Matching Plug-and-Play Algorithms to the Denoiser

Saurav K. Shastri & **Philip Schniter***
Dept. ECE
Ohio State Univ., Columbus
{shastri.19,schniter.1}@osu.edu

Rizwan Ahmad
Dept. BME
Ohio State Univ., Columbus
rizwan.ahmad@osumc.edu

Christopher A. Metzler
Dept. CS
Univ. Maryland, College Park
metzler@umd.edu

Abstract

To solve inverse problems, plug-and-play (PnP) methods have been developed that replace the proximal step in a convex optimization algorithm with a call to an application-specific denoiser, often implemented using a deep neural network (DNN). Although such methods have been successful, they can be improved. For example, denoisers are usually designed/trained to remove white Gaussian noise, but the denoiser input error in PnP algorithms is usually far from white or Gaussian. Approximate message passing (AMP) methods provide white and Gaussian denoiser input error, but only when the forward operator is a large random matrix. In this work, we design a PnP method using the generalized expectation consistent (GEC) approximation algorithm that offers predictable error statistics at each iteration, from which DNN denoisers can be effectively trained.

1 Introduction

Linear inverse problems aim to recover a signal $\boldsymbol{x}_0 \in \mathbb{R}^N$ from measurements $\boldsymbol{y} \in \mathbb{R}^M$ of the form

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w}, \tag{1}$$

where $\boldsymbol{A}$ is a known linear operator and $\boldsymbol{w}$ is unknown noise. Well known examples of linear inverse problems include deblurring; superresolution; inpainting; image recovery in magnetic resonance imaging (MRI), computed tomography, or microscopy; and decoding in communications. In many cases, one has knowledge of the class of signals from which $\boldsymbol{x}_0$ is drawn. If harnessed, such knowledge can be extremely beneficial to signal recovery. In fact, in problems where $\boldsymbol{A}$ is not full column rank, such knowledge is essential because many $\boldsymbol{x}_0$ explain $\boldsymbol{y}$ even in the absence of noise.

The traditional approach to solving inverse problems relied on an expert-based model for the signal class (e.g., sparsity in a known basis) and used an algorithm to jointly enforce that model, together with (1). The primary challenge with this approach is that some signal classes (e.g., natural images) are difficult to model. A more recent approach is to train a deep neural network (DNN) to predict $\boldsymbol{x}_0$ from $\boldsymbol{y}$. Although this method has shown promise, it requires a huge number of $(\boldsymbol{x}_0, \boldsymbol{y})$ pairs for training, which may be unavailable in some applications. Also, the trained model may not generalize to test scenarios where $\boldsymbol{A}$ and/or the statistics of $\boldsymbol{w}$ have changed.

So-called "plug-and-play" (PnP) approaches give a middle-ground between traditional algorithmic approaches and end-to-end DNNs. There, a DNN is trained as a signal denoiser, and that denoiser is called iteratively by an algorithm that knows $\boldsymbol{A}$ and the statistics of $\boldsymbol{w}$. Convolutional DNN denoisers can be trained using only patches of the signal, so that relatively few full signal examples are required. Also, because the denoiser is trained independently of $\boldsymbol{A}$ and $\boldsymbol{w}$, generalization is not an issue.

*This work was funded in part by the National Institutes of Health under grants R01HL135489 and R01EB029957, and by the National Science Foundation under grant CCF-1955587.

NeurIPS 2021 Workshop on Deep Learning and Inverse Problems, virtual.

To understand the conventional approach to PnP algorithm design, it helps to first pose the inverse problem as an optimization problem of the form

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \big\{ g_1(\boldsymbol{x}) + g_2(\boldsymbol{x}) \big\}, \tag{2}$$

where $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ promote measurement and signal-class fidelity, respectively. For example, $g_1(\boldsymbol{x}) = \frac{\gamma_w}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ is an appropriate choice when $\boldsymbol{w}$ is white Gaussian noise (WGN) with precision γ_w. For convex g_1 and g_2, many algorithms have been proposed to solve (2). One of the best known is ADMM [1], shown as Alg. 1. In lines 3–4, γ is a positive stepsize that affects convergence speed but not the fixed point, and

Algorithm 1 ADMM

Require: $g_1(\cdot), g_2(\cdot)$ and γ.

1: Select initial $\boldsymbol{x}_2, \boldsymbol{u}$
2: **repeat**
3: $\quad \boldsymbol{x}_1 \leftarrow \mathrm{prox}_{\gamma^{-1} g_1}(\boldsymbol{x}_2 - \boldsymbol{u})$
4: $\quad \boldsymbol{x}_2 \leftarrow \mathrm{prox}_{\gamma^{-1} g_2}(\boldsymbol{x}_1 + \boldsymbol{u})$
5: $\quad \boldsymbol{u} \leftarrow \boldsymbol{u} + \boldsymbol{x}_1 - \boldsymbol{x}_2$
6: **until** Terminated

$$\mathrm{prox}_{\rho}(\boldsymbol{r}) \triangleq \arg\min_{\boldsymbol{x}} \big\{ \rho(\boldsymbol{x}) + \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{r}\|^2 \big\}. \tag{3}$$

Using (3), line 4 can be interpreted as MAP estimation of $\boldsymbol{x}_0$ with prior $p(\boldsymbol{x}_0) \propto e^{-g_2(\boldsymbol{x}_0)}$ from an observation $\boldsymbol{r} = \boldsymbol{x}_0 + \boldsymbol{e}$ with γ-precision AWGN $\boldsymbol{e}$, i.e., MAP denoising. This observation led Bouman et al. to propose that the prox in line 4 be replaced by state-of-the-art denoiser $\boldsymbol{f}_2 : \mathbb{R}^N \to \mathbb{R}^N$, giving rise to PnP-ADMM [2]. The PnP formula was then applied to other algorithms, such as primal-dual splitting (PDS) in [3]. Regularization-by-denoising (RED) [4, 5] yields a related class of algorithms that iteratively call a denoiser. See [6] for a comprehensive overview of PnP and RED.

With a well-designed DNN denoiser $\boldsymbol{f}_2$, PnP and RED significantly outperform optimization-based approaches, as well as end-to-end DNNs in limited-data and mismatched-$\boldsymbol{A}$ scenarios (see, e.g., [6]). However, there is room for improvement. For example, while DNN denoisers are typically trained to remove the effects of additive WGN (AWGN), PnP and RED algorithms do not provide the denoiser with an AWGN-corrupted input at each iteration. Thus, one might wonder: Is it possible to design an iterative denoising algorithm that presents the denoiser with AWGN at each iteration?

2 Approximate Message Passing

For Lipschitz denoisers $\boldsymbol{f}_2$, there is indeed a family of iterative denoising algorithms that presents the denoiser with WGN of known variance at each iteration: approximate message passing (AMP) algorithms like [7, 8]. In fact, when used with an MMSE denoiser $\boldsymbol{f}_2$, one can prove that these algorithm converge (exponentially fast) to the MMSE estimate of $\boldsymbol{x}_0$ from $\boldsymbol{y}$ [9, 10]. But there is a catch: for these algorithms to work as intended, the forward operator $\boldsymbol{A}$ should be large and random. For example, the original AMP algorithm [7] wants $\boldsymbol{A}$ to be large and i.i.d. sub-Gaussian, while the VAMP algorithm [8] imposes the looser requirement that $\boldsymbol{A}$ be large and right-orthogonally invariant (ROI), i.e., that the right singular-vector (RSV) matrix of $\boldsymbol{A}$ should be drawn uniformly from the set of orthogonal matrices. Essentially, these algorithms want that the RSV matrix of $\boldsymbol{A}$ randomly rotates the denoiser's output error at each iteration, rendering it statistically equivalent to WGN.

In many linear inverse problems, however, $\boldsymbol{A}$ does not have sufficient randomness for these AMP algorithms to work as intended. For example, in many imaging applications, $\boldsymbol{A}$ is constructed from a Fourier operator, but the denoiser output error also has significant Fourier structure, and so the desired randomization does not happen. As a result, when these AMP algorithms are used with Fourier-structured $\boldsymbol{A}$ and $\boldsymbol{x}_0$, they tend to diverge. Although heuristic damping and initialization strategies can be applied to stabilize VAMP in these applications, and even outperform PnP-ADMM [11], they do not yield denoiser input errors that are white and Gaussian at each iteration.

That said, progress in VAMP algorithm design with Fourier-structured $\boldsymbol{A}$ and natural-image $\boldsymbol{x}_0$ has been made in [12–14]. These works all recover the wavelet coefficients of the image rather than the image pixels. The motivation is that the Fourier-Wavelet matrix is known to be approximately block diagonal [15], where the blocks are formed by the wavelet subbands. Furthermore, the subblocks of the Fourier-Wavelet matrix behave as randomizing operators for the wavelet coefficients in each wavelet subband. This gives the potential to design a VAMP-like algorithm with WGN error in each subband, as long as differences across subbands are properly handled.

In the first of these works, Schniter, Rangan, and Fletcher [12] used VAMP to recover the *whitened* wavelet coefficients in an attempt to equalize the subband error variances, but did not track the error

variances over the iterations. Later, for the special case of variable-density (VD) subsampled Fourier measurements, Millard et al. [13] proposed a VAMP modification called VDAMP that uses wavelet thresholding to track the subband error variances across iterations. More recently, Metzler and Wetzstein [14] extended VDAMP to generic denoising, and designed a DNN denoiser that exploits known error-variance within each wavelet subband. The resulting technique, coined "denoising-VDAMP" (D-VDAMP), yielded subband errors that were white and Gaussian with predictable variance at each iteration, and showed a significant boost in MRI image recovery performance over VDAMP, PnP-ADMM, and RED [14].

Although D-VDAMP works reasonably well, it is based on a heuristic modification of VAMP with poor fixed points. Also, it is not clear how to use D-VDAMP with forward operators other than VD-subsampled Fourier. These issues motivate our proposed approach, which is described next.

3 Proposed Approach

Our approach uses the GEC framework from [16], as summarized in Alg. 2. GEC is similar to VAMP, but it averages the diagonal of the Jacobian $\nabla \boldsymbol{f}_i$ separately over L coefficient subsets using $\mathrm{gdiag}\colon \mathbb{R}^{N\times N} \to \mathbb{R}^N$:

$$\mathrm{gdiag}(\boldsymbol{Q}) \triangleq [d_1 \mathbf{1}_{N_1}^{\mathsf{T}}, \ldots, d_L \mathbf{1}_{N_L}^{\mathsf{T}}]^{\mathsf{T}} \quad (4a)$$

$$d_\ell = \frac{\mathrm{tr}\{\boldsymbol{Q}_{\ell\ell}\}}{N_\ell}. \quad (4b)$$

In (4), N_ℓ is the size of the ℓth subset and $\boldsymbol{Q}_{\ell\ell} \in \mathbb{R}^{N_\ell \times N_\ell}$ is the ℓth diagonal subblock of the matrix input $\boldsymbol{Q}$. In Alg. 2, $\mathrm{Diag}(\cdot)$ creates a diagonal matrix from its vector argument. When $L=1$, GEC reduces to VAMP.

Algorithm 2 Generalized EC (GEC)

Require: $\boldsymbol{f}_1(\cdot,\cdot), \boldsymbol{f}_2(\cdot,\cdot),$ and $\mathrm{gdiag}(\cdot)$.

Select initial $\boldsymbol{r}_1, \boldsymbol{\gamma}_1$
repeat
  // Measurement fidelity
  $\widehat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{f}_1(\boldsymbol{r}_1, \boldsymbol{\gamma}_1)$
  $\boldsymbol{\eta}_1 \leftarrow \mathrm{Diag}(\mathrm{gdiag}(\nabla \boldsymbol{f}_1(\boldsymbol{r}_1, \boldsymbol{\gamma}_1)))^{-1}\boldsymbol{\gamma}_1$
  $\boldsymbol{\gamma}_2 \leftarrow \boldsymbol{\eta}_1 - \boldsymbol{\gamma}_1$
  $\boldsymbol{r}_2 \leftarrow \mathrm{Diag}(\boldsymbol{\gamma}_2)^{-1}(\mathrm{Diag}(\boldsymbol{\eta}_1)\widehat{\boldsymbol{x}}_1 - \mathrm{Diag}(\boldsymbol{\gamma}_1)\boldsymbol{r}_1)$
  // Denoising
  $\widehat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{f}_2(\boldsymbol{r}_2, \boldsymbol{\gamma}_2)$
  $\boldsymbol{\eta}_2 \leftarrow \mathrm{Diag}(\mathrm{gdiag}(\nabla \boldsymbol{f}_2(\boldsymbol{r}_2, \boldsymbol{\gamma}_2)))^{-1}\boldsymbol{\gamma}_2$
  $\boldsymbol{\gamma}_1 \leftarrow \boldsymbol{\eta}_2 - \boldsymbol{\gamma}_2$
  $\boldsymbol{r}_1 \leftarrow \mathrm{Diag}(\boldsymbol{\gamma}_1)^{-1}(\mathrm{Diag}(\boldsymbol{\eta}_2)\widehat{\boldsymbol{x}}_2 - \mathrm{Diag}(\boldsymbol{\gamma}_2)\boldsymbol{r}_2)$
until Terminated

When GEC is used to solve a convex optimization problem of the form (2), the functions $\boldsymbol{f}_i$ take the form

$$\boldsymbol{f}_i(\boldsymbol{r}, \boldsymbol{\gamma}) = \mathrm{gprox}_{g_i,\boldsymbol{\gamma}}(\boldsymbol{r}) \quad \text{for} \quad \mathrm{gprox}_{\rho,\boldsymbol{\gamma}}(\boldsymbol{r}) \triangleq \arg\min_{\boldsymbol{x}} \big\{\rho(\boldsymbol{x}) + \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{r}\|_{\boldsymbol{\gamma}}^2\big\}, \quad (5)$$

where $\|\boldsymbol{q}\|_{\boldsymbol{\gamma}} \triangleq \sqrt{\boldsymbol{q}^{\mathsf{H}}\mathrm{Diag}(\boldsymbol{\gamma})\boldsymbol{q}}$. Note that if $\boldsymbol{\gamma} = \gamma\mathbf{1}$, then $\mathrm{gprox}_{\rho,\boldsymbol{\gamma}} = \mathrm{prox}_{\gamma^{-1}\rho}$. Furthermore, if the $\boldsymbol{\gamma}_i$ vectors are held fixed over the iterations and take the form $\boldsymbol{\gamma}_i = \gamma\mathbf{1}$ for some $\gamma > 0$, then GEC reduces to the "Peaceman Rachford" variant of ADMM, which uses two dual updates. Thus, GEC can be interpreted an ADMM-like algorithm with adaptive vector-valued stepsizes, $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$.

For our proposed "denoising GEC" (D-GEC) approach, we use a DNN denoiser for $\boldsymbol{f}_2$. Also, we focus on WGN $\boldsymbol{w}$ (i.e., quadratic log-likelihood), which gives

$$\boldsymbol{f}_1(\boldsymbol{r}, \boldsymbol{\gamma}) = \big(\gamma_w \boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} + \mathrm{Diag}(\boldsymbol{\gamma})\big)^{-1}\big(\gamma_w \boldsymbol{A}^{\mathsf{H}}\boldsymbol{y} + \mathrm{Diag}(\boldsymbol{\gamma})\boldsymbol{r}\big). \quad (6)$$

For both $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, we approximate the $\mathrm{tr}\{\boldsymbol{Q}_{\ell\ell}\}$ term in (4) using the Monte Carlo approach [17]

$$\mathrm{tr}\{\boldsymbol{Q}_{\ell\ell}\} \approx \delta^{-1}\boldsymbol{q}_\ell^{\mathsf{H}}\big[\boldsymbol{f}_i(\boldsymbol{r} + \delta\boldsymbol{q}_\ell, \boldsymbol{\gamma}) - \boldsymbol{f}_i(\boldsymbol{r}, \boldsymbol{\gamma})\big], \quad (7)$$

where the coefficients in the ℓth subset of $\boldsymbol{q}_\ell$ are i.i.d. unit-variance Gaussian and the others are zero.

For Fourier-structured $\boldsymbol{A}$, we recover the wavelet coefficients $\boldsymbol{c}_0 = \boldsymbol{\Psi}\boldsymbol{x}_0$ of $\boldsymbol{x}_0$, like in [12–14]. For an orthogonal wavelet transform $\boldsymbol{\Psi}$, we have that $\boldsymbol{x}_0 = \boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{c}_0$, and so (1) implies the model

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}_0 + \boldsymbol{w} \;\text{ with }\; \boldsymbol{B} \triangleq \boldsymbol{A}\boldsymbol{\Psi}^{\mathsf{T}}. \quad (8)$$

To apply D-GEC to $\boldsymbol{c}_0$ recovery, we use $\boldsymbol{f}_1$ as in (6), but with $\boldsymbol{B}$ in place of $\boldsymbol{A}$. For the diagonalization subsets in (4), we choose the L subbands of a depth-D wavelet transform.

As in [14], we denoise the wavelet coefficient estimate $\boldsymbol{r}_2$ by converting to the pixel domain, denoising in that domain, and converting back to the wavelet domain. In the pixel-domain, GEC predicts that the noise will be correlated with correlation matrix $\boldsymbol{\Psi}\,\mathrm{Diag}(\boldsymbol{\gamma})^{-1}\boldsymbol{\Psi}^{\mathsf{T}}$. To handle this correlated noise, we propose a new approach to denoising called "corr+corr," which presents the (pixel-domain) denoiser with the noisy image in one channel and an independent noise *realization* in another. During training, the denoiser learns to extract statistics from this realization and use them productively for denoising.

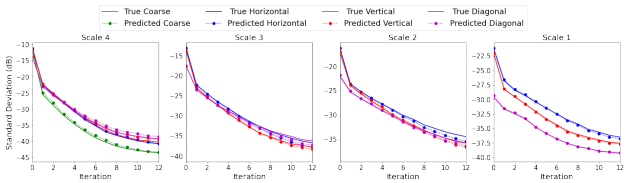


Figure 1: SD of D-GEC's denoiser input versus iteration.

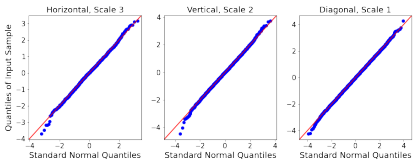


Figure 2: Wavelet coefficient QQ-plots.

4 Numerical Experiments

We compare the proposed D-GEC algorithm to the existing D-VDAMP [14] and PnP-PDS [3] algorithms for MRI image recovery. For C-coil MRI, we constructed $\boldsymbol{A} = [\boldsymbol{A}_1^{\mathsf{T}}, \ldots, \boldsymbol{A}_C^{\mathsf{T}}]^{\mathsf{T}}$ with $\boldsymbol{A}_c = \boldsymbol{M}\boldsymbol{F}\,\mathrm{Diag}(\boldsymbol{s}_c) \in \mathbb{C}^{M\times N}$, where $\boldsymbol{F}$ is the unitary 2D Fourier matrix, sampling mask $\boldsymbol{M}$ contains M rows of $\boldsymbol{I}_N$, and $\boldsymbol{s}_c \in \mathbb{C}^N$ is a coil sensitivity map generated from the Biot-Savart Law. In the single-coil case, $C=1$ and $\boldsymbol{s}_1 = \boldsymbol{1}$. Like in [14], we used VD masks (see Fig. 4 of Appendix).

Denoisers: For D-VDAMP, we used the denoiser proposed in [14], which is a modification of DnCNN [18] that accepts the noise standard deviation (SD) in each wavelet subband. Five versions of this denoiser were trained using independent subband noise SDs uniformly distributed in the ranges 0-10, 10-20, 20-50, 50-120, and 120-500, respectively. (Pixel values ranged from 0-255.) For PnP-PDS, we trained bias-free DnCNN [19] with white noise of SD uniformly distributed in the range 0-55. For D-GEC, we trained bias-free DnCNN using the previously described corr+corr approach using independent subband SDs uniformly distributed in the ranges 0-10, 10-20, 20-50, 50-120, and 120-500, respectively. All DNNs were trained using patches from 70 MRI images of the Stanford 2D FSE dataset available at `http://mridata.org`. All denoisers use only the real part of the input and generate a real-valued output.

Test data: For evaluation, we used ten 352×352 MRI images from the Stanford 2D FSE dataset that were not used as training images (see Fig. 3 of Appendix). The measurements $\boldsymbol{y}$ were constructed using (1) with complex AWGN $\boldsymbol{w}$ whose variance was adjusted to give a pre-masking SNR of 40 dB.

Algorithm parameters: For D-GEC and D-VDAMP, we used a 2D Haar wavelet transform with $D=4$ levels, giving $L=13$ wavelet subbands. D-VDAMP code was obtained from the authors and run under default settings, which are detailed in [14]. PnP-PDS was run for 200 iterations using the parameters that maximized PSNR on the training set. For D-GEC, we used the damping scheme from [11] and the auto-tuning scheme from [20].

Single-coil results: Table 1 shows that D-GEC significantly outperformed D-VDAMP and PnP-PDS in all experiments. Figure 1 shows the evolution of the SD of the error at the input to D-GEC's denoiser in each subband. There, we see a close agreement between true and predicted values. Figure 2 shows QQ plots that indicate the subband errors are Gaussian. In the Appendix, Fig. 5 shows an example of the wavelet coefficients input to D-GEC's denoiser at the 10th iteration, and their error relative to the true coefficients, while Fig. 6 shows image recoveries and error maps for one of the test images at $M/N=1/4$.

	$C=1$ coil				$C=4$ coils			
	$M/N=1/4$		$M/N=1/8$		$M/N=1/4$		$M/N=1/8$	
method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
PnP-PDS	45.97	0.978	41.28	0.957	47.98	0.992	43.81	0.977
D-VDAMP	44.61	0.974	38.43	0.901	n/a	n/a	n/a	n/a
D-GEC	**47.64**	**0.982**	**42.42**	**0.959**	**50.80**	**0.997**	**46.67**	**0.991**

Table 1: Recovery results averaged over the ten test images.

Multi-coil results: Table 1 shows D-GEC significantly outperforming PnP-PDS in PSNR and SSIM in the 4-coil case. D-VDAMP does not support multi-coil MRI and thus is not shown in the table.

5 Conclusion

We designed a GEC-based PnP algorithm that aims to present the denoiser with subband errors that are white and Gaussian with known variance at each iteration, and a denoiser that handles the resulting noise, which has a known correlation structure in the pixel domain. Experiments with MRI image recovery show significantly improved recovery performance relative to previous PnP methods.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[2] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Info. Process.*, 2013, pp. 945–948.

[3] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, 2017.

[4] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[5] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comp. Imag.*, vol. 5, no. 1, pp. 52–67, Mar. 2019.

[6] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, "Plug and play methods for magnetic resonance imaging," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 105–116, 2020.

[7] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[8] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inform. Theory*, pp. 6664–6684, 2019.

[9] R. Berthier, A. Montanari, and P.-M. Nguyen, "State evolution for approximate message passing with non-separable functions," *Inform. Inference*, 2019.

[10] A. K. Fletcher, P. Pandit, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Neural Inform. Process. Syst. Conf.*, 2018, pp. 7440–7449.

[11] S. Sarkar, R. Ahmad, and P. Schniter, "MRI image recovery using damped denoising vector AMP," in *Proc. IEEE Int. Conf. Acoust. Speech & Signal Process.*, 2021, pp. 8108–8112.

[12] P. Schniter, S. Rangan, and A. K. Fletcher, "Plug-and-play image recovery using vector AMP," presented at the Intl. Biomedical and Astronomical Signal Processing (BASP) Frontiers Workshop, Villars-sur-Ollon, Switzerland, Jan. 2017. [Online]. Available: http://www2.ece.ohio-state.edu/~schniter/pdf/basp17_poster.pdf

[13] C. Millard, A. T. Hess, B. Mailhé, and J. Tanner, "Approximate message passing with a colored aliasing model for variable density Fourier sampled images," *arXiv:2003.02701*, 2020.

[14] C. A. Metzler and G. Wetzstein, "D-VDAMP: Denoising-based approximate message passing for compressive MRI," in *Proc. IEEE Int. Conf. Acoust. Speech & Signal Process.*, 2021, pp. 1410–1414.

[15] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, "Breaking the coherence barrier: A new theory for compressed sensing," *Forum of Mathematics, Sigma*, vol. 5, no. E4, doi:10.1017/fms.2016.32.

[16] A. K. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Expectation consistent approximate inference: Generalizations and convergence," in *Proc. IEEE Int. Symp. Inform. Thy.*, 2016, pp. 190–194.

[17] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, 2008.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.

[19] S. Mohan, Z. Kadkhodaie, E. P. Simoncelli, and C. Fernandez-Granda, "Robust and interpretable blind image denoising via bias-free convolutional neural networks," in *Proc. Internat. Conf. on Learning Repres.*, 2020.

[20] A. K. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Rigorous dynamics and consistent estimation in arbitrarily conditioned linear systems," in *Proc. Neural Inform. Process. Syst. Conf.*, 2017, pp. 2542–2551.

6 Appendix

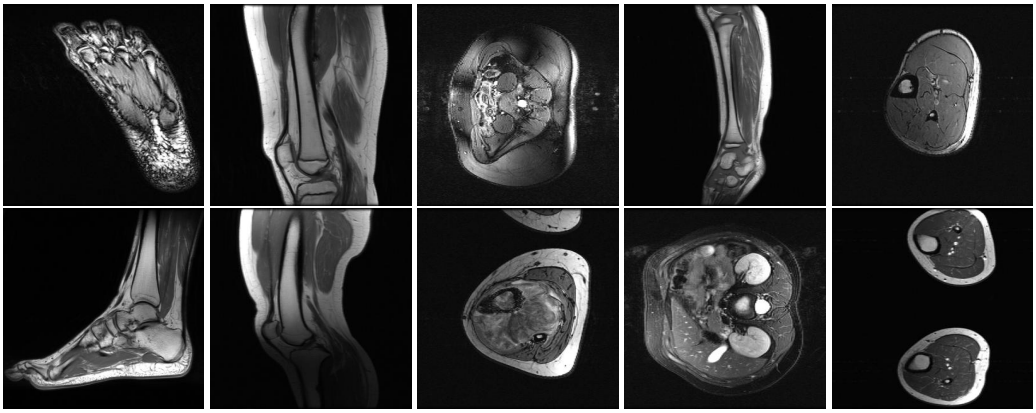


Figure 3: Test images from Stanford 2D FSE dataset at `http://mridata.org`.

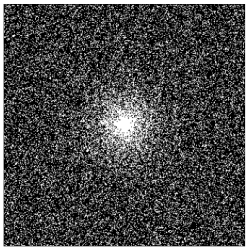
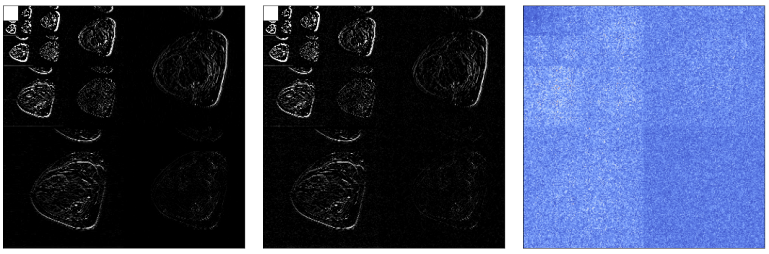


Figure 4: A variable-density sampling mask at $M/N = 1/4$.



$\boldsymbol{c}_0$ $\boldsymbol{r}_2$ $|\boldsymbol{r}_2 - \boldsymbol{c}_0|$

Figure 5: An example of the true coefficients $\boldsymbol{c}_0$, denoiser input $\boldsymbol{r}_2$, and denoiser input error $\boldsymbol{r}_2 - \boldsymbol{c}_0$ at D-GEC iteration 10.

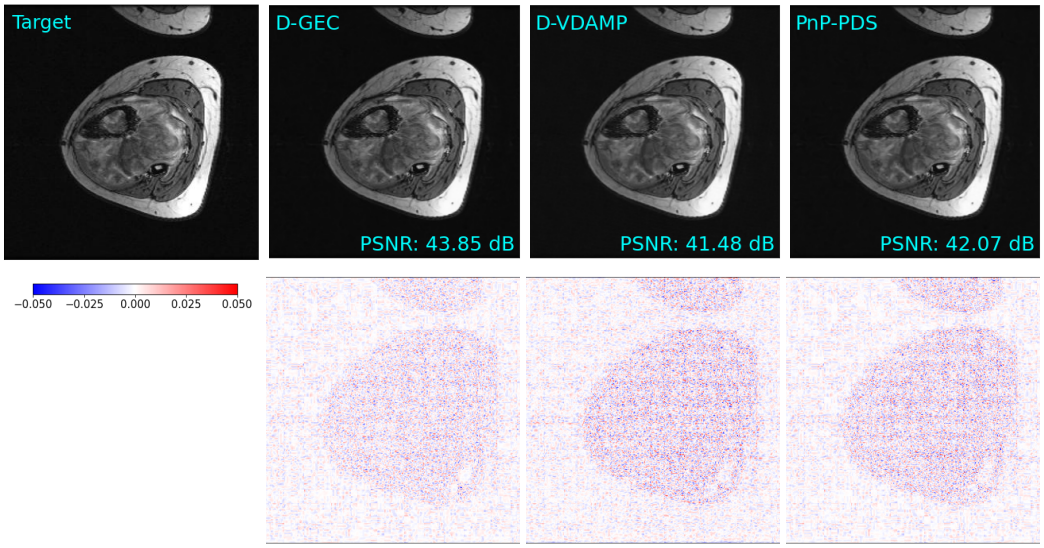


Figure 6: Example single-coil recoveries and error maps at $M/N = 1/4$.