_w^2) = \frac{\theta^t(x_o, \delta, \sigma_w^2)}{\delta}$ . Using the state evolution and the definition of near proper we have

$$\begin{aligned} & (\sigma^{t+1}(x_o, \delta, 0))^2 \\ &= \frac{1}{n\delta} \mathbb{E} \|D_{\sigma^t(x_o, \delta, 0)}(x_o + \sigma^t(x_o, \delta, 0)\epsilon) - x_o\|_2^2 \\ &\leq \frac{\kappa(\sigma^t(x_o, \delta, 0))^2 + B}{\delta}. \end{aligned}$$

Hence

$$(\sigma^t(x_o, \delta, 0))^2 \leq \left(\frac{\kappa}{\delta}\right)^t \frac{\|x_o\|_2^2}{n} + \left(\frac{1 - (\kappa/\delta)^t}{1 - \kappa/\delta}\right) \frac{B}{\delta}.$$

For  $\delta > \kappa$ , the limit of this sequence is as follows

$$\lim_{t \rightarrow \infty} (\sigma^t(x_o, \delta, 0))^2 \leq \frac{B}{\delta - \kappa}.$$

□

Note that the proof techniques employed above was first developed in [10] and was later employed to establish the phase transition of AMP extensions [35]. There are some minor differences between our derivation and the derivations presented in the other papers since we have not adopted the minimax setting.

### E. Noise sensitivity of D-AMP

In Section III-D we considered the performance of D-AMP in the noiseless setting where  $\sigma_w^2 = 0$ . This section will be devoted to the analysis of D-AMP in the presence of the measurement noise. Here we assume that the denoiser is near proper at levels  $\kappa$  and  $B$ , i.e.,

$$\sup_{\sigma^2} \sup_{x_o \in C} \frac{\mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2}{n} \leq \kappa\sigma^2 + B. \quad (18)$$

The following result shows that D-AMP is robust to the measurement noise. Let  $\theta^\infty(x_o, \sigma_w^2, \delta)$  denote the fixed point of the state evolution equation. Since there is measurement noise,  $\theta^\infty(x_o, \sigma_w^2, \delta) \neq 0$ , i.e., D-AMP will not recover  $x_o$  exactly. We define the noise sensitivity of D-AMP as

$$NS(\sigma_w^2, \delta) = \sup_{x_o \in C} \theta^\infty(x_o, \delta, \sigma_w^2).$$

The following proposition provides an upper bound for the noise sensitivity as a function of the number of measurements and the variance of the measurement noise.

**Proposition 2.** *Let  $D_\sigma$  denote a near proper family of denoisers at levels  $\kappa$  and  $B$ . Then, for  $\delta > \kappa$ , the noise sensitivity of D-AMP satisfies*

$$NS(\sigma_w^2, \delta) \leq \frac{\kappa\sigma_w^2 + B}{1 - \frac{\kappa}{\delta}}. \quad (19)$$

*Proof.* Note that  $\theta^\infty(x_o, \delta, \sigma_w^2)$  is a fixed point of the state evolution equation and hence it satisfies

$$\theta^\infty(x_o, \delta, \sigma_w^2) = \frac{1}{n} \mathbb{E} \|D_{\sigma^\infty}(x_o + \sigma^\infty(x_o, \delta, \sigma_w^2)\epsilon) - x_o\|_2^2,$$

where  $\sigma^\infty(x_o, \delta, \sigma_w^2) = \sqrt{\theta^\infty(x_o, \delta, \sigma_w^2)/\delta + \sigma_w^2}$ . Therefore,

$$\begin{aligned} NS(\sigma_w^2, \delta) &= \sup_{x_o \in C} \theta^\infty(x_o, \delta, \sigma_w^2) \\ &= \sup_{x_o \in C} \frac{1}{n} \mathbb{E} \|D_{\sigma^\infty}(x_o + \sigma^\infty(x_o, \delta, \sigma_w^2)\epsilon) - x_o\|_2^2 \\ &\leq \sup_{x_o \in C} \kappa(\sigma^\infty(x_o, \tau, \sigma_w^2))^2 + B \\ &= \sup_{x_o \in C} \kappa \left( \frac{\theta^\infty(x_o, \delta, \sigma_w^2)}{\delta} + \sigma_w^2 \right) + B \\ &= \frac{\kappa}{\delta} NS(\sigma_w^2, \delta) + \kappa\sigma_w^2 + B. \end{aligned}$$

A simple calculation completes the proof. □

Substituting in  $B = 0$  into the above result gives the noise sensitivity for proper denoisers.

$$NS(\sigma_w^2, \delta) \leq \frac{\kappa\sigma_w^2}{1 - \frac{\kappa}{\delta}}. \quad (20)$$

There are several interesting features of this proposition that we would like to emphasize.

**Remark 3.** *The bound we presented in Proposition 2 is a worst case analysis. The bound may be achieved for certain signals in  $C$  and certain noise variances. However, for most signals in  $C$  and most noise variances D-AMP will perform better than what is predicted by the bound. Figure 8 shows the performance of BM3D-AMP in terms of the standard deviation of the noise.*

The technique we employed above was first developed in [19]. The result we derived in Proposition 2 can be considered as a generalization of the result of [19] to much broader class of denoisers.

As an aside, upper and lower bounds were recently derived for the minimax noise sensitivity of any recovery algorithm when the measurement matrix is i.i.d. Gaussian and the compressively sampled signal is sparse [49]. Note that while our results can be applied to sparse signals, they have been derived under much more general setting. In this section we discussed upper bounds on the noise sensitivity. See Section III-G for some preliminary results on the lower bound.

### F. Tuning the parameters of D-AMP

Practical denoisers typically have a few free parameters and the denoisers' performance relies on the effective tuning of these parameters. One of the simplest examples of a denoiser with parameters is soft-thresholding (introduced in Example

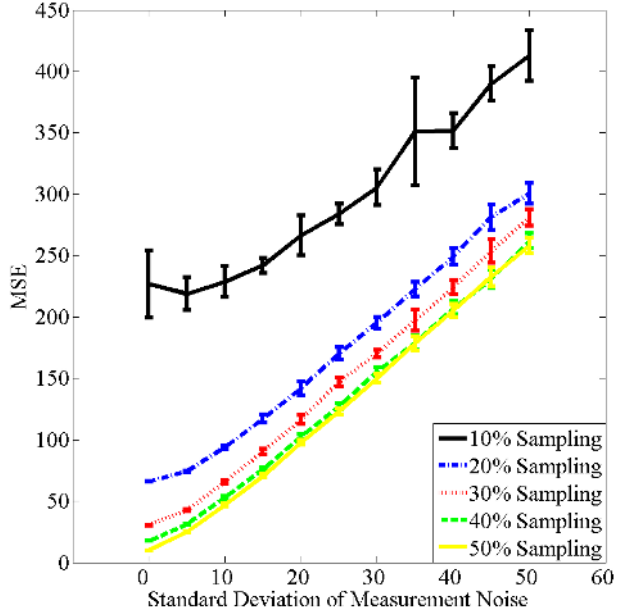


Fig. 8. The MSE of BM3D-AMP reconstructions of  $128 \times 128$  Barbara test image with varying amounts of measurement noise at different sampling rates ( $\delta$ ).

2), for which the threshold can be regarded as a parameter. There exists extensive literature on tuning the free parameters of denoisers [64], [65]. Diverse and powerful algorithms such as SURE (Stein's Unbiased Risk Estimation) have been proposed for this purpose [66].

D-AMP can employ any of these tuning schemes. However, once we use a denoising algorithm in the D-AMP framework the problem of tuning the free parameters of the denoiser seems to become dramatically more difficult: to produce good performance from D-AMP the parameters must be tuned jointly across different iterations. To state this challenge we overload our notation of a denoiser to  $D_{\sigma, \tau}$ , where  $\tau$  denotes the denoiser's parameters. According to this notation the state evolution is given by

$$o^{t+1}(\tau^0, \tau^1, \dots, \tau^t) = \frac{1}{n} \mathbb{E} \|D_{\sigma^t, \tau^t}(x_o + \sigma^t \epsilon) - x_o\|_2^2,$$

where  $(\sigma^t)^2 = \frac{o^t(\tau^0, \tau^1, \dots, \tau^{t-1})}{\delta} + \sigma_w^2$ . Note that we have changed our notation for the state evolution variables to emphasize the dependence of  $o^{t+1}$  on the choice of the parameters we pick at at the previous iterations. The first question that we ask is the following: What does the optimality of  $\tau^0, \tau^1, \dots, \tau^t$  mean? Suppose that the sequence of parameters  $\tau^t$  is bounded.

**Definition 4.** A sequence of parameters  $\tau_*^1, \dots, \tau_*^t$  is called optimal at iteration  $t + 1$  if

$$o^{t+1}(\tau_*^0, \dots, \tau_*^t) = \min_{\tau^0, \tau^1, \dots, \tau^t} o^{t+1}(\tau^0, \tau^1, \dots, \tau^t).$$

Note that  $\tau_*^0, \dots, \tau_*^t$  is optimal in the sense that they produce the smallest mean square error D-AMP can achieve

after  $t$  iterations. This definition was first given in [20] for the AMP algorithm based on soft-thresholding.

It seems from our formulation that we should solve a joint optimization on  $\tau^0, \dots, \tau^t$  to obtain the optimal values of these parameters. However, it turns out that in D-AMP the optimal parameters can be found much more easily. Consider the following greedy algorithm for setting the parameters:

- (i) Tune  $\tau^0$  such that  $o^1(\tau^0)$  is minimized. Call the optimal value  $\tau_*^0$ .
- (ii) If  $\tau^0, \dots, \tau^{t-1}$  are set to  $\tau_*^0, \dots, \tau_*^{t-1}$ , then set  $\tau^t$  such that it minimizes  $o^{t+1}(\tau_*^0, \dots, \tau_*^{t-1}, \tau^t)$ .

Note that the above strategy is a greedy parameter selection. The following result proves that in the context of D-AMP this greedy strategy is optimal:

**Lemma 3.** Suppose that the denoiser  $D_{\sigma, \tau}$  is monotone in the sense that  $\inf_{\tau} \mathbb{E} \|D_{\sigma, \tau}(x_o + \sigma \epsilon) - x_o\|_2^2$  is a non-decreasing function of  $\sigma$ . If  $\tau_*^0, \dots, \tau_*^t$  is generated according to the greedy tuning algorithm described above, then

$$o^{t+1}(\tau_*^0, \dots, \tau_*^t) \leq o^{t+1}(\tau^0, \dots, \tau^t), \quad \forall \tau^0, \dots, \tau^t,$$

for every  $t$ .

*Proof.* Our proof is based on an induction. According to the first step of our procedure we know that

$$o^1(\tau_*^0) \leq o^1(\tau^0), \quad \forall \tau^0.$$

Now suppose that the claim of the theorem is true for every  $t \leq T$ . We would like to prove that the result also holds for  $t = T + 1$ , i.e.,

$$o^{T+1}(\tau_*^0, \dots, \tau_*^T) \leq o^{T+1}(\tau^0, \dots, \tau^T), \quad \forall \tau^0, \dots, \tau^T.$$

Suppose that it is not true and for  $\tau_o^0, \dots, \tau_o^T$  we have

$$o^{T+1}(\tau_o^0, \dots, \tau_o^T) > o^{T+1}(\tau_*^0, \dots, \tau_*^T). \quad (21)$$

Clearly,

$$o^{T+1}(\tau_*^1, \tau_*^2, \dots, \tau_*^T) = \frac{1}{n} \mathbb{E} \|D_{\sigma^T, \tau_*^T}(x_o + \sigma_*^T \epsilon) - x_o\|_2^2,$$

where  $(\sigma_*^T)^2 = \frac{o^T(\tau_*^0, \dots, \tau_*^{T-1})}{\delta} + \sigma_w^2$ . If we define  $(\sigma_o^T)^2 = \frac{o^T(\tau_o^0, \dots, \tau_o^{T-1})}{\delta} + \sigma_w^2$ , then according to the induction assumption  $\sigma_*^T \leq \sigma_o^T$ . Therefore, according to the monotonicity of the denoiser

$$\begin{aligned} o^{T+1}(\tau_*^0, \tau_*^1, \dots, \tau_*^T) &= \inf_{\tau^T} \frac{1}{n} \mathbb{E} \|D_{\sigma_*^T, \tau^T}(x_o + \sigma_*^T \epsilon) - x_o\|_2^2 \\ &\leq \inf_{\tau^T} \frac{1}{n} \mathbb{E} \|D_{\sigma_o^T, \tau^T}(x_o + \sigma_o^T \epsilon) - x_o\|_2^2 \\ &\leq \frac{1}{n} \mathbb{E} \|D_{\sigma_o^T, \tau_o^T}(x_o + \sigma_o^T \epsilon) - x_o\|_2^2 \\ &= o^{T+1}(\tau_o^0, \tau_o^1, \dots, \tau_o^T). \end{aligned}$$

This is in contradiction with (21). Hence,

$$o^{T+1}(\tau_*^0, \dots, \tau_*^T) \leq o^{T+1}(\tau^0, \dots, \tau^T), \quad \forall \tau^0, \dots, \tau^T.$$

□

To summarize the above discussion, greedy parameter tun-

ing is optimal for D-AMP, thus the tuning of D-AMP is as simple (or as difficult) as the tuning of the denoising algorithm that is employed in D-AMP. Many researchers in the area of signal denoising have optimized the parameters of state-of-the-art denoisers, such as BM3D. Lemma 3 implies that optimally tuned denoisers will induce the best possible performance from D-AMP. Therefore, the tuning of D-AMP has already been thoroughly addressed in the denoising literature [64]–[66].

### G. Optimality of D-AMP

1) *Problem definition*: D-AMP is a framework by which to employ denoisers to solve linear inverse problems. But is D-AMP optimal? In other words, given a family of denoisers,  $D_\sigma$ , for a set  $C$ , can we come up with an algorithm for recovering  $x_o$  from  $y = \mathbf{A}x_o + w$  that outperforms D-AMP? Note that this problem is ill-posed in the following sense: the denoising algorithm might not capture all the structures that are present in the signal class  $C$ . Hence, a recovery algorithm employs extra structures not used by the denoiser (and thus not used by D-AMP) clearly might outperform D-AMP. In the following sections we use two different approaches to analyze the optimality of D-AMP.

2) *Uniform optimality*: Let  $\mathcal{E}_\kappa$  denote the set of all classes of signals  $C$  for which there exists a family of denoisers  $D_\sigma^C$  that satisfies

$$\sup_{\sigma^2} \sup_{x_o \in C} \frac{\mathbb{E} \|D_\sigma^C(x_o + \sigma\epsilon) - x_o\|_2^2}{n\sigma^2} \leq \kappa. \quad (22)$$

We know from Proposition 1 that for any  $C \in \mathcal{E}_\kappa$ , D-AMP recovers all the signals in  $C$  from  $\delta > \kappa$  measurements.

We now ask our uniform optimality question: *Does there exist any other signal recovery algorithm that can recover all the signals in all these classes with fewer measurements than D-AMP?* If the answer is affirmative, then D-AMP is sub-optimal in the uniform sense, meaning there exists an approach that outperforms D-AMP uniformly over all classes in  $\mathcal{E}_\kappa$ . The following proposition shows that any recovery algorithm requires at least  $m = \kappa n$  measurements for accurate recovery, i.e., D-AMP is optimal in this sense.

**Proposition 3.** *If  $m^*$  denotes the minimum number of measurements required (by any recovery algorithm) for a set  $C \in \mathcal{E}_\kappa$ , then*

$$\sup_{C \in \mathcal{E}_\kappa} \frac{m^*(C)}{n} \geq \kappa.$$

*Proof.* First note that according to Example 1 any  $\kappa n$  dimensional subspace of  $\mathbb{R}^n$  belongs to  $\mathcal{E}_\kappa$  (assume that  $\kappa n$  is an integer). From the fundamental theorem of linear algebra we know that to recover the vectors in a  $k$  dimensional subspace we require at least  $k$  measurements. Hence

$$\sup_{C \in \mathcal{E}_\kappa} \frac{m^*(C)}{n} \geq \frac{\kappa n}{n} = \kappa. \quad \square$$

According to this simple result, D-AMP is optimal for at least certain classes of signals and certain denoisers. Hence, it cannot be uniformly improved.

3) *Single class optimality*: The uniform optimality framework we introduced above considers a set of signal classes and measures the performance of an algorithm on every class in this set. However, in many applications such as imaging we are interested in the performance of D-AMP on a specific class of signals, such as images. Unfortunately, the uniform optimality framework does not provide any conclusion in such cases. Therefore, in this section we introduce another framework for evaluating the optimality of D-AMP that we call *single class optimality*.

Let  $C$  denote a class of signals. Instead of assuming that we are given a family of denoisers for the signals in class  $C$ , we assume that we can find the denoiser that brings about the best performance from D-AMP. This ensures that D-AMP employs as much information as it can about  $C$ . Let  $\theta_D^\infty(x_o, \delta, \sigma_w^2)$  denote the fixed point of the state evolution equation given in (16). Note that we have added a subscript  $D$  to our notation for  $\theta$  to indicate the dependence of this quantity on the choice of the denoiser. The best denoiser for D-AMP is a denoiser that minimizes  $\theta_D^\infty(x_o, \delta, \sigma_w^2)$ . Note that according to Finding 1,  $\theta_D^\infty(x_o, \delta, \sigma_w^2)$  corresponds to the mean square error of the final estimate that D-AMP returns.

**Definition 5.** *A family of denoisers  $D_\sigma^*$  is called minimax optimal for D-AMP at noise level  $\sigma_w^2$ , if it achieves*

$$\inf_{D_\sigma} \sup_{x_o \in C} \theta_D^\infty(x_o, \delta, \sigma_w^2).$$

Note that according to our definition, the optimal denoiser may depend on both  $\sigma_w^2$  and  $\delta$  and it is not necessarily unique. We call the version of D-AMP that employs  $D_\sigma^*$ ,  $D^*$ -AMP.

Armed with this definition, we formally ask the single class optimality question: *Can we provide a new algorithm that can recover signals in class  $C$  with fewer measurements than  $D^*$ -AMP?* If negative, it means that if we employ the optimal denoiser for D-AMP algorithm no other algorithm can outperform D-AMP. Unfortunately, we will show that there are signal classes for which D-AMP is not optimal in this sense. Our proof requires the following standard definition from statistics text books [61]:

**Definition 6.** *The minimax risk of a set of signals  $C$  at the noise level  $\sigma^2$  is defined as*

$$R_{MM}(C, \sigma^2) = \inf_D \sup_{x_o \in C} \mathbb{E} \|D(x_o + \sigma\epsilon) - x_o\|_2^2,$$

where the expected value is with respect to  $\epsilon \sim N(0, I)$ . If  $D_\sigma^M$  achieves  $R_{MM}(C, \sigma^2)$ , then it will be called the family of minimax denoisers for the set  $C$  under the square loss.

**Proposition 4.** *The family of minimax denoisers for  $C$  is a family of optimal denoisers for D-AMP. Furthermore, in order to recover every  $x_o \in C$ ,  $D^*$ -AMP requires at least  $n\kappa_{MM}$  measurements:*

$$\kappa_{MM} = \sup_{\sigma^2 > 0} \frac{R_{MM}(\sigma^2)}{n\sigma^2}.$$

*Proof.* Since the proof of this result is slightly more involved, we postpone it to Appendix A.  $\square$

Based on this result, we can simplify the single class optimality question: *Does there exist any recovery algorithm that can recover every  $x_o \in C$  from fewer observations than  $n\kappa_{MM}$ ?* Unfortunately, the answer is affirmative.

Consider the following extreme example. Let  $B_k^n$  denote the class of signals that consist of  $k$  ones and  $n - k$  zeros. Define  $\rho = k/n$  and let  $\phi(z)$  denote the density function of a standard normal random variable.

**Proposition 5.** *For very high dimensional problems, there are recovery algorithms that can recover signals in  $B_k$  accurately from 1 measurement. On the other hand,  $D^*$ -AMP requires at least  $n(\kappa_{MM} - o(1))$  measurement to recover signals from this class, where*

$$\kappa_{MM} = \sup_{\sigma^2 > 0} \frac{1}{\sigma^2} \mathbb{E}_{z_1 \sim \phi} \left( \frac{\rho \phi_\sigma(z_1)}{\rho \phi_\sigma(z_1) + (1 - \rho) \phi_\sigma(z_1 + 1)} - 1 \right)^2 + \mathbb{E}_{z_1 \sim \phi} \left( \frac{\rho \phi_\sigma(z_1 - 1)}{\rho \phi_\sigma(z_1 - 1) + (1 - \rho) \phi_\sigma(z_1)} \right)^2 (1 - \rho),$$

where  $\phi_\sigma(z) = \phi(z/\sigma)$ .

The proof of this result is slightly more involved and hence is postponed to Appendix B. According to this proposition, since  $\kappa_{MM}$  is non-zero, the number of measurements  $D^*$ -AMP requires is proportional to the ambient dimension  $n$ , while the actual number of measurements that is required for recovery is equal to 1. Hence, in such cases  $D^*$ -AMP is sub-optimal.

However, it is also important to note that while  $D$ -AMP is sub-optimal for this class, according to Proposition 3  $D$ -AMP is optimal for other classes. Characterizing the classes of signals for which  $D$ -AMP is optimal is left as an open direction for future research. Despite this sub-optimality result, we will show in Section VII that  $D$ -AMP provides impressive results for the class of natural images and outperforms state-of-the-art recovery algorithms.

#### H. Additional miscellaneous properties of $D$ -AMP

1) *Better denoisers lead to better recovery:* This intuitive result is a key feature of  $D$ -AMP. We formalize it below.

**Theorem 1.** *Let a family of denoisers  $D_\sigma^1$  be a better denoiser than a family  $D_\sigma^2$  for signal  $x_o$  in the following sense:*

$$\frac{\mathbb{E} \|D_\sigma^1(x_o + \sigma\epsilon) - x_o\|_2^2}{n\sigma^2} \leq \frac{\mathbb{E} \|D_\sigma^2(x_o + \sigma\epsilon) - x_o\|_2^2}{n\sigma^2}, \quad \forall \sigma^2 > 0. \quad (23)$$

Also, let  $\theta_{D_i}^\infty(x_o, \delta, \sigma_w^2)$  denote the fixed point of state evolution for denoiser  $D_i$ . Then,

$$\theta_{D_1}^\infty(x_o, \delta, \sigma_w^2) \leq \theta_{D_2}^\infty(x_o, \delta, \sigma_w^2).$$

*Proof.* The proof of this result is straightforward. Since, the state evolution of  $D^1$  is uniformly lower than  $D^2$ , its fixed point is lower as well.  $\square$

2)  *$D$ -AMP as a regularization technique:* Explicit regularization is a popular technique to recover signals from an undersampled set of linear measurements [5], [22], [29]–[33], [67]. In these approaches a cost function,  $J(x)$ , also known

as a regularizer, is considered on  $\mathbb{R}^n$ . This function returns large values for  $x \notin C$  and returns small values for  $x \in C$ . Regularized techniques recover  $x_o$  from measurements  $y$  by setting up and solving the following optimization problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2} \|y - \mathbf{A}x\|_2^2 + \lambda J(x). \quad (24)$$

Since in many cases  $J(x)$  is non-convex and non-differentiable, iterative heuristic methods have been proposed for solving the above optimization problem.<sup>12</sup>  $D$ -AMP provides another heuristic approach for solving (24). It has two main advantages over the other heuristics: (i)  $D$ -AMP can be analyzed by the state evolution theoretically. Hence, we can theoretically predict the number of measurements required and the noise sensitivity of  $D$ -AMP. (ii) The performance of most heuristic methods depend on their free parameters. As discussed in Section III-F there are efficient approaches for tuning the parameters of  $D$ -AMP optimally. Below we briefly review the application of  $D$ -AMP for solving (24).

Assume that there exists a computationally efficient scheme for solving the optimization problem  $\chi_J(u; \lambda) = \operatorname{argmin} \frac{1}{2} \|u - x\|_2^2 + \lambda J(x)$ .  $\chi_J(u, \lambda)$  is called the proximal operator for the function  $J$ . The  $D$ -AMP algorithm for solving (24) is given by

$$\begin{aligned} x^{t+1} &= \chi_J(x^t + \mathbf{A}^* z^t; \lambda^t), \\ z^t &= y - \mathbf{A}x^t + z^{t-1} \operatorname{div} \chi_J(x^{t-1} + \mathbf{A}^* z^{t-1}; \lambda^{t-1})/m. \end{aligned} \quad (25)$$

Considering  $\chi_J$  as a denoiser, this algorithm has exactly the same interpretation as our generic  $D$ -AMP algorithm. Furthermore, if the explicit calculation of the Onsager correction term is challenging we can employ the Monte Carlo technique that will be discussed in Section V-B.

#### IV. CONNECTION WITH OTHER STATE EVOLUTIONS

In Section III we introduced a new, “deterministic” state evolution (SE) and used it to analyze  $D$ -AMP. Here we review this SE and compare it with AMP’s Bayesian SE, which was first introduced in [10], [17].

##### A. Deterministic state evolution

The deterministic SE assumes that  $x_o$  is an arbitrary but fixed vector in  $C$ . Starting from  $\theta^0 = \frac{\|x_o\|_2^2}{n}$ , the deterministic SE generates a sequence of numbers through the following iterations:

$$\theta^{t+1}(x_o, \delta, \sigma_w^2) = \frac{1}{n} \mathbb{E}_\epsilon \|D_{\sigma^t}(x_o + \sigma^t \epsilon) - x_o\|_2^2, \quad (16)$$

where  $(\sigma^t)^2 = \frac{1}{\delta} \theta^t(x_o, \delta, \sigma_w^2) + \sigma_w^2$  and  $\epsilon \sim N(0, I)$ .

##### B. Bayesian state evolution

The Bayesian SE assumes that  $x_o$  is a vector drawn from a probability density function (pdf)  $p_x$ , where the support of  $p_x$  is a subset of  $C$ . Starting from  $\bar{\theta}^0 = \frac{\|x_o\|_2^2}{n}$ , the Bayesian

<sup>12</sup>Many of these methods solve (24) accurately when  $J$  is convex.



SE generates a sequence of numbers through the following iterations:

$$\bar{\theta}^{t+1}(p_x, \delta, \sigma_w^2) = \frac{1}{n} \mathbb{E}_{x_o, \epsilon} \|D_{\bar{\sigma}^t}(x_o + \bar{\sigma}^t \epsilon) - x_o\|_2^2, \quad (26)$$

where  $(\bar{\sigma}^t)^2 = \frac{1}{\delta} \bar{\theta}^t(p_x, \delta, \sigma_w^2) + \sigma_w^2$ . We have used the notation  $\bar{\theta}$  to distinguish the Bayesian SE from its deterministic counterpart. In Definition 6 we presented a definition of the optimal denoiser under the deterministic framework. One can do the same for the Bayesian framework.

**Definition 7.** A family of denoisers  $\tilde{D}_\sigma$  is called *Bayes-optimal for D-AMP at noise level  $\sigma_w^2$* , if it achieves

$$\inf_{D_\sigma} \bar{\theta}_D^\infty(p_x, \delta, \sigma_w^2).$$

It is straightforward to see that the family  $\tilde{D}_\sigma(x_o + \sigma\epsilon) = \mathbb{E}(x_o \mid x_o + \sigma\epsilon)$  is Bayes-optimal for D-AMP.

While the deterministic and Bayesian SEs are different, we can establish a connection between them by employing standard results in theoretical statistics regarding the connection between the minimax risk and the Bayesian risk. Next section briefly discusses this connection.

### C. Connection between the two state evolutions

In this section we would like to establish a connection between the fixed points of the Bayes-optimal denoisers and the minimax-optimal denoisers for D-AMP. Let  $\bar{\theta}^\infty(p_x, \delta, \sigma_w^2)$  denote the fixed point of the Bayesian SE (26) associated with the family of Bayes-optimal denoisers from Definition 7. Also, let  $\theta^\infty(x_o, \delta, \sigma_w^2)$  denote the fixed point of the deterministic SE (16) for the family of minimax denoisers from Definition 5.

**Theorem 2.** Let  $\mathcal{P}$  denote the set of all distributions whose support is a subset of  $C$ . Then,

$$\sup_{p_x \in \mathcal{P}} \bar{\theta}^\infty(p_x, \delta, \sigma_w^2) \leq \sup_{x_o \in C} \theta^\infty(x_o, \delta, \sigma_w^2).$$

*Proof.* For an arbitrary family of denoisers  $D_\sigma$  we have

$$\mathbb{E}_{x_o, \epsilon} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \leq \sup_{x_o \in C} \mathbb{E}_\epsilon \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2. \quad (27)$$

If we take the minimum with respect to  $D_\sigma$  on both sides of (27), we obtain the following inequality

$$\mathbb{E}_{x_o, \epsilon} \|\tilde{D}_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \leq \sup_{x_o \in C} \mathbb{E}_\epsilon \|D^{MM}(x_o + \sigma\epsilon) - x_o\|_2^2,$$

where  $D^{MM}$  denotes the minimax denoiser and  $\tilde{D}_\sigma(x_o + \sigma\epsilon)$  denotes  $\mathbb{E}(x_o \mid x_o + \sigma\epsilon)$ . Let  $(\bar{\sigma}^\infty)^2 = \frac{\bar{\theta}^\infty(x_o, \delta, \sigma_w^2)}{\delta} + \sigma_w^2$  and  $(\sigma_{mm}^\infty)^2 = \frac{\theta^\infty(x_o, \delta, \sigma_w^2)}{\delta} + \sigma_w^2$ . Also, for notational simplicity assume that  $\sup_{x_o \in C} \mathbb{E}_\epsilon \|D^{MM}(x_o + \bar{\sigma}^\infty \epsilon) - x_o\|_2^2$  is achieved at a certain value  $x_{mm}$ . We then have

$$\begin{aligned} \bar{\theta}^\infty(p_x, \delta, \sigma_w^2) &= \frac{\mathbb{E}_{x_o, \epsilon} \|\tilde{D}_{\bar{\sigma}^\infty}(x_o + \bar{\sigma}^\infty \epsilon) - x_o\|_2^2}{n} \\ &\leq \frac{\mathbb{E}_\epsilon \|D_{\bar{\sigma}^\infty}^{MM}(x_{mm} + \bar{\sigma}^\infty \epsilon) - x_{mm}\|_2^2}{n}. \end{aligned} \quad (28)$$

This inequality implies that  $\bar{\theta}^\infty(p_x, \delta, \sigma_w^2)$  is below the fixed point of the deterministic SE using  $D^{MM}$  at  $x_{mm}$ .

Therefore, because  $\sup_{x_o \in C} \theta^\infty(x_o, \delta, \sigma_w^2)$  will be equal to or above the fixed point of  $D^{MM}$  at  $x_{mm}$ , it will satisfy  $\sup_{p_x \in \mathcal{P}} \bar{\theta}^\infty(p_x, \delta, \sigma_w^2) \leq \sup_{x_o \in C} \theta^\infty(x_o, \delta, \sigma_w^2)$ .  $\square$

Under some general conditions it is possible to prove that

$$\begin{aligned} \sup_{\pi \in \mathcal{P}} \inf_{D_\sigma} \mathbb{E}_{x_o, \epsilon} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \\ = \inf_{D_\sigma} \sup_{x_o \in C} \mathbb{E}_\epsilon \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2. \end{aligned} \quad (29)$$

For instance, if we have

$$\begin{aligned} \sup_{\pi \in \mathcal{P}} \inf_{D_\sigma} \mathbb{E}_{x_o, \epsilon} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \\ = \inf_{D_\sigma} \sup_{\pi \in \mathcal{P}} \mathbb{E}_{x_o, \epsilon} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2, \end{aligned}$$

then (29) holds as well. Since we work with square loss in the SE, swapping the infimum and supremum is permitted under quite general conditions on  $\mathcal{P}$ . For more information, see Appendix A of [68]. If (29) holds, then we can follow similar steps as in the proof of Theorem 2 to prove that under the same set of conditions we can have

$$\sup_{p_x \in \mathcal{P}} \inf_{D_\sigma} \bar{\theta}^\infty(p_x, \delta, \sigma_w^2) = \inf_{D_\sigma} \sup_{x_o \in C} \theta^\infty(x_o, \delta, \sigma_w^2).$$

In words, the supremum of the fixed point of the Bayesian SE with the Bayes-optimal denoiser is equivalent to the supremum of the fixed point of the deterministic SE with the minimax denoiser.

### D. Why bother?

Considering that the deterministic and Bayesian SEs look so similar, and under certain conditions have the same supremums, it is natural to ask why we developed the deterministic SE at all. That is, what is gained by using SE (16) rather than (26)?

The deterministic SE is useful because it enables us to deal with signals with poorly understood distributions. Take, for instance, natural images. To use the Bayesian SE on imaging problems, we would first need to characterize all images according to some generalized, almost assuredly inaccurate, pdf. In contrast, the deterministic SE deals with specific signals, not distributions. Thus, even without knowledge of the underlying distribution, so long as we can come up with representative test signals, we can use the deterministic SE. Because the SE shows up in the parameter tuning, noise sensitivity, and performance guarantees of AMP algorithms, being able to deal with arbitrary signals is invaluable.

## V. CALCULATION OF THE ONSAGER CORRECTION TERM

So far, we have emphasized that the key to the success of approximate message passing algorithms is the Onsager correction term,  $z^{t-1} \text{div} D_{\bar{\sigma}^{t-1}}(x^{t-1} + \mathbf{A}^* z^{t-1})/m$ , but we have not yet addressed how one can calculate it for an arbitrary denoiser. In this section we provide some guidelines on the calculation of this term. If the input-output relation of the denoiser is known explicitly, then calculating the divergence,  $\text{div} D(x)$ , and thus the Onsager correction term, is usually

straightforward.<sup>13</sup> We will review some popular denoisers and calculate the corresponding Onsager correction terms in the next section. However, most state-of-the-art denoisers are complicated algorithms for which the input-output relation is not explicitly known. In Section V-B we show that, even without an explicit input-output relationship, we can calculate a good approximation for the Onsager correction term.

### A. Soft-thresholding for sparse and low-rank signals

Three of the most popular signal classes in the literature are sparse, group-sparse, and low-rank signals (when the signal has a matrix form). The most popular denoisers for these signals are soft-thresholding, block soft-thresholding, and singular value soft-thresholding, respectively. The goal of this section is to derive the Onsager correction term for each of these denoisers. Most of the results mentioned in this section have been derived elsewhere. We summarize these results to help the reader understand the steps involved in explicitly computing the Onsager correction term.

1) *Soft-thresholding*: Let  $\eta_\tau$  denote the soft-threshold function.  $\eta_\tau(x)$  for  $x \in \mathbb{R}^n$  denotes the component-wise application of the soft-threshold function to the elements of  $x$ . In this case we have  $\text{div}\eta_\tau(x) = \sum_{i=1}^n \mathbb{I}(|x_i| > \tau)$ , where  $\mathbb{I}$  denotes the indicator function.

2) *Block soft-thresholding*: For a vector  $x_B \in \mathbb{R}^B$  block soft-thresholding is defined as  $\eta_\tau^B(x_B) = (\|x_B\|_2 - \tau) \frac{x_B}{\|x_B\|_2}$ . In other words, the threshold function retains the phase of the vector  $x_B$  and shrinks its magnitude toward zero. Let  $n = MB$  and  $x = [(x_B^1)^T, (x_B^2)^T, \dots, (x_B^M)^T]^T$ . The notation  $\eta_\tau^B(x)$  is defined as the block soft-thresholding function that is applied to each individual block. The divergence of block soft-thresholding can then be calculated according to

$$\text{div}\eta_\tau^B(x) = \sum_{\ell=1}^M \left( B - \frac{(B-2)^2}{\|x_B^\ell\|_2^2} \right) \mathbb{I}(\|x_B^\ell\|_2 \geq \tau).$$

This result was derived in [35], [36], [69].

3) *Singular value thresholding*: Let  $\mathbf{X}_o \in \mathbb{R}^{n \times n}$  denote our signal of interest. If  $\mathbf{X}_o$  is low-rank then it can be estimated accurately from its noisy version  $\Phi = \mathbf{X}_o + \sigma \mathbf{W}$  where  $\mathbf{W}_{ij}$  denote i.i.d.,  $N(0, 1)$  random variables. If the singular value decomposition of  $\Phi$  is given by  $\mathbf{USV}^T$ , with  $\mathbf{S} = \text{diag}(\sigma_1, \dots, \sigma_n)$ , where  $\sigma_i$ s denote the singular values of  $\Phi$ , then the estimate of  $\mathbf{X}_o$  has the form

$$\begin{aligned} \hat{\mathbf{X}} &= \mathbf{SVT}_\lambda(\Phi) \\ &= \mathbf{U} \text{diag}((\sigma_1 - \lambda)_+, (\sigma_2 - \lambda)_+, \dots, (\sigma_n - \lambda)_+) \mathbf{V}^T, \end{aligned}$$

in which  $\lambda$  is a regularization parameter that can be optimized for the best performance. Again this denoiser can be employed in the D-AMP framework to recover low-rank matrices from their underdetermined set of linear equations. To calculate the Onsager correction term we should compute  $\text{div}\mathbf{SVT}_\lambda(\Phi)$ .

<sup>13</sup>In the context of this work the divergence  $\text{div}D(x)$  is simply the sum of the partial derivatives with respect to each element of  $x$ , i.e.,  $\text{div}D(x) = \sum_{i=1}^n \frac{\partial D(x_i)}{\partial x_i}$ .

According to [70] the divergence of singular value thresholding is given by

$$\text{div}\mathbf{SVT}_\lambda(\Phi) = \sum_{i=1}^n \mathbb{I}(\sigma_i > \lambda) + 2 \sum_{i,j=1, i \neq j}^n \frac{\sigma_i(\sigma_i - \lambda)_+}{\sigma_i^2 - \sigma_j^2}.$$

### B. Monte Carlo method

While simple denoisers often yield a closed form for their divergence, high-performance denoisers are often data dependent; making it very difficult to characterize their input-output relation explicitly. Here we explain how a good approximation of the divergence can be obtained in such cases. This method relies on a Monte Carlo technique first developed in [66]. The authors of that work showed that given a denoiser  $D_{\sigma,\tau}(x)$ , using an i.i.d. random vector  $b \sim N(0, I)$ , we can estimate the divergence with

$$\begin{aligned} \text{div}D_{\sigma,\tau} &= \lim_{\epsilon \rightarrow 0} \mathbb{E}_b \left\{ b^* \left( \frac{D_{\sigma,\tau}(x + \epsilon b) - D_{\sigma,\tau}(x)}{\epsilon} \right) \right\} \\ &\approx \mathbb{E}_b \left( \frac{1}{\epsilon} b^* (D_{\sigma,\tau}(x + \epsilon b) - D_{\sigma,\tau}(x)) \right), \end{aligned}$$

for very small  $\epsilon$ .

The only challenge in using this formula is calculating the expected value. This can be done efficiently using Monte Carlo simulation. We generate  $M$  i.i.d.,  $N(0, I)$  vectors  $b^1, b^2, \dots, b^M$ . For each vector  $b^i$  we obtain an estimate of the divergence  $\widehat{\text{div}}^i$ . We then obtain a good estimate of the divergence by averaging

$$\text{div}\hat{D}_{\sigma,\tau} = \frac{1}{M} \sum_{i=1}^M \widehat{\text{div}}^i.$$

According to the weak law of large numbers, as  $M \rightarrow \infty$  this estimate converges to  $\mathbb{E}_b \left( \frac{1}{\epsilon} b^* (D_{\sigma,\tau}(x + \epsilon b) - D_{\sigma,\tau}(x)) \right)$ . When dealing with images, due to the high dimensionality of the signal, we can accurately approximate the expected value using only a single random sample. That is, we can let  $M = 1$ .<sup>14</sup> Note that in this case the calculation of the Onsager correction term is quite efficient and requires only one additional application of the denoising algorithm. In all of the simulations in this paper we have used either the explicit calculation of the Onsager correction term or the Monte Carlo method with  $M = 1$ .

## VI. SMOOTHING A DENOISER

The denoiser used within D-AMP can take on almost any form. However, the state evolution predictions are not necessarily accurate if the denoiser is not Lipschitz continuous. This requirement seems to disallow some popular denoisers with discontinuities, such as hard-thresholding. Figure 9 compares the state evolution predictions for the hard thresholding denoiser alongside the actual performance of D-AMP based on hard thresholding; the state evolution predictions fail entirely. One simple idea to resolve this issue is to “smooth” the

<sup>14</sup>When dealing with short signals ( $n < 1000$ ), rather than images, we found that using additional Monte Carlo samples produced more consistent results.

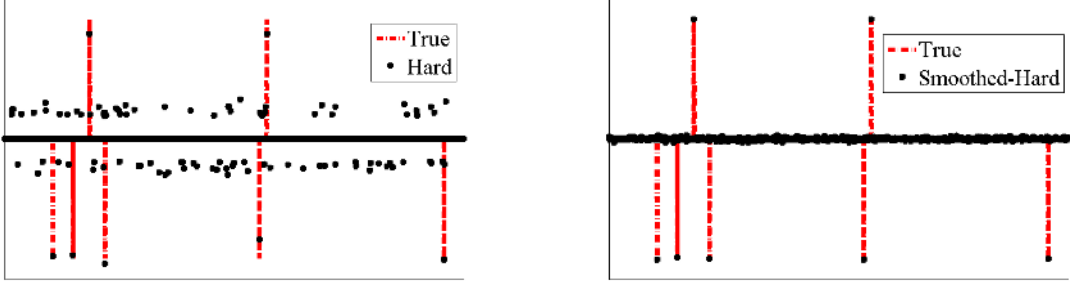


Fig. 10. Reconstructions of a sparse signal that was sampled at a rate of  $\delta = 1/3$ . Notice that D-AMP based on smoothed-hard-thresholding successfully reconstructs the signal whereas D-AMP based on hard-thresholding, does not. The failure of hard-thresholding-based D-AMP is due to the discontinuity of the hard-thresholding denoiser.

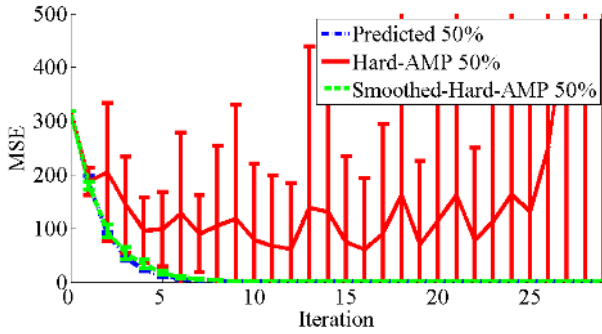


Fig. 9. Predicted and observed intermediate MSEs of hard-thresholding-based AMP, with and without smoothing. Notice that the smoothed version is well predicted by the state evolution whereas hard-thresholding-based AMP without smoothing is not. This discrepancy is due the fact that the hard-thresholding denoiser is not continuous.

denoisers. The smoothed version should behave nearly the same as the original denoiser but, because it has no discontinuities, should satisfy the state evolution equations. The concept of smoothing simple denoisers and this process's effects on the performance of simple denoisers has been analyzed in [71], [72]. Here we explain how smoothing can be applied in practice.

Let  $\eta(x)$  be a discontinuous denoiser. Now define a new denoiser  $\tilde{\eta}(x)$  as follows

$$\tilde{\eta}(x) = \int_{\zeta \in \mathbb{R}^n} \eta(x - \zeta) \frac{1}{(2\pi)^{n/2} r^n} e^{-\|\zeta\|_2^2 / 2r^2} d\zeta, \quad (30)$$

where  $d\zeta = d\zeta_1 d\zeta_2 \dots d\zeta_n$ .

The denoiser  $\tilde{\eta}(x)$  is simply the convolution of  $\eta(x)$  with a Gaussian kernel with standard deviation  $r$ . Note that the width  $r$  dictates the amount of smoothness we apply to  $\eta$ . Larger values of  $r$  lead to a smoother  $\tilde{\eta}$ . Below we present a simple lemma that proves  $\tilde{\eta}(x)$  is in fact smooth.

Suppose that  $\eta(x_1, x_2, \dots, x_n)$  satisfies the following con-

dition:

$$\int |\eta_i(\tilde{\zeta}) \tilde{\zeta}_i| e^{-\frac{\|\tilde{\zeta}\|_2^2}{4r^2}} d\tilde{\zeta} < \infty. \quad (31)$$

Note that this condition implies that  $\eta_i$  is not growing very fast as  $\zeta_1, \zeta_2, \dots, \zeta_n \rightarrow \infty$ .

**Lemma 4.** *If  $\eta$  satisfies (31), then  $\tilde{\eta}(x)$  defined in (30) is continuously differentiable with a bounded derivative and is thus Lipschitz continuous.*

*Proof.* To prove  $\tilde{\eta}$  is continuously differentiable with bounded derivative, we prove that all the partial derivatives exist, are bounded, and are continuous. Let  $\eta = (\eta_1, \eta_2, \dots, \eta_n)$  and  $\tilde{\eta} = (\tilde{\eta}_1, \tilde{\eta}_2, \dots, \tilde{\eta}_n)$ . By a simple change of integration variables we obtain

$$\tilde{\eta}(x) = \int_{\tilde{\zeta} \in \mathbb{R}^n} \eta(\tilde{\zeta}) \frac{1}{(2\pi)^{n/2} r^n} e^{-\|x - \tilde{\zeta}\|_2^2 / 2r^2} d\tilde{\zeta}. \quad (32)$$

Now we calculate the  $j^{\text{th}}$  partial derivative of  $\tilde{\eta}_i$ . Let  $b^j \in \mathbb{R}^n$  denote a vector whose elements are all zero except for the  $j^{\text{th}}$  element, which is equal to one. Then,

$$\begin{aligned} \frac{\partial \tilde{\eta}_i(x)}{\partial x_j} &= \lim_{\gamma \rightarrow 0} \frac{\tilde{\eta}_i(x + \gamma b^j) - \tilde{\eta}_i(x)}{\gamma} \\ &= \lim_{\gamma \rightarrow 0} \int_{\tilde{\zeta} \in \mathbb{R}^n} \frac{\eta_i(\tilde{\zeta})}{(2\pi)^{n/2} r^n} \left( \frac{e^{-\|x + \gamma b^j - \tilde{\zeta}\|_2^2 / 2r^2}}{\gamma} - \frac{e^{-\|x - \tilde{\zeta}\|_2^2 / 2r^2}}{\gamma} \right) d\tilde{\zeta}. \end{aligned} \quad (33)$$

From the mean value theorem we conclude that there exists  $\tilde{\gamma}$  between 0 and  $\gamma$  such that

$$\begin{aligned} &\left( \frac{e^{-\|x + \gamma b^j - \tilde{\zeta}\|_2^2 / 2r^2} - e^{-\|x - \tilde{\zeta}\|_2^2 / 2r^2}}{\gamma} \right) \\ &= \frac{(\tilde{\zeta}_j - \tilde{\gamma} - x_j)}{r^2} e^{-\|x + \tilde{\gamma} b^j - \tilde{\zeta}\|_2^2 / 2r^2}, \end{aligned} \quad (34)$$

where the last equality is due to the fact that the  $j^{\text{th}}$  element

of  $b^j$  is equal to one. Also, note that for  $|\gamma| < 1$  we have

$$\begin{aligned} & \left| \frac{\tilde{\zeta}_j - \tilde{\gamma} - x_j}{r^2} e^{-\|x + \tilde{\gamma} b^j - \tilde{\zeta}\|_2^2 / 2r^2} \right| \\ & \leq \frac{|\tilde{\zeta}_j| + 1 + |x_j|}{r^2} e^{-\sum_{k \neq j} (x_k - \tilde{\zeta}_k)^2 / 2r^2} e^{-\frac{-\tilde{\zeta}_j^2 + 2(|x_j| + 1)|\tilde{\zeta}_j|}{2r^2}}, \end{aligned}$$

where to obtain the last equality we used the fact that all the element of  $b_j$  except the  $j^{\text{th}}$  one are zero. Define

$$h(\tilde{\zeta}) = \frac{|\tilde{\zeta}_j| + 1 + |x_j|}{r^2} e^{-\sum_{k \neq j} (x_k - \tilde{\zeta}_k)^2 / 2r^2} e^{-\frac{-\tilde{\zeta}_j^2 + 2(|x_j| + 1)|\tilde{\zeta}_j|}{2r^2}}.$$

It is straightforward to use (31) and check that

$$\int \left| \eta_i(\tilde{\zeta}) \frac{1}{(2\pi)^{n/2} r^n} h(\tilde{\zeta}) \right| d\tilde{z} < \infty.$$

So far we have proved that the absolute value of the integrand in (33) is less than or equal to  $\frac{C}{(2\pi)^{n/2} r^n} h(\tilde{\zeta})$ , which is an integrable function. Hence, we can employ the dominated convergence theorem to show that

$$\begin{aligned} \frac{\partial \tilde{\eta}_i(x)}{\partial x_j} &= \lim_{\gamma \rightarrow 0} \frac{\tilde{\eta}_i(x + \gamma b^j) - \tilde{\eta}_i(x)}{\gamma} \\ &= \int_{\tilde{\zeta} \in \mathbb{R}^n} \frac{\eta_i(\tilde{\zeta})}{(2\pi)^{n/2} r^n} \lim_{\gamma \rightarrow 0} \left( \frac{e^{-\frac{\|x + \gamma b^j - \tilde{\zeta}\|_2^2}{2r^2}} - e^{-\frac{\|x - \tilde{\zeta}\|_2^2}{2r^2}}}{\gamma} \right) d\tilde{\zeta} \\ &= \int_{\tilde{\zeta} \in \mathbb{R}^n} \frac{\eta_i(\tilde{\zeta})}{(2\pi)^{n/2} r^n} \frac{\tilde{\zeta}_j - x_j}{r^2} e^{-\|x - \tilde{\zeta}\|_2^2 / 2r^2} d\tilde{\zeta}. \end{aligned} \quad (35)$$

It is straightforward to conclude that this derivative is bounded. Proving the continuity of the derivative employs the same lines of reasoning and hence we skip it.  $\square$

Calculating  $\tilde{\eta}(x)$  from  $\eta(x)$  is not straightforward for the following two reasons: (i) Equation (30) dictates that we integrate over all of  $\mathbb{R}^n$ , and (ii) We usually do not have access to the explicit form of  $\eta$ . To get around this problem we again turn to Monte Carlo sampling.

To approximately calculate (30) using Monte Carlo sampling first generate a series of  $M$  random vectors  $h^1, h^2, \dots, h^M$ , each with i.i.d. Gaussian elements with standard deviation  $r$ . Next, for each  $h^i$ , pass  $h^i + x$  through the discontinuous denoiser  $\eta(x)$  and then average the results to get a smooth denoiser  $\hat{\eta}(x)$ . That is approximate  $\tilde{\eta}(x)$  with

$$\hat{\eta}(x) = \frac{1}{M} \sum_{i=1}^M \eta(x + h^i), \quad (36)$$

where  $h^i \sim N(0, r^2 \mathbf{I})$  for all  $i$ .

Figure 11 compares the input-output relationship of the hard-thresholding denoiser before and after it has been smoothed using this method. Notice the smoothing process completely removes the discontinuities but otherwise leaves the function intact.

The above discussion does not provide any suggestion on how we should pick the smoothing parameter  $r$ . In fact, rigorous study of the effect of  $r$  in AMP requires the evaluation

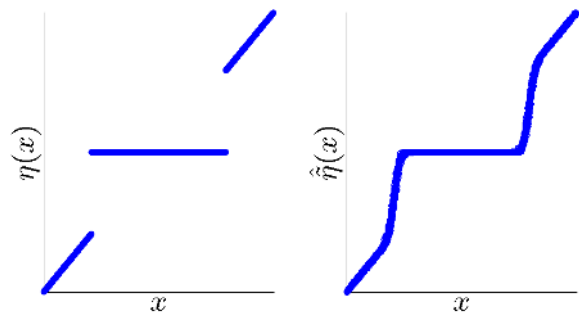


Fig. 11. Hard-thresholding and smoothed-hard-thresholding denoisers. Note how the smoothing process has removed the discontinuities.

of the difference  $\left| \frac{\|x^t - x_o\|_2^2}{N} - \theta^t(x_o, \delta, \sigma_w^2) \right|$ , in terms of the dimension. We leave it as an open problem for future research. Nevertheless, from a practical perspective  $r$  can be considered as just another denoiser parameter. The problem of optimizing denoiser parameters has been extensively studied in the field of image processing [64]–[66].

Figures 9 and 10 demonstrate the benefits of smoothing a denoiser using this approach. Unlike D-AMP using the original hard-thresholding denoiser, D-AMP using the smoothed denoiser closely follows the state evolution. This change is significant because it allows us to take advantage of the theory and tuning strategies developed in Section III. More importantly, Figure 10 illustrates how D-AMP based on smoothed-hard-thresholding dramatically outperforms its discontinuous counterpart.

Before proceeding, we would like to emphasize that the above process is *not* needed for any of the advanced denoisers that we explored in this paper. We found that advanced denoisers satisfy the state evolution and perform exceptionally in D-AMP without any smoothing. We believe this finding implies they are sufficiently smooth to begin with.

## VII. SIMULATION RESULTS FOR IMAGING APPLICATIONS

To demonstrate the efficacy of the D-AMP framework, we evaluate its performance on imaging applications.

### A. A menagerie of image denoising algorithms

As we have discussed so far, D-AMP employs a denoising algorithm for signal recovery problems. In this section, we briefly review some well-known image denoising algorithms that we would like to use in D-AMP. We later demonstrate that any of these denoisers, as well as many others, can be used within our D-AMP algorithm in order to reconstruct various compressively sampled signals. As we discussed in Section III-H, theory says that if denoising algorithm  $\mathcal{M}$  outperforms denoising algorithm  $\mathcal{N}$ , then D-AMP based on  $\mathcal{M}$  will outperform D-AMP based on  $\mathcal{N}$ . We will see this behavior in our simulations as well.

Below we represent a noisy image with the vector  $f$ ;  $f = x + \sigma z$  where  $x$  is the noise-free version of the image,  $\sigma$  is the

standard deviation of the noise, and the elements of  $z$  follow an i.i.d. Gaussian distribution.

- 1) Gaussian kernel regression: One of the simplest and oldest denoisers is Gaussian kernel regression, which is implemented via a Gaussian filter. As the name suggests, it simply applies a filter whose coefficients follow a Gaussian distribution to the noisy image. It takes the form:

$$\hat{x} = f \star G, \quad (37)$$

where  $G$  and  $\star$  denote the Gaussian kernel and the convolution operator, respectively. The Gaussian filter operates under the model that a signal is smooth. That is, neighboring pixels should have similar values. Note that this assumption is violated on image edges and hence this denoiser tends to over-smooth them. Compared to other approaches Gaussian kernel regression has a very low implementation cost. However, it does not remove noise as well as other denoisers.

- 2) Bilateral filter: Similar to kernel regression, the bilateral filter [73] sets each pixel value according to a weighted average of neighboring pixels. However, whereas the Gaussian filter computes weights based on how close to one another two pixels are, the bilateral filter computes weights based on the similarity of the pixel values (in addition to their spatial proximity). The estimate produced by the bilateral filter can be written as

$$\hat{x}(i) = \frac{\sum_{j \in \Omega_i} w(i, j) f(j)}{\sum_{j \in \Omega_i} w(i, j)} \quad (38)$$

$$w(i, j) = e^{-\frac{(f(i) - f(j))^2}{h^2}}, \quad (39)$$

where  $f(j)$  is the value of the  $j^{\text{th}}$  pixel,  $\Omega_i$  is a search window around pixel  $i$ , and  $h$  is a smoothing parameter set according to the amount of noise in the signal. Note that the bilateral filter tries to avoid averaging together light and dark pixels on opposite sides of an edge. The bilateral filter has generally proven much more effective than the Gaussian filter. However, it fails entirely when a very large amount of noise is present and the denoiser cannot determine which pixels should be alike.

- 3) Non-local means (NLM): Non-local means [39] extends the bilateral filter concept of averaging pixels with similar values to pixels with similar neighborhoods. NLM's original implementation takes the same form as the bilateral filter (38) but with the following weights:

$$w(i, j) = e^{-\frac{\|N(i) - N(j)\|_2^2}{h^2}}, \quad (40)$$

where  $N(i)$  represents a patch of pixels neighboring pixel  $i$  and  $h$  is a smoothing parameter set according to the variance of the noise. Because the true value of a pixel is better reflected by the noisy value of its neighborhood than by just its noisy pixel value, NLM better recognizes which pixels should be alike and thus outperforms the bilateral filter. However, because two pixels on opposite sides of an edge usually have very similar neighborhoods, NLM still produces artifacts around edges [74].

- 4) Wavelet thresholding: Wavelet thresholding [63] denoises natural images by assuming they are sparse in the wavelet domain. It transforms signals into a wavelet basis, thresholds the coefficients, and then inverses the transform. Hence if  $\Psi'$  and  $\Psi$  denote the wavelet transform and its inverse, respectively, then the denoised image is given by

$$\hat{x} = \Psi(\eta_\tau(\Psi' f)), \quad (41)$$

where  $\eta$  is some sort of thresholding function. The two most popular thresholding techniques are soft-thresholding  $\eta_\tau^s(x) = (|x| - \tau)_+ \text{sign}(x)$  and hard-thresholding  $\eta_\tau^h(x) = (x) \mathbb{I}(|x| \geq \tau)$ . Wavelet thresholding has superb performance if the signal is sparse in the wavelet domain. Unfortunately, images do not have an exactly sparse wavelet representation. As a result, the performance of wavelet thresholding denoising is generally worse than NLM.

- 5) BLS-GSM: Bayes least squares Gaussian scale mixtures [62] extends simple wavelet thresholding by using an overcomplete wavelet basis and computing denoised coefficient values not with a thresholding function, but via a Bayesian least squares estimate. This estimate is computed by considering a neighborhood around every coefficient and then modeling the distribution of the coefficients within that neighborhood as the product of a Gaussian random vector and a random scalar, each with a carefully defined prior. The algorithm uses these priors to compute the expected value of the noiseless coefficient value. Because the distributions of the wavelet coefficients of natural images are highly dependent on one another, a Bayesian least squares estimate can remove noise while retaining far more structure than coefficient thresholding alone. Accordingly, BLS-GSM significantly outperforms wavelet thresholding. Its performance relative to NLM depends on the statistics of the image being denoised.
- 6) BM3D: Block matching 3D collaborative filtering [56] can be considered a combination of NLM and wavelet thresholding. The algorithm begins by comparing patches around the pixels in an image and then grouping similar patches into stacks. It then performs 2D and 1D transforms on the group. These transforms are a 2D DCT and a 1D Haar transform or a 2D bi-orthogonal spline wavelet (Bior) and a 1D Haar transform. Which pair is used depends on the amount of noise in the image. Next the algorithm shrinks the coefficients of these groups and performs an inverse transform to estimate each pixel. It performs this process twice; once by hard-thresholding the coefficients and a second time using Wiener filtering based on the spectra of the initial estimate. In practice BM3D significantly outperforms NLM and wavelet thresholding techniques. It does a great job at removing noise and produces fewer artifacts than competing methods. Additionally, the authors of BM3D have provided well optimized code that makes this complicated algorithm quite efficient.
- 7) BM3D-SAPCA: BM3D with shape adaptive principal



component analysis [75] combines two extensions to the original BM3D algorithm; block matching using shape adaptive patches and thresholding/filtering in a PCA derived basis. Using adaptive patches helps ensure that the algorithm groups only similar patches. The use of an adaptive basis means that features not well captured by the DCT/Bior and Haar bases of BM3D will be retained. The performance of BM3D-SAPCA tends to be incrementally better than BM3D. Unfortunately, this small increase in performance comes at a huge increase in computational cost.

Table I provides a comparison among the above denoising algorithms. The parameters of the Gaussian filter, the bilateral filter, non-local means, and wavelet thresholding were all experimentally tuned so as to maximize PSNR.<sup>15</sup> The parameters for the other 3 algorithms were set automatically using their respective packages. The BM3D, BM3D-SAPCA, and BLS-GSM packages are available online.<sup>16 17</sup>

### B. Implementation details of D-AMP and D-IT

1) *Terminology*: Our goal is to plug each of the denoising algorithms that we reviewed in Section VII-A into our D-AMP algorithm. In the rest of the paper we use the following terminology: If denoising method  $\mathcal{M}$  is employed in D-AMP, then we call the reconstruction algorithm  $\mathcal{M}$ -AMP. For instance, if we use NLM, the resulting algorithm will be called NLM-AMP and if we use BM3D, the resulting algorithm will be called BM3D-AMP.

We use the same terminology for D-IT: If we use the BM3D denoiser then we call the resulting algorithm BM3D-IT.

2) *Denoising parameters*: One of the main challenges in comparing different recovery algorithms is the tuning of each algorithm's free parameters. As discussed in Section III-F, the parameters of D-AMP can be tuned efficiently with a greedy strategy. In other words, at every iteration we optimize the parameters to obtain the minimum MSE at that iteration. Toward this goal, we can employ different strategies that have been proposed in the literature for setting the parameters of denoising algorithms [64]–[66].

A variety of techniques exist to estimate the standard deviation of the noise in an image; however, we tackled this problem by using a convenient feature of AMP algorithms:  $\|z^t\|_2^2/m \approx (\sigma^t)^2$  [21]. Additionally, the packages provided with many of the state-of-the-art denoising algorithms [56], [62], [75], work with just two inputs; the noisy signal and an estimate of the standard deviation of the Gaussian noise. The packages then tune all other parameters internally so as to minimize the MSE. Thus, for the BM3D, BM3D-SAPCA, and BLS-GSM variants of D-AMP we use  $(\hat{\sigma}^t)^2 = \|z^t\|_2^2/m$  along with the packages and skip the parameter tuning problem.

<sup>15</sup>PSNR stands for peak signal-to-noise ratio and is defined as  $10 \log_{10}(\frac{255^2}{\text{mean}((\hat{x}-x_o)^2)})$  when the pixel range is 0 to 255. It is a measure of how closely a signal estimate  $\hat{x}$  is to the true signal  $x_o$ . In this paper we use PSNR to measure both the denoising algorithms' and CS recovery algorithms' rescaled MSE.

<sup>16</sup><http://www.cs.tut.fi/~foi/GCF-BM3D/>

<sup>17</sup><http://www.io.csic.es/PagsPers/JPortilla/software>

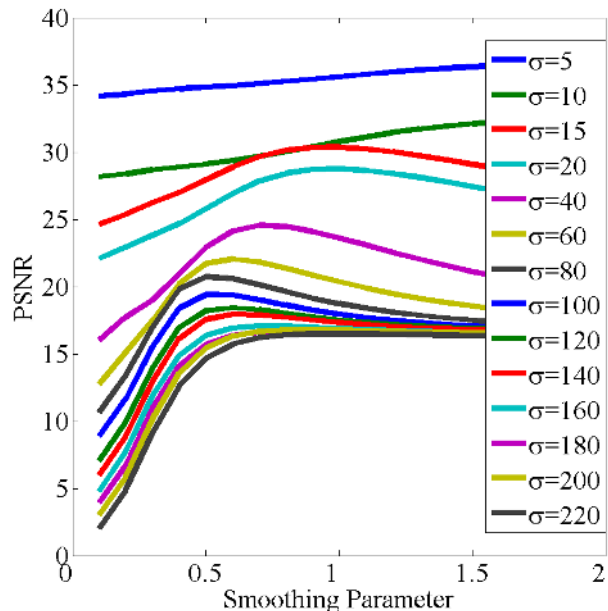


Fig. 12. PSNR of a NLM denoised image as a function of the smoothing parameter,  $h$  in (40) divided by the standard deviation of the noise,  $\sigma$ . The noisy images had been contaminated with AWGN with various standard deviations. Notice that different noise levels required different smoothing parameters. We used this data to create a look-up table used for parameter control within the NLM-AMP algorithm.

For denoisers without self-tuning packages, such as NLM, the tuning problem is challenging because at early iterations the effective noise has a large standard deviation but at later iterations the effective noise has a small standard deviation. This means the best parameters for early iterations are very different than the best parameters for later iterations. To get around this problem we use look-up-tables to set the parameters according to  $\hat{\sigma}^t$ . We naively generated these tables by first constructing artificial denoising problems with varying amounts of additive white Gaussian noise and then sweeping through the tuning parameters at each noise level. Figure 12 presents how we chose the parameter  $h$  used in NLM. At each noise level we simply chose the parameter values that maximized the PSNR of the denoising problem. For example, for NLM our look-up-table set  $h$  to .9 for  $\hat{\sigma}^t$  between 15 and 30. The same parameters were applied to all images; we did not optimize our code for individual images.

Recall that the state evolution comparison (Figure 7) showed that the MSE of BM3D-IT rose as the number of iterations increased. We attribute this to non-Gaussian effective noise and correct for this behavior by over-smoothing BM3D-IT at each iteration. The over-smoothing was set by using parameters optimized for  $2\hat{\sigma}$  rather than  $\hat{\sigma}$ . The scalar 2 was chosen as it provided the best MSE among the scalar values we tested.

3) *Stopping criterion*: AMP is typically designed to stop after some number of iterations or when  $\frac{\|x^t - x^{t-1}\|_2}{\|x^t\|_2}$  is less

TABLE I  
PERFORMANCE (PSNR IN dB) AND COMPUTATION TIME (SECONDS) COMPARISON OF SEVERAL DENOISERS. RESULTS ARE FOR  $256 \times 256$  IMAGES WITH ADDITIVE WHITE GAUSSIAN NOISE WITH STANDARD DEVIATION 15.

Denoiser	Lena	Barbara	Boat	Fingerprint	House	Peppers	Average Time
Gaussian filter	26.5	25.9	24.4	18.0	28.1	24.2	0.005
Bilateral filter	27.9	27.2	27.5	25.6	29.3	27.6	1.430
Non-local means	31.3	30.8	30.0	27.6	32.8	30.8	7.507
Wavelet thresholding	28.9	28.3	28.2	25.7	29.5	28.8	0.063
BLS-GSM	32.4	30.7	30.9	27.8	33.8	31.9	14.548
BM3D	33.2	32.4	31.2	28.3	35.1	32.6	1.128
BM3D-SAPCA	33.5	32.8	31.5	28.6	35.3	32.9	1251.633

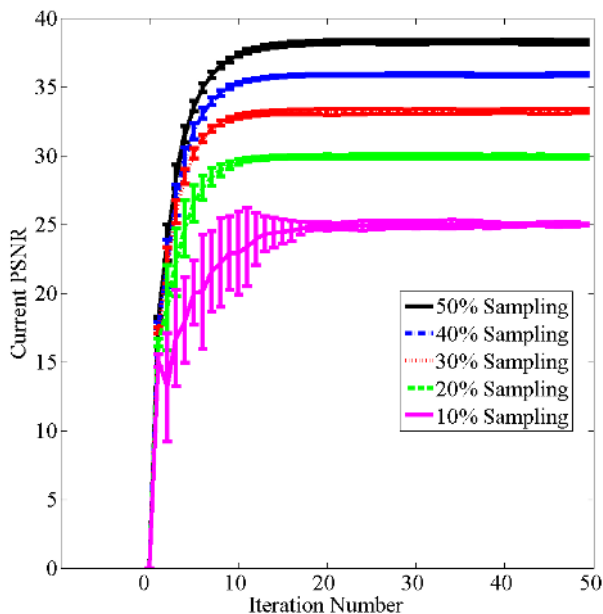


Fig. 13. The progression of the intermediate estimates' PSNRs on a  $128 \times 128$  Barbara test image over several iterations at different sampling rates. Notice that the estimates have high variance at first but generally stabilize by iteration 30.

than a threshold. Figure 13 demonstrates the PSNR evolution of BM3D-AMP (as a function of iterations) for different sampling rates of the  $128 \times 128$  Barbara test image. As the figure suggests, after about 10 iterations the PSNR has generally approached its maximum, but the variance of the estimates remains very high. After 30 iterations the variance is quite low. Therefore to reduce variation in our results, we decided to run BM3D-AMP for 30 iterations. The other D-AMP algorithms, as well as D-IT, IST, and AMP, exhibited similar behavior and were also run for 30 iterations.

4) *Onsager correction*: In all our implementations of D-AMP (except for the original AMP for which we used the closed form solution) we have used the Monte Carlo method for calculating the Onsager correction term, as reviewed in

Section V-B. While the algorithm seems to be insensitive to the exact value of  $\epsilon$  and works for a wide range of values of  $\epsilon$ , we used  $\epsilon = \frac{\|x\|_\infty}{1000}$ . We found this value was small enough for the approximation to be effective while not so small as to result in rounding errors. In the case of the original AMP, we have used the calculations we described in Section V-A.

### C. State evolution of D-AMP

Because the effective noise within D-AMP iterations is Gaussian, as further illustrated in Figure 14, state evolution serves as an effective predictor of D-AMP's performance. As the first step in our simulations, we would like to provide evidence of this prediction accuracy. To do so we compare the predicted and observed performance of D-AMP with NLM, wavelet thresholding, BLS-GSM, and BM3D.

Recall that the state evolution of D-AMP is defined by

$$\theta^{t+1}(x_o, \delta, \sigma_w^2) = \frac{1}{n} \mathbb{E} \|D_{\sigma^t}(x_o + \sigma^t \epsilon) - x_o\|_2^2,$$

where  $(\sigma^t)^2 = \frac{\theta^t}{\delta}(x_o, \delta, \sigma_w^2) + \sigma_w^2$ . To compute this value in practice, at every iteration we added white Gaussian noise with standard deviation  $\sigma^t$  to  $x_o$ , denoised the signal with denoiser  $D_{\sigma^t}$  (using the true, rather than estimated,  $\sigma^t$ ), and then computed the MSE.

Figure 15 displays the state evolutions alongside the true intermediate MSEs of the four aforementioned denoising-based algorithms when applied to a  $\delta = 0.4$  sampled  $128 \times 128$  House test image with no measurement noise. The average true MSEs at iteration 29 of AMP, NLM-AMP, BLS-GSM-AMP, and BM3D-AMP are all within 1.2% of the MSEs predicted by their respective state evolutions. We have posted our code online<sup>18</sup> to enable other researchers to verify and explore the validity of our state evolution predictions for these and other D-AMP algorithms.

### D. One-dimensional synthetic test

As a simple demonstration of the improvements that can be achieved by employing better denoising algorithms in AMP, we compare the performance of the original AMP (that employs sparsity in the wavelet domain) with the performance of NLM-AMP on a piecewise constant signal. Within the test

<sup>18</sup><http://dsp.rice.edu/software/DAMP-toolbox>

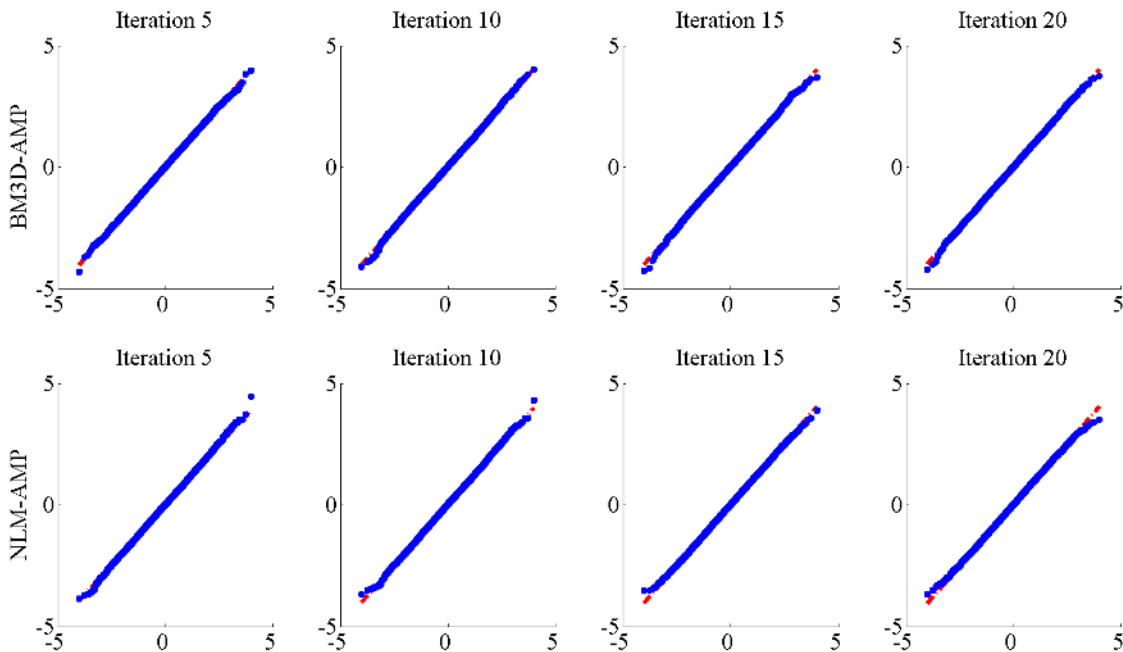


Fig. 14. QQplots of the effective noise at various iterations of BM3D-AMP and NLM-AMP. Notice that the effective noise remains Gaussian.

AMP used a Haar basis for wavelet thresholding and used the max-min optimal threshold as determined by [47]. The Haar basis was chosen because it well captures signal discontinuities. NLM-AMP used a length 11 patch,  $|N(i)| = 11$ , a length 21 search window,  $|\Omega_i|=21$ , and a smoothing parameter of 1.5,  $h = 1.5$ . These settings were chosen because they allow NLM to effectively denoise piecewise constant signals at a variety of noise levels. The results of our simulation are shown in Figure 3. As is clear from the figure, NLM-AMP significantly outperforms the original AMP. Even though the signal is relatively sparse in the wavelet domain, NLM captures its structure far more effectively. Hence NLM-AMP outperforms the standard AMP that employs sparsity in the wavelet domain.

### E. Imaging tests

1) *State-of-the-art recovery algorithms*: In this section we compare the performance of D-AMP, using a variety of denoisers, with other CS reconstruction algorithms. In particular, we compare the performance of our D-AMP algorithm with turbo-AMP [28]<sup>19</sup>, which is a hidden Markov tree model-based AMP algorithm, and ALSB [33]<sup>20</sup> and NLR-CS [31]<sup>21</sup>, which both utilize non-local group-sparsity. NLR-CS represents the current state-of-the-art in CS image reconstruction algorithms. We compare these 3 algorithms to D-AMP based on the NLM, BLS-GSM, BM3D, and BM3D-SAPCA denoisers. The performance of D-AMP using the Gaussian filter and the

bilateral filter was not competitive and has been omitted from the results. We include comparisons with AMP, using a wavelet basis. We also include comparisons with the BM3D-IT algorithm to illustrate the importance of the Onsager correction term in the performance of D-AMP. Other D-IT algorithms demonstrated considerably worse performance and are therefore omitted from the results.

2) *Test Settings*: ALSB uses rows drawn from a  $32^2 \times 32^2$  orthonormalized Gaussian measurement matrix to perform block-based compressed sensing, as described in [58]. All other tests used an  $m \times n$  measurement matrix that was generated by first using Matlab's `randn(m, n)` command and then normalizing the columns. All simulations were conducted on a 3.16 GHz Xeon quad-core processor with 32GB of memory.

For the AMP algorithm we used Daubechies 4 wavelets as the sparsifying basis and set its threshold optimally according to [47]. The parameters of D-AMP and D-IT were set following the methods described in section VII-B2. All D-IT and D-AMP algorithms were run for 30 iterations. AMP was run for 30 iterations as well. Turbo-AMP was run for 10 iterations. We experimented with running turbo-AMP for 30 iterations but found that this yielded no improvement in performance while nearly tripling the computation time. Because the DCT-sparsity-based iterative soft-thresholding method used to generate an initial estimate in NLR-CS's provided source code failed for Gaussian measurement matrices, we generated the initial estimates used by NLR-CS by running BM3D-AMP for 8 iterations for noiseless tests and 4 iterations for noisy tests. Only 4 iterations of BM3D-AMP

<sup>19</sup><http://www2.ece.ohio-state.edu/~schniter/turboAMPimaging/>

<sup>20</sup><http://idm.pku.edu.cn/staff/zhangjian/ALSB/>

<sup>21</sup>[http://see.xidian.edu.cn/faculty/wsdong/NLR\\_Exps.htm](http://see.xidian.edu.cn/faculty/wsdong/NLR_Exps.htm)

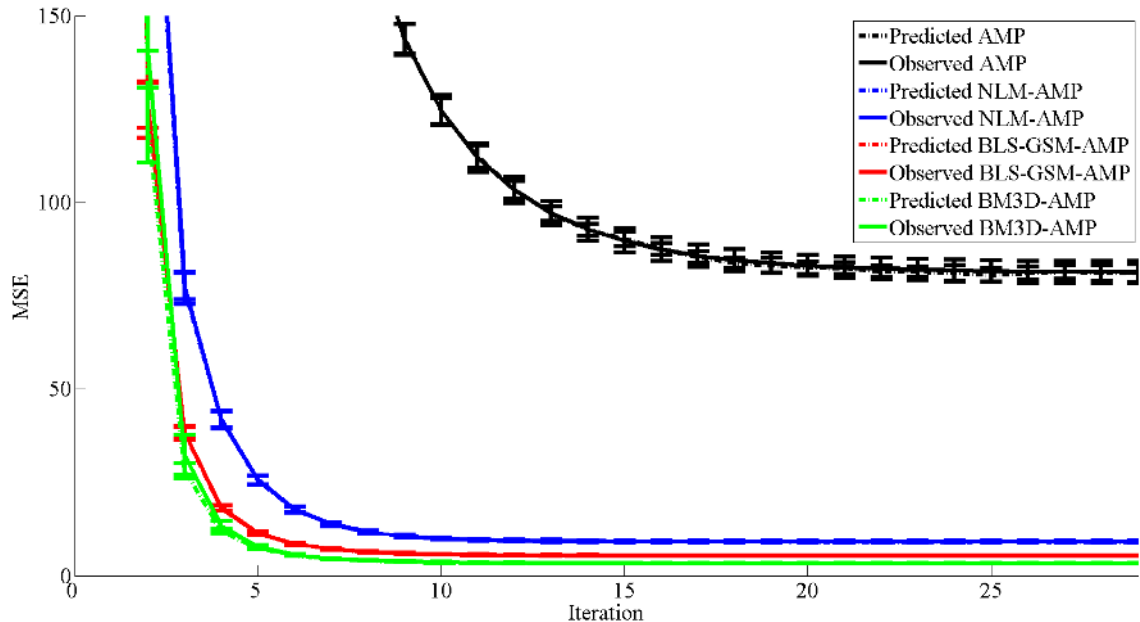


Fig. 15. State evolutions of multiple D-AMP algorithms when applied to a 40% sampled  $128 \times 128$  House test image with no measurement noise. There is near perfect correspondence between the predicted and true MSE.

were used during noisy tests because if run for 8 iterations the initial estimates from BM3D-AMP were often better than the final estimates from NLR-CS. Turbo-AMP, ALSB, and NLR-CS were otherwise tested under their default settings.

3) *Image database*: The data was generated using six standard image processing images drawn from Javier Portilla’s dataset:<sup>22</sup> Lena, Barbara, Boat, Fingerprint, House, and Peppers. The images each have a pixel range of roughly 0 – 255. Each of these images, except the examples presented in Figures 16 and 17, were rescaled to  $128 \times 128$  for testing. Restricting the tests to  $128 \times 128$  enabled the entire measurement matrix  $\mathbf{A}$  to be stored in memory. We also created a version of D-AMP that does not store  $\mathbf{A}$  but instead generates sections of  $\mathbf{A}$  as required. This version can handle images of arbitrarily large size but is extremely slow.

4) *Noiseless image recovery*: While matching the denoiser to the signal produces impressive results in one-dimensional settings (as summarized in Section VII-D), the results in 2D are even more pronounced. We begin this section with a visual comparison of three algorithms: Figure 16 illustrates the image recovery performance of AMP, NLR-CS, and our BM3D-SAPCA-AMP algorithm. BM3D-SAPCA-AMP outperformed NLR-CS slightly; 29.96 dB vs 29.31 dB. Both of these algorithms dramatically outperformed the wavelet sparsity-based AMP algorithm; 20.07 dB.

We also present a more complete comparison of D-AMP

with other algorithms in Table II.<sup>23</sup> As is clear from this table, BM3D-AMP or BM3D-SAPCA-AMP outperform all the other algorithms in a large majority of the tests. In the next section we demonstrate that the denoising based-AMP algorithms perform far better than competing methods when in the presence of measurement noise.

5) *Imaging in the presence of measurement noise*: In realistic settings compressive samples are subject to measurement noise. Noisy sampling can be modeled by  $y = \mathbf{A}x + w$  where  $w$  represents additive white Gaussian noise (AWGN). In Figure 17 we provide a visual comparison between the reconstructions of BM3D-SAPCA-AMP (26.86 dB) and NLR-CS (25.30 dB) in the presence of measurement noise. In Table III we compare the performance of the BM3D variant of D-AMP to NLR-CS and ALSB when varying amounts of measurement noise are present. As one might expect from a denoising-based algorithm, D-AMP was found to be exceptionally robust to noise. It outperformed the other methods in almost all tests and in some tests by as much as 7.4 dB.

6) *Computational complexity*: Table IV demonstrates that, depending on the denoiser in use, D-AMP can be quite

<sup>23</sup>Model-CoSaMP and other model-based techniques have not been included in our simulation results. First and foremost these methods were too slow for us to gather data before finishing the report. Additionally, we found they were not competitive: In the original Model-CoSaMP paper [24] the authors reported a RMSE of 11.1 (PSNR of 27.22 dB) from a reconstruction of a  $128 \times 128$  pepper test image using 5000 Gaussian measurements. By comparison, BM3D-AMP returns a RMSE of 5.1 (PSNR of 33.98 dB) on the same test.

<sup>22</sup><http://www.io.csic.es/PagsPers/JPortilla/software>



Fig. 16. Reconstructions of 10% sampled  $256 \times 256$  Barbara test image. The performance of BM3D-SAPCA-AMP is slightly better than the state-of-the-art NLR-CS algorithm and dramatically better than AMP.

efficient: The BM3D variant of D-AMP is dramatically faster than NLR-CS and ALSB. The table also illustrates how using different denoisers within D-AMP presents not only a means of capturing different signal models, but also a way to balance performance and run times.

### VIII. CONCLUSIONS

Through extensive testing we have demonstrated that the approximate message passing (AMP) compressed sensing recovery algorithm can be extended to use arbitrary denoisers to great effect. Variations of this denoising-based AMP algorithm (D-AMP) deliver state-of-the-art compressively sampled image recovery performance while maintaining a low computational footprint. Our theoretical results and simulations show that the performance of D-AMP can be predicted accurately by state evolution. We have also proven that the problem of tuning the parameters of D-AMP is no more difficult than the tuning of the denoiser that is used in the algorithm. Finally, we have shown that D-AMP is extremely robust to measurement noise. D-AMP represents a plug and play method

to recover compressively sampled signals of arbitrary class; simply choose a denoiser well matched to the signal model and plug it in the AMP framework. Since designing denoising algorithms that employ complicated structures is usually much easier than designing recovery algorithms, D-AMP can benefit many different application areas.

A significant amount of work remains to be done. First and foremost, all of the theory we developed for D-AMP relies upon the assumption that residual signals follow Gaussian distributions. In this paper we supported this assumption with state evolution and QQplot experiments. Theoretical validation of this assumption is left for future research. Likewise, all theory and results have been for i.i.d. Gaussian (or subGaussian) measurement matrices. Extension to other measurement matrices such as Fourier samples is another open direction that is left for future research.





Fig. 17. Reconstructions of 10% sampled  $256 \times 256$  Barbara test image with additive white Gaussian measurement noise with standard deviation 30. Note that BM3D-SAPCA-AMP exhibits far fewer artifacts than NLR-CS.

## APPENDIX

### A. Proof of Proposition 4

Let  $D_\sigma^M$  and  $D_\sigma^*$  denote the minimax denoiser, and minimax optimal family of denoisers for D-AMP, respectively. For notational simplicity we assume that  $\sup_{x_o} \mathbb{E} \|D_\sigma^M(x_o + \sigma\epsilon) - x_o\|_2^2$  is achieved at certain point  $x_o^{m,\sigma}$ , and that  $\sup_{x_o} \mathbb{E} \|D_\sigma^*(x_o + \sigma\epsilon) - x_o\|_2^2$  is achieved at certain point  $x_o^{*,\sigma}$ . Note that according to the state evolution for every  $x_o$  the fixed point of state evolution is given by

$$\theta_{DM}^\infty(x_o, \delta, \sigma_w^2) = \frac{1}{n} \mathbb{E} \|D_\sigma^M(x_o + \sigma\epsilon) - x_o\|_2^2,$$

where  $\sigma^2 = \frac{\theta_{DM}^\infty(x_o, \delta, \sigma_w^2)}{\delta} + \sigma_w^2$ . Define

$$\tilde{\theta}_{DM}^\infty(\delta, \sigma_w^2) = \sup_{x_o \in \{x_o^{m,\sigma} : \sigma > 0\}} \theta_{DM}^\infty(x_o, \delta, \sigma_w^2).$$

Again for notational simplicity assume that the supremum is achieved at  $x_o^{m,*}$ . The following lemma will be useful in our proof. It also has a nice interpretation that we describe after proving it.

**Lemma 5.** *If  $\theta_{DM}^\infty(x_o, \delta, \sigma_w^2)$  denotes the fixed point of the state evolution with denoiser  $D$  at signal  $x_o$ , then*

$$\theta_{DM}^\infty(x_o, \delta, \sigma_w^2) \leq \tilde{\theta}_{DM}^\infty(\delta, \sigma_w^2) = \theta_{DM}^\infty(x_o^{m,*}, \delta, \sigma_w^2).$$

*Proof.* We first claim that for every  $\theta > \tilde{\theta}_{DM}^\infty(\delta, \sigma_w^2)$  and for every  $x_o$  we have

$$\theta > \frac{1}{n} \mathbb{E} \|D_\sigma^M(x_o + \sigma\epsilon) - x_o\|_2^2, \quad (42)$$

where  $\sigma^2 = \frac{\theta}{\delta} + \sigma_w^2$ . Suppose that this is not true, i.e., there exists  $x_o$  and  $\theta > \theta_{DM}^\infty(\delta, \sigma_w^2)$  such that

$$\theta \leq \frac{1}{n} \mathbb{E} \|D_\sigma^M(x_o + \sigma\epsilon) - x_o\|_2^2.$$

Then

$$\begin{aligned} \theta &\leq \frac{1}{n} \mathbb{E} \|D_\sigma^M(x_o + \sigma\epsilon) - x_o\|_2^2 \\ &\leq \frac{1}{n} \mathbb{E} \|D_\sigma^M(x_o^{\sigma,M} + \sigma\epsilon) - x_o^{\sigma,M}\|_2^2, \end{aligned} \quad (43)$$

where the last inequality is due to the definition of  $x_o^{\sigma,M}$ . This implies that the fixed point of  $D_\sigma^M$  for vector  $x_o^{\sigma,M}$  will be larger than  $\theta$  and hence will be larger than  $\tilde{\theta}_{DM}^\infty(\delta, \sigma_w^2)$ . This is in contradiction with the definition of  $\tilde{\theta}_{DM}^\infty(\delta, \sigma_w^2)$ . Therefore, for any  $x_o$  (42) holds. Furthermore, (42) implies that for every  $x_o$  the fixed point of the state evolution of  $D_\sigma^M$  can only happen for  $\theta < \tilde{\theta}_{DM}^\infty(\delta, \sigma_w^2)$ . Hence establishes the result.  $\square$

This result has an interesting interpretation. The least favorable signal for D-AMP, i.e., the signal that leads to the highest fixed point, is one of the least favorable signals for the denoiser  $D_\sigma$ . While we proved this result for a specific denoiser  $D_\sigma^M$ , the proof can be easily extended to any denoiser  $D_\sigma$ .

We may now return to the proof of Proposition 4. Similar to Lemma 5 define

$$\tilde{\theta}_{D^*}^\infty(\delta, \sigma_w^2) = \sup_{x_o \in \{x_o^{*,\sigma} : \sigma > 0\}} \theta_{D^*}^\infty(x_o, \delta, \sigma_w^2).$$

Also, suppose that the supremum is achieved at  $x_o^{*,*}$ . Clearly, for any  $\theta > \tilde{\theta}_{D^*}^\infty(\delta, \sigma_w^2)$  we have

$$\begin{aligned} \theta &> \frac{1}{n} \mathbb{E} \|D_\sigma^*(x_o^{*,*} + \sigma\epsilon) - x_o^{*,*}\|_2^2 \\ &= \sup_{x_o} \frac{1}{n} \mathbb{E} \|D_\sigma^*(x_o + \sigma\epsilon) - x_o\|_2^2 \\ &\geq \inf_{D_\sigma} \sup_{x_o} \frac{1}{n} \mathbb{E} \|D_\sigma(x_o + \sigma\epsilon) - x_o\|_2^2 \\ &= \frac{1}{n} \mathbb{E} \|D_\sigma^M(x_o^{m,*} + \sigma\epsilon) - x_o^{m,*}\|_2^2. \end{aligned} \quad (44)$$

Hence the fixed point of  $D_\sigma^M$  is less than or equal to the fixed point of  $D_\sigma^*$ . Hence the proof is complete.

### B. Proof of Proposition 5

Let  $B_k$  denote the class of  $k$ -sparse signals with zero-one elements. Suppose that we have observed  $y = Ax_o$  ( $x_o \in B_k$ ) and the goal is to recover  $x_o$  from  $y$ . Consider the following recovery algorithm that is a special form of compressible signal pursuit proposed in [76], [77]:

$$\hat{x}_o = \arg \min_{x \in B_k} \|y - Ax\|_2^2.$$

**Lemma 6.** For  $m \geq 1$ ,

$$\mathbb{E}\|\hat{x}_o - x_o\|_2^2 = 0.$$

*Proof.* First note that  $\mathbb{P}(\hat{x}_o \neq x_o) = \mathbb{P}(A(x_o - \hat{x}_o) = 0) = 0$ . We can use the union bound and the fact that there are only  $\binom{n}{k}$  vectors in this space, to show that

$$\mathbb{P}(\exists x_o \in B_k : \hat{x}_o \neq x_o) = 0.$$

When the algorithm incorrectly estimates  $x_o$ ,  $\|\hat{x}_o - x_o\|_2^2$  is at most  $2k$ . Since the error is bounded our result is established.  $\square$

This is essentially the proof of the second part of the theorem. We now prove the first part of the theorem.

Consider the distribution  $\underline{\pi}_i^* = (1 - \frac{k}{n} + \gamma)\delta_0 + (\frac{k}{n} - \gamma)\delta_1$ , where  $\delta_a$  denotes a point mass at  $a$ . Construct a distribution on  $\mathbb{R}^n$  in the following way:

$$\underline{\pi}^* = \underline{\pi}_1^* \times \underline{\pi}_2^* \times \dots \times \underline{\pi}_n^*.$$

Here are the main steps of the proof:

- (i) We first prove that the samples we draw from  $\underline{\pi}^*$  belong to  $B_k$  with high probability.
- (ii) We employ the result of step one to derive a lower bound for the minimax risk.

Step (i) is a simple application of Hoeffding inequality. Let  $\underline{x}_o$  be a sample from this distribution. By using Hoeffding inequality we obtain:

$$\mathbb{P}\left(\left|\frac{1}{n}\|\underline{x}_o\|_0 - \frac{k}{n} - \gamma\right| < \gamma\right) \leq 2e^{-n\gamma^2/2}.$$

Therefore,

$$\mathbb{P}\left(\frac{1}{n}\|\underline{x}_o\|_0 < \frac{k}{n}\right) \leq 2e^{-n\gamma^2/2}.$$

In other words, with very high probability the samples that are generated from  $\underline{\pi}^*$  belong to  $B_k$ . Set  $\gamma = \frac{\sqrt{2}}{n^{1/4}}$ , define the event  $\mathcal{A}$  as  $\|\underline{x}_o\|_0 \leq k$  and let  $\pi^{**}$  denote the distribution of  $x_o$  conditioned on event  $\mathcal{A}$ . Note that the support of  $\pi^{**}$  is a subset of  $B_k$ . Now we can discuss step (ii), i.e., deriving a lower bound for minimax risk. Since the support of  $\pi^{**}$  is a subset of  $B_k$ , for every denoiser  $D_\sigma$  we have

$$\begin{aligned} & \mathbb{E}_{x_o \sim \pi^{**}} \mathbb{E}(\|D_\sigma(x_o + \sigma z) - x_o\|_2^2 \mid x_o) \\ & \leq \sup_{x_o \in B_k} \mathbb{E}(\|D_\sigma(x_o + \sigma z) - x_o\|_2^2 \mid x_o). \end{aligned} \quad (45)$$

By taking the infimum over  $D_\sigma$  from both sides, since the

optimal denoiser on the left is the Bayes denoiser, we obtain

$$\begin{aligned} & \mathbb{E}\|\mathbb{E}_{x_o \sim \pi^{**}}(x_o \mid x_o + \sigma z) - x_o\| \\ & \leq \inf_{D_\sigma} \sup_{x_o \in B_k} \mathbb{E}\|D_\sigma(x_o + \sigma z) - x_o\|_2^2. \end{aligned} \quad (46)$$

In other words we have derived a lower bound for the minimax risk based on  $\pi^{**}$ . Our next step is to calculate the lower bound we have on the left hand side. Note that  $\mathbb{P}(\mathcal{A}^c) = O(e^{-\sqrt{n}})$ , and

$$\mathbb{E}_{\pi^{**}}(x_o \mid y = x_o + \sigma z) = \mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z, \mathcal{A}). \quad (47)$$

Hence we have

$$\begin{aligned} & \mathbb{E}_{\pi^{**}}(x_o \mid y = x_o + \sigma z) \\ & = \frac{\mathbb{E}_{\pi^*}(x_o \mid x_o + \sigma z) - \mathbb{E}_{\pi^*}(x_o \mid x_o + \sigma z, \mathcal{A}^c)\mathbb{P}(\mathcal{A}^c)}{\mathbb{P}(\mathcal{A})} \end{aligned} \quad (48)$$

Define  $\phi(z_i) = \frac{1}{\sqrt{2\pi}}e^{-z_i^2/2}$  and  $\underline{\phi}(z) = \phi(z_1) \times \phi(z_2) \times \dots \times \phi(z_n)$ .

$$\begin{aligned} & \mathbb{E}_{x_o \sim \pi^{**}, z \sim \underline{\phi}} \|x_o - \mathbb{E}_{\pi^{**}}(x_o \mid y = x_o + \sigma z)\|_2^2 \\ & = \mathbb{E}_{x_o \sim \pi^{**}, z \sim \underline{\phi}} \left\| x_o - \frac{\mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z)}{1 - \mathbb{P}(\mathcal{A}^c)} \right\|_2^2 \\ & \quad + O(ne^{-\sqrt{n}}) \\ & = \mathbb{E}_{x_o \sim \pi^{**}, z \sim \underline{\phi}} \left\| \frac{x_o - \mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z)}{1 - \mathbb{P}(\mathcal{A}^c)} \right\|_2^2 \\ & \quad + O(ne^{-\sqrt{n}}) \\ & = \mathbb{E}_{x_o \sim \pi^{**}, z \sim \underline{\phi}} \|x_o - \mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z)\|_2^2 \\ & \quad + O(ne^{-\sqrt{n}}) \\ & = \mathbb{E}_{x_o \sim \pi^*, z \sim \underline{\phi}} \|x_o - \mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z)\|_2^2 \\ & \quad + O(ne^{-\sqrt{n}}). \end{aligned} \quad (49)$$

Define  $\phi_\sigma(z) = \phi(z/\sigma)$ ,  $\tilde{\gamma} = k/n - \gamma$  and  $\bar{\tilde{\gamma}} = 1 - k/n + \gamma$ . Note that since the prior we defined on  $x_o$ , i.e.,  $\pi^*$  is a product of similar measure on the individual  $x_{o,i}$  we conclude that

$$\begin{aligned} & \mathbb{E}_{x_o \sim \pi^*, z \sim \underline{\phi}} \|x_o - \mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z)\|_2^2 \\ & = n \mathbb{E}_{x_{o,1} \sim \pi_1^*, z_1 \sim \phi} (x_{o1} - \mathbb{E}_{\pi_1^*}(x_{o1} \mid y_1 = x_{o1} + \sigma z_1))^2 \\ & = \mathbb{E}_{z_1 \sim \phi} \left( \frac{(k/n - \gamma)\phi_\sigma(z_1)}{(\tilde{\gamma})\phi_\sigma(z_1) + (\bar{\tilde{\gamma}})\phi_\sigma(z_1 + 1)} - 1 \right)^2 (\tilde{\gamma}) \\ & \quad + \mathbb{E}_{z_1 \sim \phi} \left( \frac{(k/n - \gamma)\phi_\sigma(z_1 - 1)}{(\tilde{\gamma})\phi_\sigma(z_1 - 1) + (\bar{\tilde{\gamma}})\phi_\sigma(z_1)} \right)^2 (\bar{\tilde{\gamma}}). \end{aligned}$$

Finally, by the dominated convergence theorem we prove that

$$\begin{aligned} & \lim_{n \rightarrow \infty} \mathbb{E}_{x_o \sim \pi^*, z \sim \underline{\phi}} \|x_o - \mathbb{E}_{\pi^*}(x_o \mid y = x_o + \sigma z)\|_2^2 \\ & = \mathbb{E}_{z_1 \sim \phi} \left( \frac{\rho\phi_\sigma(z_1)}{\rho\phi_\sigma(z_1) + (1 - \rho)\phi_\sigma(z_1 + 1)} - 1 \right)^2 \rho \\ & \quad + \mathbb{E}_{z_1 \sim \phi} \left( \frac{(\rho)\phi_\sigma(z_1 - 1)}{\rho\phi_\sigma(z_1 - 1) + (1 - \rho)\phi_\sigma(z_1)} \right)^2 (1 - \rho). \end{aligned}$$

TABLE II  
PSNR OF  $128 \times 128$  RECONSTRUCTIONS WITH NO MEASUREMENT NOISE.

<b>10% Sampling</b>	<b>Lena</b>	<b>Barbara</b>	<b>Boat</b>	<b>Fingerprint</b>	<b>House</b>	<b>Peppers</b>
AMP	18.47	17.67	18.96	15.87	19.98	17.50
Turbo-AMP	18.35	17.46	18.62	16.30	21.77	17.01
ALSB	25.30	24.01	22.44	16.25	31.09	24.01
NLR-CS	<b>26.74</b>	<b>24.95</b>	23.97	18.11	<b>34.46</b>	<b>25.21</b>
BM3D-IT	5.68	5.97	5.43	4.70	4.93	5.72
NLM-AMP	21.81	20.17	21.43	17.69	24.81	20.42
BLS-GSM-AMP	24.92	23.35	23.98	17.53	30.52	24.09
BM3D-AMP	26.01	24.24	<b>24.07</b>	<b>18.24</b>	34.12	24.41
BM3D-SAPCA-AMP	15.04	24.28	22.62	18.17	32.74	23.99
<b>20% Sampling</b>	<b>Lena</b>	<b>Barbara</b>	<b>Boat</b>	<b>Fingerprint</b>	<b>House</b>	<b>Peppers</b>
AMP	21.26	20.08	21.62	16.86	22.97	20.27
Turbo-AMP	23.48	21.45	23.36	16.31	28.20	21.78
ALSB	28.66	27.98	26.09	17.42	36.28	28.12
NLR-CS	31.88	30.31	26.96	21.10	38.70	30.42
BM3D-IT	25.64	24.38	23.79	6.63	32.92	23.87
NLM-AMP	27.73	24.27	23.97	19.72	31.75	23.70
BLS-GSM-AMP	29.77	28.00	27.06	18.45	35.76	29.14
BM3D-AMP	31.12	29.83	<b>27.58</b>	21.14	38.30	30.00
BM3D-SAPCA-AMP	<b>32.15</b>	<b>30.41</b>	27.35	<b>22.02</b>	<b>38.94</b>	<b>31.09</b>
<b>30% Sampling</b>	<b>Lena</b>	<b>Barbara</b>	<b>Boat</b>	<b>Fingerprint</b>	<b>House</b>	<b>Peppers</b>
AMP	23.90	22.70	23.67	17.57	26.15	23.12
Turbo-AMP	25.88	24.35	24.80	16.33	32.18	24.58
ALSB	31.91	30.69	28.69	22.76	38.51	31.85
NLR-CS	35.86	33.78	30.27	23.01	41.15	34.80
BM3D-IT	28.16	27.21	24.43	18.44	35.48	25.49
NLM-AMP	29.94	29.39	27.67	20.81	36.49	29.92
BLS-GSM-AMP	33.29	31.06	29.95	19.20	39.17	32.73
BM3D-AMP	34.87	33.14	30.60	22.95	40.92	33.83
BM3D-SAPCA-AMP	<b>36.21</b>	<b>34.18</b>	<b>31.22</b>	<b>23.71</b>	<b>41.55</b>	<b>34.91</b>
<b>40% Sampling</b>	<b>Lena</b>	<b>Barbara</b>	<b>Boat</b>	<b>Fingerprint</b>	<b>House</b>	<b>Peppers</b>
AMP	26.35	24.77	25.41	18.65	29.20	25.36
Turbo-AMP	27.91	26.11	26.98	16.65	35.37	26.83
ALSB	34.17	34.19	30.92	24.14	41.13	35.15
NLR-CS	39.07	36.99	32.75	24.78	43.45	37.63
BM3D-IT	29.50	28.22	25.13	19.47	36.94	28.86
NLM-AMP	32.58	32.15	28.94	21.53	38.62	31.47
BLS-GSM-AMP	36.50	34.33	32.41	20.32	40.84	35.86
BM3D-AMP	38.05	35.94	32.77	24.59	42.97	36.77
BM3D-SAPCA-AMP	<b>39.33</b>	<b>37.05</b>	<b>33.56</b>	<b>25.01</b>	<b>43.86</b>	<b>38.06</b>
<b>50% Sampling</b>	<b>Lena</b>	<b>Barbara</b>	<b>Boat</b>	<b>Fingerprint</b>	<b>House</b>	<b>Peppers</b>
AMP	28.12	27.19	27.44	19.84	31.86	27.99
Turbo-AMP	30.64	27.69	28.80	19.24	37.54	29.17
ALSB	36.95	37.10	32.96	25.80	42.76	38.11
NLR-CS	42.05	<b>39.86</b>	35.31	26.26	45.65	40.51
BM3D-IT	30.95	29.18	27.14	20.24	38.19	29.56
NLM-AMP	35.09	34.72	31.45	25.34	39.71	34.10
BLS-GSM-AMP	38.92	36.42	34.72	21.61	42.34	38.72
BM3D-AMP	40.89	38.21	35.07	25.99	44.91	39.38
BM3D-SAPCA-AMP	<b>42.12</b>	39.49	<b>36.05</b>	<b>26.76</b>	<b>45.70</b>	<b>40.61</b>

TABLE III  
PSNR OF RECONSTRUCTION OF  $128 \times 128$  BARBARA TEST IMAGE WITH ADDITIVE WHITE GAUSSIAN MEASUREMENT NOISE WITH VARIOUS STANDARD DEVIATIONS (S.D.).

<b>AWGN with s.d. 10</b>					
<b>Sampling rate (%)</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>
ALSB	21.82	24.20	25.44	26.52	27.30
NLR-CS	24.29	27.84	28.85	29.24	28.48
BM3D-AMP	24.25	28.44	29.88	31.06	31.34
<b>AWGN with s.d. 20</b>					
<b>Sampling Rate (%)</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>
ALSB	19.32	20.83	21.40	22.15	22.72
NLR-CS	22.30	25.43	25.74	25.43	23.84
BM3D-AMP	23.79	26.65	27.54	28.18	28.24
<b>AWGN with s.d. 30</b>					
<b>Sampling Rate (%)</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>
ALSB	15.89	16.92	17.69	18.16	17.96
NLR-CS	21.90	22.49	22.05	20.46	18.38
BM3D-AMP	22.61	24.21	24.38	24.75	24.89
<b>AWGN with s.d. 40</b>					
<b>Sampling Rate (%)</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>
ALSB	15.89	16.92	17.69	18.16	17.96
NLR-CS	21.92	22.48	21.99	20.44	18.38
BM3D-AMP	22.65	24.22	24.61	24.88	25.06
<b>AWGN with s.d. 50</b>					
<b>Sampling Rate (%)</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>
ALSB	14.54	15.72	16.42	16.62	16.20
NLR-CS	21.02	21.49	20.66	18.77	16.56
BM3D-AMP	22.04	23.36	23.47	23.82	23.95

TABLE IV  
AVERAGE COMPUTATION TIMES, IN MINUTES, OF  $128 \times 128$  RECONSTRUCTIONS AT VARIOUS SAMPLING RATES.

<b>Sampling Rate (%)</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>
AMP	0.3	0.6	1.0	1.3	1.6
Turbo-AMP	1.4	2.4	3.4	4.5	5.5
ALSB	52.4	60.1	66.2	70.9	71.7
NLR-CS	31.6	60.6	88.1	122.8	152.2
BM3D-IT	0.8	1.2	1.4	1.7	2.0
NLM-AMP	11.3	6.7	4.4	4.2	3.8
BLS-GSM-AMP	5.1	5.0	5.3	5.6	5.9
BM3D-AMP	1.0	1.3	1.5	1.7	2.2
BM3D-SAPCA-AMP	318.3	328.7	345.1	362.1	378.0

## REFERENCES

- [1] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, pp. 489–509, Feb. 2006.
- [2] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, pp. 1182–1195, Dec. 2007.
- [3] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *Signal Processing Magazine*, vol. 25, pp. 83–91, Mar. 2008.
- [4] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, pp. 4203–4215, Dec. 2005.
- [5] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.
- [6] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415, Dec. 1993.
- [7] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2231–2242, Oct. 2004.
- [8] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, 2009.
- [9] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.
- [10] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci.*, vol. 106, no. 45, pp. 18914–18919, 2009.
- [11] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Select. Top. Signal Processing*, vol. 1, no. 4, pp. 586–598, 2007.
- [12] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. on Sci. Computing*, vol. 31, no. 2, pp. 890–912, 2008.
- [13] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. and Sim.*, vol. 4, no. 4, pp. 1168–1200, 2005.
- [14] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, "A wide-angle view at iterated shrinkage algorithms," *Proc. SPIE (Wavelet XII)*, Aug. 2007.
- [15] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing," *SIAM J. on Imag. Sci.*, vol. 1, no. 1, pp. 143–168, 2008.
- [16] I. Daubechies, M. DeFrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. on Pure and Applied Math.*, vol. 75, pp. 1412–1457, 2004.
- [17] A. Maleki and D. L. Donoho, "Optimally tuned iterative thresholding algorithm for compressed sensing," *IEEE J. Select. Top. Signal Processing*, Apr. 2010.
- [18] A. Y. Yang, Z. Zhou, A. Ganesh, S. S. Sastry, and Y. Ma, "Fast  $\ell_1$ -minimization algorithms for robust face recognition," *IEEE Trans. Image Processing*, vol. 22, pp. 3234–3246, Aug. 2013.
- [19] D. L. Donoho, A. Maleki, and A. Montanari, "Noise sensitivity phase transition," *IEEE Trans. Inform. Theory*, vol. 57, Oct. 2011.
- [20] A. Mousavi, A. Maleki, and R. G. Baraniuk, "Parameterless optimal approximate message passing," *arXiv preprint arXiv:1311.0035*, 2013.
- [21] A. Maleki, "Approximate message passing algorithm for compressed sensing," *Stanford University PhD Thesis*, Nov. 2010.
- [22] K. T. Block, M. Uecker, and J. Frahm, "Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint," *Magn. Reson. Med.*, vol. 57, pp. 1086–1098, Jun. 2007.
- [23] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Processing*, vol. 58, pp. 3042–3054, Jun. 2010.
- [24] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 56, pp. 1982–2001, Apr. 2010.
- [25] C. Hegde, P. Indyk, and L. Schmidt, "A fast approximation algorithm for tree-sparse recovery," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, pp. 1842–1846, IEEE, 2014.
- [26] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a hidden markov tree model," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, pp. 5137–5140, Mar. 2008.
- [27] Y. Kim, M. S. Nadar, and A. Bilgin, "Compressed sensing using a gaussian scale mixtures model in wavelet domain," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 3365–3368, Sept. 2010.
- [28] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a markov-tree prior," *IEEE Trans. Signal Processing*, vol. 60, no. 7, pp. 3439–3448, 2012.
- [29] G. Peyre, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," in *Computer Vision ECCV 2008* (D. Forsyth, P. Torr, and A. Zisserman, eds.), vol. 5304 of *Lecture Notes in Computer Science*, pp. 57–68, Springer Berlin Heidelberg, 2008.
- [30] J. Zhang, S. Liu, and D. Zhao, "Improved total variation based image compressive sensing recovery by nonlocal regularization," *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, pp. 2135–2138, Mar. 2013.
- [31] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *to appear in IEEE Trans. Image Processing*, 2014.
- [32] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, pp. 1193–1198, Nov. 2011.
- [33] J. Zhang, C. Zhao, D. Zhao, and W. Gao, "Image compressive sensing recovery using adaptively learned sparsifying basis via  $\ell_0$  minimization," *Signal Processing*, vol. 103, no. 0, pp. 114–126, 2014. *Image Restoration and Enhancement: Recent Advances and Applications*.
- [34] C. Metzler, A. Maleki, and R. Baraniuk, "From denoising to compressed sensing," *ELECS99 Project Report*, Apr. 2014.
- [35] D. L. Donoho, I. Johnstone, and A. Montanari, "Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising," *IEEE Trans. Inform. Theory*, vol. 59, no. 6, pp. 3396–3433, 2013.
- [36] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," *IEEE Trans. Inform. Theory*, vol. 59, pp. 4290–4308, Jul. 2013.
- [37] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 764–785, 2011.
- [38] U. Kamilov, S. Rangan, M. Unser, and A. K. Fletcher, "Approximate message passing with consistent parameter estimation and applications to sparse learning," in *Proc. Adv. in Neural Processing Systems (NIPS)*, pp. 2438–2446, 2012.
- [39] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Simul.*, vol. 4, pp. 490–530, 2005.
- [40] D. L. Donoho, A. Maleki, and A. Montanari, "Construction of message passing algorithms for compressed sensing," *Preprint*, 2010.
- [41] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. IEEE Conf. Inform. Science and Systems (CISS)*, Mar. 2010.
- [42] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. Allerton Conf. Communication, Control, and Computing*, pp. 347–353, 2010.
- [43] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE International Symposium Information Theory*, pp. 2168–2172, 2011.
- [44] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing," *arXiv preprint arXiv:1310.2632*, 2013.
- [45] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová, "Statistical-physics-based reconstruction in compressed sensing," *Physical Review X*, vol. 2, no. 2, p. 021005, 2012.
- [46] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," in *Proc. Int. Symposium Info. Theory (ISIT)*, pp. 1231–1235, 2012.
- [47] A. Maleki and A. Montanari, "Analysis of approximate message passing algorithm," in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pp. 1–7, Mar. 2010.
- [48] J. Barbier, F. Krzakala, M. Mézard, and L. Zdeborová, "Compressed sensing of approximately-sparse signals: Phase transitions and optimal reconstruction," in *Proc. Allerton Conf. Communication, Control, and Computing*, pp. 800–807, IEEE, 2012.
- [49] G. Reeves and D. Donoho, "The minimax noise sensitivity in compressed sensing," in *Proc. IEEE Int. Symp. Inform. Theory*, pp. 116–120, IEEE, 2013.
- [50] M. Bayati, M. A. Erdogdu, and A. Montanari, "Estimating Lasso risk and noise level," in *Proc. Adv. in Neural Processing Systems (NIPS)*, pp. 944–952, 2013.

- [51] M. Borgerding and P. Schniter, "Generalized approximate message passing for the cospars analysis model," *arXiv preprint arXiv:1312.3968*, 2013.
- [52] B. Cakmak, O. Winther, and B. H. Fleury, "S-AMP: Approximate message passing for general matrix ensembles," in *IEEE Inform. Theory Workshop (ITW)*, pp. 192–196, IEEE, 2014.
- [53] Y. Kabashima and M. Vehkaperä, "Signal recovery using expectation consistent approximation for linear observations," in *Proc. IEEE Int. Symp. Inform. Theory*, pp. 226–230, IEEE, 2014.
- [54] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with image denoising," *IEEE Trans. Inform. Theory*, vol. 63, no. 8, pp. 2085–2092, 2015.
- [55] K. Egiazarian, A. Foi, and V. Katkovnik, "Compressed sensing image reconstruction via recursive spatially adaptive filtering," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, pp. 1 – 549–1 – 552, Sept. 2007.
- [56] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Processing*, vol. 16, pp. 2080–2095, Aug. 2007.
- [57] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 3021–3024, 2009.
- [58] L. Gan, "Block compressed sensing of natural images," in *Proc. IEEE Int. Conf. on Dig. Signal Processing*, pp. 403–406, 2007.
- [59] J. E. Fowler, S. Mun, E. W. Tramel, M. R. Gupta, Y. Chen, T. Wiegand, and H. Schwarz, "Block-based compressed sensing of images and video," *Found. Trends in Signal Process*, vol. 4, no. 4, pp. 297–416, 2010.
- [60] A. Mousavi, A. Maleki, and R. G. Baraniuk, "Asymptotic analysis of Lassos solution path with implications for approximate message passing," *arXiv preprint arXiv:1309.5979*, 2013.
- [61] E. L. Lehmann and G. Casella, *Theory of point estimation*, vol. 31. Springer, 1998.
- [62] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. Image Processing*, vol. 12, pp. 1338–1351, Nov. 2003.
- [63] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.
- [64] X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE Trans. Image Processing*, vol. 19, no. 12, pp. 3116–3132, 2010.
- [65] N. P. Galatsanos and A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *IEEE Trans. Image Processing*, vol. 1, pp. 322–336, Jul. 1992.
- [66] S. Ramani, T. Blu, and M. Unser, "Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Image Processing*, pp. 1540–1554, 2008.
- [67] K. Kose, O. Gunay, and A. E. Cetin, "Compressive sensing using the modified entropy functional," *Digital Signal Processing*, vol. 24, pp. 63–70, 2014.
- [68] I. M. Johnstone, "Function estimation and gaussian sequence models," *Unpublished manuscript*, 2013.
- [69] L. Anitori, A. Maleki, M. Otten, R. G. Baraniuk, and W. van Rossum, "Compressive CFAR radar detection," in *Proc. IEEE Radar Conference (RADAR)*, pp. 0320–0325, May 2012.
- [70] E. J. Candes, C. A. Sing-Long, and J. D. Trzasko, "Unbiased risk estimates for singular value thresholding and spectral estimators," *IEEE Trans. Signal Processing*, vol. 61, no. 19, pp. 4643–4657, 2012.
- [71] C. Deledalle, G. Peyré, and J. Fadili, "Stein consistent risk estimator (score) for hard thresholding," *arXiv preprint arXiv:1301.5874*, 2013.
- [72] L. Zheng, A. Maleki, X. Wang, and T. Long, "Does  $\ell_p$ -minimization outperform  $\ell_1$ -minimization?," *arXiv preprint arXiv:1501.03704*, 2015.
- [73] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, pp. 839–846, IEEE, 1998.
- [74] A. Maleki, M. Narayan, and R. G. Baraniuk, "Suboptimality of nonlocal means for images with sharp edges," *Appl. Comput. Harmon. Anal.*, vol. 33, no. 3, pp. 370–387, 2012.
- [75] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," in *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS09)*, 2009.
- [76] S. Jalali and A. Maleki, "From compression to compressed sensing," in *Proc. IEEE Int. Symposium Info. Theory*, pp. 111–115, 2013.
- [77] S. Jalali, A. Maleki, and R. G. Baraniuk, "Minimum complexity pursuit for universal compressed sensing," *IEEE Trans. Inform. Theory*, vol. 60, pp. 2253–2268, Apr. 2014.

**Christopher A. Metzler** is a Ph.D. student in the Department of Electrical and Computer Engineering at Rice University. He received his M.S.E.E. and B.S.E.E. degrees from Rice in 2014 and 2013, respectively. He is currently supported by an NSF Graduate Research Fellowship and was previously supported by a DoD NDSEG fellowship. His current research focuses on the application of signal processing to imaging and communication systems.

**Arian Maleki** is an assistant professor in the Department of Statistics at Columbia University. He received Ph.D. from Stanford University in 2010. Before joining Columbia University, he was a postdoctoral scholar in the department of Electrical and Computer Engineering at Rice University.

**Richard G. Baraniuk** is the Victor E. Cameron Professor of Electrical and Computer Engineering at Rice University and the Founder and Director of OpenStax. His research interests lie in new theory, algorithms, and hardware for sensing, signal processing, and machine learning. He is a Fellow of the IEEE and AAAS and has received national young investigator awards from the US NSF and ONR, the Rosenbaum Fellowship from the Isaac Newton Institute of Cambridge University, the ECE Young Alumni Achievement Award from the University of Illinois, the Wavelet Pioneer and Compressive Sampling Pioneer Awards from SPIE, the IEEE Signal Processing Society Best Paper Award, and the IEEE Signal Processing Society Technical Achievement Award. His work on the Rice single-pixel compressive camera has been widely reported in the popular press and was selected by MIT Technology Review as a TR10 Top 10 Emerging Technology. For his teaching and education projects, including Connexions (cnx.org) and OpenStax College (openstaxcollege.org), he has received the C. Holmes MacDonald National Outstanding Teaching Award from Eta Kappa Nu, the Tech Museum of Innovation Laureate Award, the Internet Pioneer Award from the Berkman Center for Internet and Society at Harvard Law School, the World Technology Award for Education, the IEEE-SPS Education Award, the WISE Education Award, and the IEEE James H. Mulligan, Jr. Medal for Education. He is a Capricorn.