Published in final edited form as:

IEEE Trans Med Imaging. 2023 April ; 42(4): 1133–1144. doi:10.1109/TMI.2022.3224359.

ENSURE: A general approach for unsupervised training of deep image reconstruction algorithms

Hemant Kumar Aggarwal [Member, IEEE],
Aniket Pramanik [Student Member, IEEE],
Maneesh John,
Mathews Jacob [Fellow, IEEE]
Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA, 52242

Abstract

Image reconstruction using deep learning algorithms offers improved reconstruction quality and lower reconstruction time than classical compressed sensing and model-based algorithms. Unfortunately, clean and fully sampled ground-truth data to train the deep networks is often unavailable in several applications, restricting the applicability of the above methods. We introduce a novel metric termed the ENsemble Stein's Unbiased Risk Estimate (ENSURE) framework, which can be used to train deep image reconstruction algorithms without fully sampled and noise-free images. The proposed framework is the generalization of the classical SURE and GSURE formulation to the setting where the images are sampled by different measurement operators, chosen randomly from a set. We evaluate the expectation of the GSURE loss functions over the sampling patterns to obtain the ENSURE loss function. We show that this loss is an unbiased estimate for the true mean-square error, which offers a better alternative to GSURE, which only offers an unbiased estimate for the projected error. Our experiments show that the networks trained with this loss function can offer reconstructions comparable to the supervised setting. While we demonstrate this framework in the context of MR image recovery, the ENSURE framework is generally applicable to arbitrary inverse problems.

Keywords

Unsupervised Learning; Inverse Problems; Deep Learning; SURE; MRI

I. Introduction

The recovery of images from a few of their noisy measurements is a classical inverse problem, which is central to several imaging modalities. For instance, the recovery of MR images from a few of their multi-channel k-space samples using compressed sensing (CS) [1]–[3] algorithms is a popular approach to speed up the scans. Deep learning methods have recently emerged as powerful alternatives to CS-based approaches. These methods rely on

jnu.hemant@gmail.com .

convolutional neural network (CNN) models that offer reduced computational complexity and improved performance. These deep learning methods can be categorized as direct inversion schemes [4], [5]and model-based schemes [6]–[11]. The model-based methods account for the imaging physics using a numerical model along with a deep, learned prior. The CNN modules in the above algorithms have been trained in a supervised mode, where an extensive amount of fully sampled training data is required. While large multi-center datasets are emerging for common applications such as brain and knee imaging [12], similar datasets are not available for all settings. More importantly, it is often challenging to acquire fully sampled datasets in applications including dynamic and static MRI with high spatial and temporal resolution. Unsupervised methods that can learn the deep networks from undersampled data directly can significantly improve the applicability of these approaches.

The unsupervised optimization of the parameters of image reconstruction and denoising algorithms have a long history. Early approaches relied on L-curve [13] and generalized cross-validation [14] to optimize the regularization parameters in regularized image recovery. Another approach is to use Stein's Unbiased Risk Estimate (SURE) [15] to determine the optimal parameters. The SURE loss is an unbiased estimate of the mean-square-error (MSE) that only depends on the recovered image and the noisy measurements. For example, SURE-let- and PURE-let-based methods optimize the thresholds in wavelet-based image denoising with Gaussian [16], [17] and Poisson noise[18]. Recently, SURE has been used to train deep image denoisers in [19]–[21]. The SURE approach was extended to inverse problems with a rank-deficient measurement operator, which is termed generalized SURE (GSURE) [22]. GSURE provides an unbiased estimate of the projected MSE, which is the expected error of the projections in the range space of the measurement operator. GSURE was recently used for inverse problems in [19]. Unfortunately, the experiments in [19] show that the GSURE-based projected MSE is a poor approximation of the actual MSE in the highly undersampled setting. To improve the performance, the authors trained the denoisers at each iteration in a message-passing algorithm in a layer-by-layer fashion using classical SURE, which is termed LDAMP-SURE [19]. A similar approach was used for MRI recovery in [21]. The GSURE metric was also used in a deep image prior framework recently [23]. In the supervised context, the end-to-end training of model-based algorithms [6]–[11] has been shown to be significantly more efficient than the layer-by-layer training strategies [19], [21].

Unsupervised strategies that do not use SURE have also been recently introduced. For instance, Noise2Noise [24] relies on a pair of noisy images to train a denoiser without the need for clean images. Noise2Void [25] is a blind spot method that excludes the central pixel from the network's receptive field. A similar strategy was used in self-supervised learning using data undersampling (SSDU) [25] for the end-to-end training of unrolled algorithms. This approach partitions the acquired measurements into two disjoint sets. The first set is used for the data consistency and the remaining set is used to define the loss. These methods offer reasonable results in the denoising (full-sampled) and low undersampling setting. However, as the undersampling factor increases, the available measurements are limited, restricting performance.

We introduce a novel loss function termed ensemble Stein's Unbiased Risk Estimate (ENSURE) to train deep networks. We assume that the sampling operator for each image is randomly chosen from a set; we evaluate the expectation of the GSURE losses over the sampling patterns. We show that the resulting loss metric is an unbiased estimate of the true MSE, hence, is a superior loss function when compared to projected SURE [22]. We note that such a measurement scheme can be realized in practice, where a different sampling mask can be chosen from a lookup table for each image. Similar to classical SURE metrics [16], [22], the proposed ENSURE loss metric is the sum of data consistency and divergence terms. While the divergence term is similar to GSURE, the data consistency term in ENSURE is the sum of the weighted projected losses [22], where the weighting depends on the class of sampling operators. When the divergence term is not used, the proposed loss reduces to only the projection error, which is observed to offer poor performance. The divergence term serves as a network regularization that minimizes the over-fitting to the measurement noise. The experiments demonstrate that the proposed ENSURE approach can offer results comparable to supervised training.

The main focus of this paper is to introduce the ENSURE metric for unsupervised training of deep networks. The main contributions of the paper are as follows

- We propose to sample different images using different undersampling operators. We introduce an new metric involving the weighted average of projected mean-square errors. We mathematically show that the expectation of this metric is essentially the true mean-square error.
- When fully sampled images are not available, we show that the weighted projected MSE can be approximated using the ENSURE metric, thus eliminating the need for fully sampled training datasets.
- We rigorously validate the proposed ENSURE metric in multiple datasets, undersampling patterns, and with different deep image reconstruction algorithms. We also compare the proposed framework against state-of-the-art unsupervised methods.

II. Background

In this section, we briefly review the background to make the paper self-contained.

A. Inverse Problems & Supervised Learning

We consider the acquisition of the complex-valued image $\boldsymbol{\rho} \in \mathbb{C}^N$ using the multichannel measurement operator $\mathscr{A}_s$ as

$$\boldsymbol{y}_s = \mathscr{A}_s\, \boldsymbol{\rho} + \boldsymbol{n}\,. \tag{1}$$

We assume the noise $\boldsymbol{n}$ to be complex-valued Gaussian distributed with zero mean and covariance matrix $\mathbf{C}$ such that $\boldsymbol{n} \sim \mathscr{N}(0, \mathbf{C})$. We note that the probability distribution function (PDF) of $\boldsymbol{y}_s$ is given by $p(\boldsymbol{y}_s \mid \boldsymbol{s}) = \mathscr{N}(\boldsymbol{A}_s\boldsymbol{\rho}, \mathbf{C})$.

In this work, we assume the measurement operator $\mathscr{A}_s$ to be randomly chosen and parameterized by a random vector $\boldsymbol{s}$ that is independent of $\boldsymbol{n}$, drawn from a set $\mathscr{S}$. For example, $\mathscr{A}_s$ can be viewed as a Fourier transform followed by multiplication with a k-space sampling mask parameterized by $\boldsymbol{s}$ in the single-channel MRI setting. The binary mask s is chosen at random, where the probability of a specific value being one is pre-specified. In the multi-channel setting, $\mathscr{A}_s$ corresponds to multiplying $\boldsymbol{\rho}$ with the coil sensitivities to obtain multiple sensitivity weighted images, followed by the undersampled Fourier transform of each of the images. The goal of the image reconstruction scheme is to recover $\boldsymbol{\rho}$ from $\boldsymbol{y}_s$. The inverse problem in (1) gets simplified to the image denoising model when $\mathscr{A}_s = \mathscr{I}$.

Supervised deep learning methods learn to recover the fully sampled image $\widehat{\boldsymbol{\rho}}$ only from noisy and undersampled measurements $\boldsymbol{y}_s$. Let

$$\boldsymbol{u} = \mathscr{A}_s^H \mathbf{C}^{-1} \boldsymbol{y}_s \tag{2}$$

We note that $\boldsymbol{u}$ is a random variable because of the additive noise in (1). In that case, the recovery using a deep neural network f_Φ with trainable parameters Φ is represented as

$$\widehat{\boldsymbol{\rho}} = f_\Phi(\boldsymbol{u}) \,. \tag{3}$$

Here f_Φ can be a direct inversion or a model-based deep neural network. Supervised approaches often rely on the loss function:

$$\text{Sup-Loss} \; = \sum_{i\,=\,1}^{N_{\text{train}}} \|\widehat{\boldsymbol{\rho}}_i - \boldsymbol{\rho}_i\|_2^2 \tag{4}$$

to train the network using N_{train} number of images.

B. Unsupervised Learning of Denoisers

When noise-free training data is unavailable, denoising approaches such as Noise2Noise [24] and blind-spot methods [25]–[27] have been introduced for unsupervised learning of the network parameters Φ. Assuming the additive noise $\boldsymbol{n}$ to be Gaussian distributed, the SURE [15] approach uses the loss function

$$\text{SURE}(\widehat{\boldsymbol{\rho}}, \boldsymbol{u}) = \; \| \, \widehat{\boldsymbol{\rho}} - \boldsymbol{u} \, \|_2^2 \; + 2\sigma^2 \nabla_{\boldsymbol{u}} \cdot f_\Phi(\boldsymbol{u}) - N\sigma^2, \tag{5}$$

which is an unbiased estimate of the true mean-square error, denoted by

$$\text{MSE} = \mathbb{E}_{\boldsymbol{u}} \, \| \, \widehat{\boldsymbol{\rho}} - \boldsymbol{\rho} \, \|^2 \tag{6}$$

Note that the expression in (5) does not depend on the noise-free images $\boldsymbol{\rho}$; it only depends on the noisy images $\boldsymbol{u}$ and the estimates $\widehat{\boldsymbol{\rho}}$. In (5), $\nabla_{\boldsymbol{u}} \cdot f_\Phi(\boldsymbol{u})$ represents network divergence, which is often estimated using Monte Carlo simulations [28]. Several researchers have adapted SURE as a loss function for the unsupervised training of deep image denoisers [19], [21] with performance approaching that of supervised methods.

C. Unsupervised training with rank deficient $\mathscr{A}$

The unsupervised training of deep networks for image recovery is significantly more challenging than the denoising setting when $\mathscr{A}$ is rank deficient. In particular, the measurement model only acquires partial information about the images, in addition to the additive noise. The Deep Image Prior (DIP) [29] approach exploits the structural bias of CNNs towards natural images; they optimize the network parameters such that the loss $\| \boldsymbol{A}_s\hat{\boldsymbol{\rho}} - \boldsymbol{y}_s \|_2^2$ is minimized. Since this approach requires the network to be trained for each image, the computational complexity is high during inference. In addition, the quality of the recovered images is often inferior to the ones obtained from supervised training. Another challenge is the need for manual early stopping of the algorithm to minimize the over-fitting to noise.

One may use the generalization of a DIP approach for an ensemble of images, which we term as measurement domain (k-space in MRI) MSE loss:

$$\text{K-MSE} = \sum_{i=1}^{N_{\text{train}}} \|\mathscr{A}_{s_i}\hat{\boldsymbol{\rho}}_i - \boldsymbol{y}_{s_i}\|_2^2 . \tag{7}$$

This K-MSE approach is prone to overfitting the reconstructed image to the noisy measurements [26]. The SSDU approach [26] was introduced to minimize this over-fitting [26]. SSDU, which may be viewed as an extension of blind-spot methods (e.g., [25]), suggests partitioning the available k-space into two disjoint groups for the data-consistency step and loss function estimation, respectively. The partitioning of k-space, along with the use of different sampling masks for different images, is observed to provide improved results when compared to (7).

The projected GSURE approach was used to train model-based deep learning algorithms in the unsupervised setting in [19]. Specifically, the GSURE loss is an unbiased estimate of the projected MSE

$$\text{MSE}_s = \mathbb{E}_{\rho \sim \mathscr{M}} \| \mathbf{P}_s(\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}) \|^2, \tag{8}$$

where $\mathbf{P}_s = \mathscr{A}_s^H(\mathscr{A}_s\mathscr{A}_s^H)^{-1}\mathscr{A}_s$ is the projection operator to the range space of $\mathscr{A}_s^H$ and $\mathscr{M}$ is the set of images. The authors of [19] noted that MSE_s is a poor approximation of MSE in the highly undersampled setting. Hence, instead of directly using GSURE, the LDAMP algorithm instead relies on layer-by-layer training of deep learned denoisers [19] using the SURE loss discussed in Section II-B. This scheme implicitly assumes that the alias artifacts at each iteration are Gaussian distributed, which is not a realistic assumption. In addition, the end-to-end training of model-based algorithms [6]–[9] often offers improved performance compared to the above layer-by-layer strategies.

III. Ensemble SURE (ENSURE) framework

The training of the deep network f_Φ in (3) using only the measurements $\boldsymbol{y}_s$ in (1) is challenging when the sampling operator $\mathscr{A}_s$ is rank deficient. More specifically, $\boldsymbol{y}_s$ carries only partial information about the images $\boldsymbol{\rho}$, in addition to being noisy. To improve the training in the setting without fully sampled images, we consider the sampling of each image using a different sampling pattern. In the MRI context, we assume the k-space sampling mask s to be a random vector drawn from the distribution $\mathcal{S}$. We note that this acquisition scheme is realistic and can be implemented in many applications. For instance, it may be difficult to acquire a specific image in a fully sampled fashion due to time constraints. However, one could use a different undersampling mask for each image $\boldsymbol{\rho} \sim \mathscr{M}$. The main theoretical contribution of this work is that the expectation of the weighted SURE metrics over the sampling patterns is equal to the true MSE, which offers a better metric than GSURE, which approximates the projected MSE.

In the next section, we consider the expectation of the projected MSE over the sampling patterns, which involve the ground truth images. We show that averaging the projected MSE values will not approximate the true MSE, but a weighted version of the MSE. We thus define a projected and weighted error, which when averaged over the sampling patterns will approximate the true MSE. In Section III-B, we introduce the ENSURE metric, which evaluates the projected and weighted error metric without using the ground truth images; we show that the expectation of ENSURE yields the true MSE.

A. Averaging the projected error over sampling patterns

We consider a specific image ρ and consider the projected error $\| \mathbf{P}_s\widehat{\boldsymbol{\rho}} - \mathbf{P}_s\boldsymbol{\rho} \|^2$. In the noiseless setting, we compute $\mathbf{P}_s\boldsymbol{\rho}$, which depends on the sampling pattern $\boldsymbol{s}$. In a noisy setting, this measure cannot be directly computed. We now compute the expectation of the projected errors over an ensemble of measurement operators:

$$\mathscr{Q}(\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{u}}\left[\mathbb{E}_{s \sim \mathcal{S}}\left[\| \mathbf{P}_s(\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}) \|^2\right]\right]. \tag{9}$$

The following result shows that the above measure only gives a weighted version of the true MSE in (6).

Lemma 1: The expected loss $\mathscr{Q}$ in (9) is equal to the weighted MSE:

$$\mathscr{Q} = \mathbb{E}_{\boldsymbol{u}}\left[\| \boldsymbol{W}(\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}) \|^2 \right] \tag{10}$$

where $\mathbf{W}$ is a weighting matrix that is dependent on the set of measurement operators.

Proof: Denoting the error $\boldsymbol{e} = \widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}$ and ignoring its dependence on $\boldsymbol{s}$, we obtain

$$\mathscr{Q} = \mathbb{E}_{\boldsymbol{u}}\left[\mathbb{E}_s\left[\| \mathbf{P}_s\boldsymbol{e} \|^2\right]\right] \tag{11a}$$

$$\approx \mathbb{E}_{\boldsymbol{u}}\left[\boldsymbol{e}^T\mathbb{E}_s[\mathbf{P}_s]\boldsymbol{e}\right] = \mathbb{E}_{\boldsymbol{u}}\left[\boldsymbol{e}^T\mathbf{Q}\boldsymbol{e}\right] \tag{11b}$$

We used the symmetry of the projection operators and $\mathbf{P}^2 = \mathbf{P}$ in the above step. Here, $\mathbf{Q}$ is the ensemble of projection operator. We denote the matrix square root of $\mathbf{Q}$ by $\mathbf{W}$ (i.e., $\mathbf{W}^2 = \mathbf{Q}$) to obtain

$$\mathcal{Q} = \mathbb{E}_{u}\left[\boldsymbol{e}^{T}\mathbf{W}^{T}\mathbf{W}\boldsymbol{e}\right] \tag{11c}$$

$$= \mathbb{E}_{u}\left[\parallel \mathbf{W}(\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}) \parallel_2^2 \right], \tag{11d}$$

Note that $\mathbf{Q}$ corresponds to the average of the projection operators. Depending on the sampling operators, the weighting by $\mathbf{W}$ amounts to weighting some aspects of the images more heavily than other features.

We will now illustrate the above result in the special case of single-channel MRI when $\mathcal{A}_s = \mathbf{SF} \in \mathbb{C}^{n \times m}$, where $\mathbf{F} \in \mathbb{C}^{m \times m}$ is the Fourier matrix and $\mathbf{S} \in \mathbb{R}^{n \times m}$; $n < m$ is the sampling matrix. The projection $\mathbf{P}_s$ gets simplified as $\mathbf{P}_s = \mathcal{A}_s^H(\mathcal{A}_s\mathcal{A}_s^H)^{-1}\mathcal{A}_s = \mathbf{F}^H\mathbf{M}_s\mathbf{F}$. Here, $\mathbf{M}_s = \mathbf{S}^T\mathbf{S} \in \mathbb{C}^{m \times m}$ is a diagonal matrix. The diagonal entries are one if the specific sample is acquired and zero otherwise.

We now consider the diagonal entries to be drawn from a Bernoulli distribution with a probability that depends on the spatial frequency location. Note that variable-density masks are routinely used in CS settings. We thus obtain $\mathbf{Q} = \mathbf{F}^H\mathrm{diag}(\boldsymbol{d})\mathbf{F}$, which translates to

$$\mathbf{W} = \mathbf{F}^{H}\mathrm{diag}(\sqrt{\boldsymbol{d}})\mathbf{F} \tag{12}$$

The weighted loss $\mathcal{Q}$ is equivalent to the MSE if the sampling density is uniform. However, most sampling operators rely on variable-density masks with higher density in the k-space center. In this case, the training of the networks using $\mathcal{Q}$ will translate to reduced weighting for higher Fourier samples, resulting in reduced high-frequency details.

B. ENSURE: Unbiased Estimate of MSE

To compensate for this weighting, we consider the weighted version of the projected error and compute the expectation over the set of sampling patterns $\mathcal{S}$ and the images, or equivalently $\boldsymbol{u}$:

$$\mathcal{L} = \mathbb{E}_{u}\left[\mathbb{E}_{s \sim \mathcal{S}}\left[\parallel \mathbf{R}_s(\widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}) \parallel_2^2\right]\right], \tag{13}$$

where $\mathbf{R}_s = \mathbf{W}^{-1}\mathbf{P}_s$ is the weighted projection operator. We note that $\mathbf{R}_s\boldsymbol{\rho}$ still lives in the range space of $\mathcal{A}$. Using a similar argument in Lemma 1, we can show that $\mathcal{L} = \mathrm{MSE}$.

We note that the loss function (13) depends on the fully sampled ground truth images $\boldsymbol{\rho}_i$; $i = 1, \ldots N$. Hence, it is impossible to directly compute it in an unsupervised setting. We hence use the SURE approach to evaluate it without requiring the ground truth images. We expand (13) as

$$\mathscr{L} = \mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\| \mathbf{R}_{s}\hat{\boldsymbol{\rho}} \|_2^2\right] + \mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\| \mathbf{R}_{s}\boldsymbol{\rho} \|_2^2\right] \\ - 2\mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\hat{\boldsymbol{\rho}}^{T}\underbrace{\mathbf{R}_{s}{}^{T}\mathbf{R}_{s}\rho\Big]}_{\mathbf{D}}\right] \tag{14}$$

We note that the estimate $\widehat{\boldsymbol{\rho}} = f_{\Phi}(\boldsymbol{u})$ in our setting is provided by a deep network f, whose weights are denoted by Φ and hose input is

$$\boldsymbol{u} = \mathscr{A}_s^{H}\mathbf{C}^{-1}\boldsymbol{y} \tag{15}$$

Our goal is to train the network f_{Φ} or equivalently minimize the above expression with respect to $\widehat{\boldsymbol{\rho}}$.

We note that the first term in (14) is independent of ρ and can be computed. The second term is a constant that does not depend on $\widehat{\boldsymbol{\rho}}$. The third term, which includes both $\boldsymbol{\rho}$ and $\widehat{\boldsymbol{\rho}}$. We show in the Appendix that this term can be computed using SURE as

$$\mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\hat{\boldsymbol{\rho}}^{T}\mathbf{R}_{s}^{T}\mathbf{R}_{s}\mathrm{\rho}\right] = \mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\hat{\boldsymbol{\rho}}^{T}\mathbf{D}\boldsymbol{\rho}_{\mathrm{LS}} - \nabla_{\boldsymbol{u}}\cdot\mathbf{D}\hat{\boldsymbol{\rho}}\right] \tag{16}$$

Here, $\boldsymbol{\rho}_{\mathrm{LS}}$ is the least-squares solution specified by

$$\boldsymbol{\rho}_{\mathrm{LS}} = \left(\mathscr{A}_s^{H}\mathbf{C}^{-1}\mathscr{A}_s\right)^{\dagger}\mathscr{A}_s^{H}\mathbf{C}^{-1}\boldsymbol{y}_s\,. \tag{17}$$

The term $\nabla_{\boldsymbol{u}}\cdot\mathbf{D}f_{\Phi}(\boldsymbol{u})$ represents the divergence of the network f_{Φ} with respect to its input $\boldsymbol{u}$.

Combining (14) and (16), we obtain

$$\mathscr{L} = \mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\| \mathbf{R}_{s}\hat{\boldsymbol{\rho}} \|_2^2 + \| \mathbf{R}_{s}\boldsymbol{\rho} \|_2^2 - 2(\mathbf{R}_{s}\hat{\boldsymbol{\rho}})^{\mathrm{T}}\mathbf{R}_{s}\boldsymbol{\rho}_{\mathrm{LS}}\right] \\ + 2\mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\nabla_{\boldsymbol{u}}\cdot(\mathbf{D}\hat{\boldsymbol{\rho}})\right] \tag{18}$$

$$= \mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\| \mathbf{R}_{s}(\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}_{\mathrm{LS}}) \|_2^2 + 2\,\nabla_{\boldsymbol{u}}\cdot(\mathbf{D}\hat{\boldsymbol{\rho}})\right] + \kappa \tag{19}$$

We combined the first and third terms in (14) and completed the square to obtain the ENSURE loss as (19).

Here,

$$\kappa = \mathbb{E}_{\boldsymbol{u}}\mathbb{E}_{s}\left[\| \mathbf{R}_{s}\rho \|_2^2 - \| \mathbf{R}_{s}\rho_{\mathrm{LS}} \|_2^2\right] \tag{20}$$

is a constant that is independent of the network.

In practice, we approximate the expectation by a summation

$$\mathrm{ENSURE} \; = \frac{1}{N}\sum_{i=1}^{N}\left(\underbrace{\left\|\mathbf{R}_{s_i}\widehat{\boldsymbol{\rho}}_i - \boldsymbol{\rho}_{\mathrm{LS,i}}\right\|_2^2}_{\text{dataterm}} + 2\underbrace{\nabla_{\boldsymbol{u}_i}\cdot\left(\mathbf{D}f_{\Phi}(\boldsymbol{u}_i)\right)}_{\text{divergence}}\right) \tag{21}$$

where we ignore the constant κ. Here, $\boldsymbol{u}_i$ are the zero-filled reconstructions (see Section (III-B.1) for details) of the i^{th} image. We assume that each image ρ_i is sampled by the corresponding sampling pattern $\boldsymbol{s}_i$.

The loss function in (21) is an unbiased estimate for the true MSE, specified by (6), up to a constant. Since this loss function does not require the fully sampled reference images, it can be used to train a deep learning image reconstruction algorithm in an unsupervised fashion. The data term in (21) involves the average of the weighted projected losses evaluated over the N training examples. The divergence term is computed over all the data and serves as a regularization term to account for the impact of noise.

We now focus on some special cases in the context of MRI.

1) Single-channel setting with $\mathbf{C} = \sigma^2\mathbf{I}$: In many cases, we assume that $\mathbf{C} = \sigma^2\mathbf{I}$, when we obtain

$$\boldsymbol{\rho}_{LS} = \left(\mathscr{A}_s^H\mathscr{A}_s\right)^{\dagger}\mathscr{A}_s^H\boldsymbol{y}_s = \mathscr{A}_s^H\left(\mathscr{A}_s\mathscr{A}_s^H\right)^{-1}\boldsymbol{y}_s\,. \tag{22}$$

In the single-channel setting, $\mathscr{A}_s\mathscr{A}_s^H = \mathscr{I}_{n\times n}$ because of the orthogonality of Fourier exponentials. Thus ρ_{LS} simplifies to

$$\boldsymbol{v} = \mathscr{A}_s^H\boldsymbol{y}_s, \tag{23}$$

which is often termed as the zero-filled reconstruction. In this case, the data term involves the comparison of the projection of the reconstructed image and the zero-filled reconstructions.

2) Multichannel case with $\mathbf{C} = \boldsymbol{\sigma}^2\mathbf{I}$: : In the multichannel setting, $\mathscr{A}_s\mathscr{A}_s^H \neq \mathscr{I}$, and hence the least-square solution is not equal to the zero-filled reconstruction $\boldsymbol{v}$. Many deep learning algorithms use the zero-filled image $\boldsymbol{v}$ as the input to the deep networks rather than its scaled version $\boldsymbol{u} = \sigma^2\boldsymbol{v}$; we now express the divergence term in (21) in terms of $\boldsymbol{v}$. We consider a network g_ϕ, whose input is v rather than $\boldsymbol{u}$: $g_\phi(\boldsymbol{v}) = f_\phi(\boldsymbol{u})$. We thus have

$$\nabla_{\boldsymbol{u}_i}\cdot f_\Phi(\boldsymbol{u}_i) = \nabla_{\boldsymbol{u}_i}\cdot g_\Phi(\boldsymbol{v}_i) = \nabla_{\boldsymbol{v}_i}\cdot g_\Phi(\boldsymbol{v}_i)\underbrace{d\boldsymbol{v}_i/d\boldsymbol{u}_i}_{\sigma^2} \tag{24}$$

In this case, the ENSURE loss in (21) can be rewritten as

$$\text{ENSURE } = \frac{1}{N}\sum_{i=1}^{N}\left(\underbrace{\left\|\mathbf{R}_{s_i}\left(\widehat{\boldsymbol{\rho}}_i - \boldsymbol{\rho}_{\text{LS,i}}\right)\right\|_2^2}_{\text{dataterm}} + 2\sigma^2\underbrace{\nabla_{v_i}\cdot g_\Phi(\boldsymbol{v}_i)}_{\text{divergence}}\right) \tag{25}$$

Here, we ignored the constant term κ. We note that this form is similar to (5), especially when $\mathbf{R}_s = \mathbf{I}$.

We note that the divergence term in (23) is scaled by σ^2. The divergence serves as a regularization term on the deep network, making the learned network more robust to input noise/perturbations. The balance between the data consistency term and the divergence term in (25) is controlled by $2\sigma^2$; more regularization is applied when the input is more noisy. When $\sigma \to 0$, the divergence term disappears, when the ENSURE based approach simplifies to a loss-function that only involves k-space losses. In this work, we do not consider any additional tuning parameter to the divergence term; our loss function is specified by (25).

C. Implementation details

We compute the least-squares solution as

$$\rho_{\mathrm{LS}} = \underset{\boldsymbol{\rho}}{\arg\min} \| \mathscr{A}_s \boldsymbol{\rho} - \boldsymbol{y}_s \|^2 + \lambda \| \boldsymbol{\rho} \|^2 \tag{26}$$

$$= (\mathscr{A}_s^H \mathscr{A}_s + \lambda \mathscr{I})^{-1} \mathscr{A}_s^H \boldsymbol{y}_s \tag{27}$$

with $\lambda \to 0$. When $\lambda = 0$, the above equation can be shown to be equivalent to $\mathscr{A}_s^H (\mathscr{A}_s \mathscr{A}_s^H)^{-1} \boldsymbol{y}_s$. In the parallel MRI setting, the least-square solution $\rho_{\mathrm{LS},s}$ in (22) is often called the SENSE [30] solution. We solve (26) using the conjugate gradients algorithm.

We compute the projection operator $\mathbf{P}_s$ as $\mathbf{P}_s = (\mathscr{A}_s^H \mathscr{A}_s)^{\dagger} (\mathscr{A}_s^H \mathscr{A}_s)$ in (9) and (21). We note that the projection can be obtained by replacing $\boldsymbol{y}_s$ in (26) with $\boldsymbol{y}_s = \mathscr{A}_s \boldsymbol{\rho}$. We thus have:

$$\mathbf{P}_s \boldsymbol{\rho} = \underset{\boldsymbol{\zeta}}{\arg\min} \| \mathscr{A}_s \boldsymbol{\zeta} - \mathscr{A}_s \rho \|^2 + \lambda \| \boldsymbol{\zeta} \|^2 \tag{28}$$

with $\lambda \to 0$; in the MRI setting, this corresponds to the SENSE recovery from $\mathscr{A}_s \boldsymbol{e}$. When the weighting is also included, we compute

$$\mathbf{R}_s \boldsymbol{e} = \underset{\boldsymbol{\zeta}}{\arg\min} \| \mathbf{D}_s \mathscr{A}_s (\boldsymbol{\zeta} - \boldsymbol{e}) \|^2 + \lambda \| \boldsymbol{\zeta} \|^2 , \tag{29}$$

where $\mathbf{D}_s = \mathrm{diag}\left(\boldsymbol{d}_s^{-\frac{1}{2}}\right)$, where $\boldsymbol{d}_s$ corresponds to the density at the sampling locations.

Combining (26) and (29), the computational structure of the data term is shown in Fig. 1(a). In particular, we compare the recovered images with their CG-SENSE reconstructions specified by ρ_{LS}. Note that the CG-SENSE solutions may have residual aliasing components. To make the data consistency term insensitive to these errors, the error $\boldsymbol{e} = \widehat{\boldsymbol{\rho}} - \boldsymbol{\rho}_{\mathrm{LS}}$ is projected to the range space of $\mathscr{A}_s$. An additional weighting is used to compensate for the non-uniform density of the sampling patterns in k-space, which is implemented as in (29).

The divergence term in (21) is computed using Monte-Carlo simulations [28] as in Fig. 1(b). Specifically, noise perturbations are added to the input to the network, and the corresponding perturbations in the output of the network are estimated. The divergence term is approximated as

$$\mathrm{div}_u(f_\Phi(u)) = \lim_{\epsilon \to 0} \frac{1}{\epsilon}\mathbb{E}_n\left[\boldsymbol{n}^T\left(f_\Phi(\boldsymbol{u}_s + \epsilon\,\boldsymbol{n}) - f_\Phi(\boldsymbol{u}_s)\right)\right].$$

Here $\boldsymbol{n}$ is the standard Gaussian. As in [28], we observe that the expectation can be approximated by a Monte-Carlo approach involving different noise realizations. In practice, one noise realization per image per epoch is sufficient to obtain a good approximation as observed in [28]. Note that the use of the data term alone will result in over-fitting, similar to observations in DIP methods as the number of epochs increases. The divergence term may be viewed as a network regularization, which serves to minimize noise amplification.

In this work, we estimate the **W** matrix by evaluating multiple sampling matrices and by computing their average.

In this work, we have assumed the Cartesian sampling mask, where each entry is a Bernoulli random variable whose probability at each spatial frequency location to be given by a Gaussian function; the probability of the samples is higher in the k-space center and lower at higher frequencies. We do not fully sample the k-space center. Algorithm I summarizes the steps of the proposed ENSURE loss for performing unsupervised training.

Algorithm 1

ENSURE loss function calculation.

```
Input: $\boldsymbol{y}_s$, $\mathscr{A}_s$
Output: ENSURE loss
1: Estimate the zero-filled reconstruction $u = \mathscr{A}_s^H \boldsymbol{y}_s$.
2: Estimate the least square solution $\rho_{LS}$ by solving Eq. (22).
3: Estimate the network prediction $\widehat{\boldsymbol{\rho}}$ using $f_\Phi$.
4: find the error as $e = \widehat{\boldsymbol{\rho}} - \rho_{LS}$.
5: Solve Eq. (29) to apply weighted projected $\mathscr{R}_s$ on $e$.
6: Take the L2-norm of $\mathscr{R}_s e$ to get the data-term.
7: Find divergence term of Eq. (25) as in Fig.1 (b).
```

```
8: Estimate the ENSURE loss as the average of data-term and scaled divergence term as in Eq. (25).
```

D. Deep network architectures

We used the model-based deep learning architecture, as shown in Fig. 2 for the initial comparisons. The CNN had five layers, each with 3 × 3 convolution, batch normalization, and ReLU non-linearity. Fig. 2 also shows the number of feature maps on top of each layer. The network had shared parameters in three unrolling steps. We trained the network for a total of 50 epochs with a fixed learning rate of 10^{-3}. The data-consistency (DC) step utilized the complex data, whereas the CNN used the real and imaginary components of the complex data as channels.

IV. Results

We consider publicly available [6] parallel MRI brain data acquired using a 3-D T2 CUBE sequence with Cartesian readouts using a 12-channel head coil at the University of Iowa on a 3T GE MR750w scanner. The Institutional Review Board at the University of Iowa approved the data acquisition, and written consent was obtained from the subjects. The matrix dimensions were 256 × 232 × 208 with a 1 mm isotropic resolution. Fully sampled multi-channel brain images of nine volunteers were collected, out of which data from five subjects were used for training. The data from two subjects were used for testing and the remaining two for validation.

The validation data was used to terminate the training process. We utilized the ESPiRiT algorithm [31] to estimate the coil sensitivity maps.

We also used a subset of the fast-MRI [12] knee dataset, consisting of 100 subjects, to demonstrate the utility of ENSURE on multiple unrolled architectures. The data from 43 subjects were used for training, two for validation, and the rest for testing.

A. Comparison with state-of-the-art methods

We compare the proposed ENSURE approach with the following supervised and unsupervised methods.

Two supervised learning algorithms—E-to-E-Sup and L-by-L-Sup: The E-to-E-Sup approach refers to the end-to-end supervised training approach [6]. To make it consistent with the rest of the experiments, we use multiple sampling patterns during training. The L-by-L-Sup uses a similar acquisition and reconstruction architecture but relies on the supervised training of the CNNs in each layer. Both approaches use the loss function in (4) for training. We note that the number of free parameters in L-by-L-Sup is threefold higher than E-to-E-Sup, where the CNN parameters are shared across iterations.

GSURE-based unsupervised learning: The GSURE loss, which is an unbiased estimate of projected MSE in (8), was originally introduced for regularization parameter selection. We consider the direct use of GSURE to train deep image reconstruction algorithms. Here, the sampling operator is assumed to be the same for all training images as considered in [19].

Unsupervised learning with different sampling patterns: We compare the proposed scheme against K-MSE, SSDU [26], and LDAMP-SURE [19]. The measurement operators are randomly chosen for each image in all of these approaches, but are assumed to be fixed for each image; we do not vary them during iterations. The K-MSE (7) loss function involves the square of the error between measured and predicted k-space. As discussed previously, the K-MSE approach is vulnerable to over-fitting to noise. To minimize over-fitting, SSDU partitions the measured k-space samples into two disjoint groups: one group is used to reconstruct the images, while the second is used to evaluate the loss function. The 80–20 partition of k-space offered the best PSNR in our experimental setup. The LDAMP-SURE algorithm relies on layer-by-layer training. The five-layer CNN at each layer is independently trained as a denoiser using the SURE loss function.

We consider the recovery from 2-D random sampled multi-channel data in Table I. The PSNR and SSIM [32] for two different acceleration factors of 4x and 6x are reported in the first and last columns. To study the impact of measurement noise on the algorithms, we also consider a setting with additional Gaussian noise of standard deviation $\sigma = 0.03$ added to the undersampled k-space data at 4x acceleration; these results are shown in the second column of Table I. The top two rows consider the supervised setting, while the remaining rows denote the unsupervised methods discussed above.

Fig. 3 visually compares the reconstruction quality of different unsupervised learning techniques with E-2-E-Sup learning at the 4x acceleration factor. In this low-noise setting, K-MSE offers nearly the same reconstruction quality as SSDU, which is also seen in Table I. The zoomed region shows that LDAMP-SURE, K-MSE, and SSDU have visible artifacts in the reconstructions compared to GSURE and ENSURE. The image quality of GSURE and ENSURE are comparable in this setting.

Fig 4 shows reconstruction results at a higher acceleration factor of 6x. We note that the performance of the ENSURE approach is comparable to that of the supervised approach at this high acceleration rate. By contrast, we observe that GSURE, K-MSE, and LDAMP-SURE exhibit more noise-like artifacts along with blurring. A green rectangle in the zoomed images shows a hallucinated feature in the LDAMP-SURE because of layer-by-layer training of individual networks instead of end-to-end training in the proposed ENSURE approach. The SSDU approach offers less-noisy reconstructions compared to these approaches. However, the PSNR is around 0.5 dB lower than ENSURE. An arrow in the zoomed cerebellum region points to a feature that is well preserved by ENSURE while blurred by SSDU.

B. Performance in high-noise setting

As described previously, the divergence serves as a regularization term on the deep network, making the learned network more robust to input noise/perturbations. In the low noise setting ($\sigma \to 0$), the weight for the divergence term in (25) tends to zero. In this case, ENSURE simplifies to the DC term, which is a weighted version of the K-MSE loss. The difference of ENSURE with this simple strategy is higher at high noise settings, which might happen at low-field or FLAIR acquisitions. To study the benefit of the proposed formulation in high noise settings, Gaussian noise of standard deviation $\sigma = 0.03$ was added to the measured k-space data at 4x acceleration. The comparison of the methods is shown in Fig. 5, which can also be seen in the second column of Table I. In this setting, we observe that E-to-E-Sup and L-to-L-Sup have nearly the same mean PSNR values. We observe that the LDAMP-SURE approach performs better than the GSURE approach in this high noise setting, as reported in [19]. In particular, the projected MSE is a poor approximation for the MSE, which translates to poor reconstructions in the presence of noise. The improved performance of SSDU over KMSE shows that the partitioning strategy used in SSDU can reduce the over-fitting to noise. The ENSURE approach offers the best PSNR out of the unsupervised strategies and is comparable in PSNR and image quality to E-to-E-Sup. We observe that ENSURE is robust to increased noise variance when compared to other algorithms that do not use the divergence term. While GSURE also uses the divergence term, the data consistency term is very different from ENSURE, which causes an imbalance between the two terms; we attribute the lower performance of GSURE in this setting to the imbalance.

C. Impact of different loss terms: ablation study

The ENSURE approach has two key differences from the GSURE framework: (a) the use of an ensemble of sampling patterns, and (b) the use of additional weighting within the DC term. We now study the impact of these components on performance.

1) Need for using an ensemble of measurement operators: We compare the GSURE framework, which uses a single sampling pattern for all the images, with the ENSURE formulation, which uses multiple sampling operators in Figure 6. A one-dimensional Cartesian sampling at 2x acceleration with an additional Gaussian noise of $\sigma = 0.02$ was used. We note that aliasing artifacts and blurring are visible in the GSURE approach. In contrast, the ENSURE approach offers comparable performance to the E-to-E-Sup-MSE. A major source of the improved performance is the use of the ensemble of measurement operators, which makes ENSURE closely approximate the true MSE. By contrast, the GSURE approach is only approximating the projected MSE, which is inferior in the undersampled setting, as discussed in [19].

2) Impact of the weighting: This experiment compares the impact of the proposed weighting strategy in (13) at 4x acceleration using a 2D random variable-density sampling pattern. We trained two networks with experimental setups that are similar except for the loss function. When we set the weights $\mathbf{W}$ to identity in (21), it results in a network trained with the ENSURE loss without weighting. The reconstruction quality and error maps in Fig. 7 demonstrate that the proposed weighting scheme improves the reconstruction quality significantly.

D. Applicability to other deep image recovery algorithms

We demonstrate the applicability of the proposed loss function to other deep image recovery algorithms at 2x acceleration in Table II; the reconstructed images are not shown in this case because there are no substantial differences between them. Here, we use a subset of 100 datasets from fastMRI [12]. The noise variance was estimated from one fully-sampled dataset, which was used for the rest of the datasets. The first four columns in Table II correspond to different algorithms trained using fully sampled ground truth and ENSURE, respectively. A uniform density sampling pattern with the fully sampled center for coil sensitivity estimation, provided by fastMRI challenge [12], was used in these experiments. In each column, we report the PSNR (top) and the SSIM (bottom) values. The comparisons show that all of the algorithms (ADMMnet, FISTAnet, and MoDL) offer somewhat similar performance on this dataset when trained with MSE. The results also show that ENSURE is capable of offering comparable reconstructions for these algorithms.

We also study the impact of ENSURE under different sampling conditions in the last three columns on Table II. MoDL-MS refers to ENSURE implementation, where a different random undersampled mask is used at each epoch. We note that such an approach is not feasible in settings where only the undersampled data is available. This experiment is to determine the impact of mask shuffling on image quality. We note that the results in this setting are compared to MoDL without mask shuffling. This shows that the approximation of the expectation over the sampling pattern in (13) is accurate, even when only one sampling pattern per image is considered (MoDL setting, without mask, shuffling). The last two columns, SENSE-VD and MoDL-VD, consider 2x variable-density sampling, using masks similar to the brain experiments. We note that the higher density in the center of k-space translates to improved performance. However, the gap between MSE and ENSURE remains almost the same, indicating that the specific choice of sampling patterns have less impact on the ENSURE approximation.

V. Discussion

We observe from Table I, as well as the figures, that the proposed unsupervised learning using the ENSURE loss function offers PSNR and SSIM values that are comparable to those of E-to-E-Sup. The experiments in Section IV-C show that the ensemble of measurement operators and the weighted loss function enabled the ENSURE loss to closely approximate the true MSE when fully sampled and noise-free images are not available.

We note that the performance of the L-by-L-Sup approach, which has threefold more parameters, is only marginally worse than the E-to-E-Sup approach in all the settings. We note that the LDAMP-SURE approach, which relies on layer-by-layer training, is worse than ENSURE. This deterioration in performance may be attributed to the non-Gaussianity of the alias artifacts at each iteration, which is the key assumption used in LDAMP-SURE. We observe that the K-MSE scheme offers low performance, especially at high-noise and highly accelerated settings. This is caused by the over-fitting of the network to the noise within the measurements. The partitioning strategy used in SSDU is observed to offer improved performance over K-MSE, especially in the high-noise setting. Similarly, the performance improvement of ENSURE over GSURE is more significant in the high-noise and highly accelerated setting. We trained the networks for each setting independently. The experiments with additional synthetic noise involved both training and testing the networks at the high noise setting. We note that all model-based algorithms (e.g. SENSE that combines information from multiple channels and compressed sensing) simultaneously perform reconstruction and denoising; the reconstructed signal is not data-consistent in any of these cases.

The ENSURE and GSURE techniques depend on the estimation of noise variance in the measured data. In the experiments in Table II, we estimate the noise variance from one fully sampled complex coil combined image. Since the image is complex, the noise will still be Gaussian distributed. We threshold the reconstructed image to identify the image support. The variance of the image in the background region is used as the noise variance. We estimate the noise variance for LDAMP-SURE [19], [21] as suggested in [21].

We used the network architecture in Fig. 2 for the comparisons of loss functions. However, any direct-inversion or model-based network architecture can use the proposed ENSURE loss function. Unlike LDAMP-SURE, our proposed method does not depend on the unrolling of any specific algorithm.

The SSDU method [26] suggests partitioning the measured k-space such that 60% is used in the DC step and 40% in the loss function estimation. With our dataset and network architecture, we experimented with several different partitions of the k-space, including the suggested 60–40 split, and found that an 80–20 division performed best.

The main focus of this work is on how to train a network without fully sampled reference images. We note that there are several practical considerations, including dependence of the trained network on the specific field of view and contrasts, which are non-trivial issues. As shown in earlier work [6], [33], the training strategy may be modified to make the network relatively less sensitive to these changes. However, these problems are shared between supervised and unsupervised methods. To keep the paper focused, we do not address these aspects in this work. In this work, we train different networks for different undersampling ratios. As shown in [6], [33], the proposed work may be extended to make the trained network insensitive to the specific sampling ratio. However, we leave this potential extension for future work to keep the paper brief. In this work, we assume that the average weight matrix $\mathbf{W}$ is invertible. In the single channel setting, this will not be true if some Fourier samples are not included or acquired by any of the sampling patterns. In this case, it will be impossible to realize an unbiased estimate for the true MSE. One may come up with a projected MSE onto the observed samples. We leave this potential extension for future work, with the focus on keeping the paper brief.

We note that there is a gap between the supervised MSE training and the ENSURE results. We note that ENSURE is only an unbiased estimate for the MSE, which is an approximation. The approximation becomes exact only when the expectation is taken over the sampling patterns. In our implementation, we assume that each image is sampled with a specific sampling pattern that is fixed during the training process. We also note that the gap in performance of ENSURE and MSE is comparatively higher in the fastMRI setting than in our earlier experiments. We attribute this to the larger variability in image quality between the datasets in fastMRI (likely acquired on different scanners and coils) and the inaccuracy in the noise estimation. We will consider the estimation of noise variance for each subject in the future to close the gap.

The main focus of this paper is to establish the theory and validate it with datasets where fully sampled images are available. We note that this approach is most useful in dynamic MRI and high-resolution static MRI, where the acquisition of fully sampled ground-truth images is not feasible. We will consider the application of ENSURE to these settings in our future work; this is beyond the scope of this preliminary publication.

We note that SURE-based approaches have been recently used in MRI [34]. The main focus of the above paper is on uncertainty quantification using variational auto-encoders, while the authors also use SURE to approximate the MSE loss. By contrast, the main focus of this work is to introduce the ENSURE metric and to elaborately compare it with MSE and state-of-the-art unsupervised methods under different undersampling and noise settings. In addition, the authors approximate the MSE by SURE, assuming the density-weighted undersampling noise to be Gaussian [34]; while the mean of the density-weighted noise term is shown to be zero in [34], it is not sufficient to guarantee that the undersampling-induced noise is Gaussian. By contrast, we define a new metric in (13) which is the average of the errors over datasets and sampling patterns, which is well approximated by the ENSURE metric.

VI. Conclusions

We introduced a novel loss function for the unsupervised training of deep-learning-based image reconstruction algorithms when fully sampled training data is not available. The proposed approach is the extension of the generalized SURE (GSURE) approach. The key distinction of the framework from GSURE is the assumption that different images are measured by different sampling operators. We evaluate the expectation of the GSURE losses over the sampling patterns to obtain the ENSURE loss function, which is an unbiased estimate for the true MSE. Our theoretical results show that the proposed ENSURE loss function closely approximates the true MSE without requiring the fully sampled images. More specifically, the use of the different sampling operators enables us to obtain a better approximation of the loss than GSURE, which approximates the projected MSE; the MSE of the projections of the images to the range space of the measurement operator is a poor approximation of the true MSE, especially in highly undersampled settings. The experiments confirm that the deep learning networks trained using the ENSURE approach without fully sampled training data closely resemble the networks trained using a supervised loss function. While our focus in this work was on an MRI reconstruction problem, this approach is broadly applicable to general inverse problems, including deblurring and tomography.

Acknowledgments

This work is supported by 1R01EB019961-01A1 and 1R01AG067078-01A1. This work was conducted on an MRI instrument funded by 1S10OD025025-01.

Appendix

From (1), we note that $\boldsymbol{u} = \mathscr{A}_s^H \mathbf{C}^{-1} \boldsymbol{y}_s$ is Gaussian distributed with mean $\mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s \boldsymbol{\rho}$ and covariance matrix $\mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s$, i.e.,

$$p(\boldsymbol{u} \mid \boldsymbol{\rho}) = \mathscr{N}\left(\mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s \boldsymbol{\rho}, \mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s\right) \tag{30}$$

$$= q(\boldsymbol{u}) \exp\left(\boldsymbol{\rho}^T \boldsymbol{u} - g(\boldsymbol{\rho})\right). \tag{31}$$

Here, $q(\boldsymbol{u})$ is independent of $\boldsymbol{\rho}$. Similarly, $g(\boldsymbol{\rho})$ is dependent on ρ, which is not often available, and is independent of $\boldsymbol{u}$:

$$q(\boldsymbol{u}) \ = K \exp\left[-\frac{1}{2} \boldsymbol{u}^H \left[\mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s\right]^\dagger \boldsymbol{u}\right] \tag{32}$$

$$g(\boldsymbol{\rho}) = \frac{1}{2} \boldsymbol{\rho}^H \mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s \boldsymbol{\rho} \tag{33}$$

Here, K is a constant. We consider the third term in (14). Denoting $\mathbf{D} = \mathbf{W}_s{}^T\mathbf{W}_s$, we obtain

$$\mathbb{E}_{\boldsymbol{u}}\left[\hat{\boldsymbol{\rho}}^T \mathbf{D} \boldsymbol{\rho}\right] = \int_{-\infty}^{+\infty} \hat{\boldsymbol{\rho}}^T \mathbf{D} \boldsymbol{\rho} \underbrace{q(\boldsymbol{u}) \overbrace{\exp\left(\boldsymbol{\rho}^T \boldsymbol{u} - g(\boldsymbol{\rho})\right)}^{h(\boldsymbol{u})}}_{p(\boldsymbol{u})} d\boldsymbol{u}$$

Let $h(\boldsymbol{u}) = \exp\left(\boldsymbol{\rho}^T \boldsymbol{u} - g(\boldsymbol{\rho})\right)$, then $\nabla_u h(\boldsymbol{u}) = \boldsymbol{\rho}\, h(\boldsymbol{u})$. Substituting for $\boldsymbol{\rho}\, h\, (\boldsymbol{u})$, we obtain

$$\begin{aligned} \mathbb{E}_u\left[\hat{\boldsymbol{\rho}}^T \mathbf{D} \boldsymbol{\rho}\right] \ &= \int_{-\infty}^{+\infty} \widehat{\boldsymbol{\rho}}^T \mathbf{D} q(\boldsymbol{u}) \nabla_u h(\boldsymbol{u}) d\boldsymbol{u} \\ &= \langle q(\boldsymbol{u}) \mathbf{D} \widehat{\boldsymbol{\rho}}, \nabla_u h(\boldsymbol{u}) \rangle \\ &= \ - \left\langle h(\boldsymbol{u}), \nabla_u \cdot [q(\boldsymbol{u}) \mathbf{D} \hat{\boldsymbol{\rho}}] \right\rangle \end{aligned}$$

Expanding the second term in the inner-product using the chain rule, we obtain

$$\begin{aligned} \nabla_u \cdot [q(\boldsymbol{u}) \mathbf{D} \widehat{\boldsymbol{\rho}}] &= q(\boldsymbol{u}) \nabla_u \cdot \mathbf{D} \widehat{\boldsymbol{\rho}} + \mathbf{D} \widehat{\boldsymbol{\rho}} \cdot \nabla_u q(\boldsymbol{u}) \\ &= q(\boldsymbol{u}) \left(\nabla_u \cdot \mathbf{D} \hat{\boldsymbol{\rho}} + \mathbf{D} \widehat{\boldsymbol{\rho}} \cdot \frac{\nabla_u q(\boldsymbol{u})}{q(\boldsymbol{u})} \right) \\ &= q(\boldsymbol{u}) \left(\nabla_u \cdot \mathbf{D} \hat{\boldsymbol{\rho}} + \mathbf{D} \widehat{\boldsymbol{\rho}} \cdot \underbrace{\nabla_u \ln q(\boldsymbol{u})}_{-\boldsymbol{\rho}_{\mathrm{LS}}} \right) \end{aligned}$$

In the last step, we used the property

$$-\nabla_u \ln q(\boldsymbol{u}) = \left[\mathscr{A}_s^H \mathbf{C}^{-1} \mathscr{A}_s\right]^\dagger \mathscr{A}_s^H \mathbf{C}^{-1} \boldsymbol{y}_s = \boldsymbol{\rho}_{\mathrm{LS}} \tag{34}$$

Because $q(\boldsymbol{u}) h(\boldsymbol{u}) = p(\boldsymbol{u})$, we obtain

$$\mathbb{E}_{\boldsymbol{u}}\left[\hat{\rho}^{T}\mathbf{D}\rho\right] = -\mathbb{E}_{\boldsymbol{u}}\left[\nabla_{\boldsymbol{u}}\cdot\mathbf{D}\hat{\rho}\right] + \mathbb{E}_{\boldsymbol{u}}\left[\hat{\rho}^{T}\mathbf{D}\rho_{\mathrm{LS}}\right]$$

References

[1]. Lustig M, Donoho DL, Santos JM et al. , “Compressed sensing MRI,” IEEE Signal Process. Mag, vol. 25, no. 2, p. 72, 2008.

[2]. Candes E and Romberg J, “Sparsity And Incoherence In Compressive Sampling,” Inverse Probl, vol. 23, no. 3, p. 969, 2007.

[3]. Fessler JA, “Model-Based Image Reconstruction for MRI,” IEEE Signal Process. Mag, vol. 27, no. 4, pp. 81–89, 2010. [PubMed: 21135916]

[4]. Chen H, Zhang Y, Kalra MK et al. , “Low-Dose CT with a Residual Encoder-Decoder Convolutional Neural Network,” IEEE Trans. Med. Imag, vol. 36, no. 12, pp. 2524–2535, 2017.

[5]. Han Y, Sunwoo L, and Ye JC, “k-space deep learning for accelerated MRI,” IEEE Trans. Med. Imag, 2019.

[6]. Aggarwal HK, Mani MP, and Jacob M, “MoDL: Model Based Deep Learning Architecture for Inverse Problems,” IEEE Trans. Med. Imag, vol. 38, no. 2, pp. 394–405, 2019.

[7]. Yang Y, Sun J, Li H et al. , “Deep ADMM-Net for Compressive Sensing MRI,” in Advances in Neural Information Processing Systems 29, 2016, pp. 10–18.

[8]. Adler J and Oktem O, “Learned primal-dual reconstruction,”¨ IEEE Trans. Med. Imag, vol. 37, no. 6, pp. 1322–1332, 2018.

[9]. Hammernik K, Klatzer T, Kobler E et al. , “Learning a Variational Network for Reconstruction of Accelerated MRI Data,” Magnetic resonance in Medicine, vol. 79, no. 6, pp. 3055–3071, 2017. [PubMed: 29115689]

[10]. Zhang L and Zuo W, “Image Restoration: From Sparse and Low-Rank Priors to Deep Priors,” IEEE Signal Process. Mag, vol. 34, no. 5, pp. 172–179, 2017.

[11]. Yang G, Yu S, Dong H et al. , “DAGAN: Deep De-Aliasing Generative Adversarial Networks For Fast Compressed Sensing MRI Reconstruction,” IEEE Trans. Med. Imag, vol. 37, no. 6, pp. 1310–1321, 2017.

[12]. Zbontar J, Knoll F, Sriram A et al. , “fastMRI: An open dataset and benchmarks for accelerated MRI,” arXiv preprint arXiv:1811.08839, 2018.

[13]. Hansen PC and O’Leary DP, “The Use Of The L-Curve In The Regularization Of Discrete Ill-Posed Problems,” SIAM journal on scientific computing, vol. 14, no. 6, pp. 1487–1503, 1993.

[14]. Golub GH, Heath M, and Wahba G, “Generalized Cross-Validation As a Method For Choosing a Good Ridge Parameter,” Technometrics, vol. 21, no. 2, pp. 215–223, 1979.

[15]. Stein CM, “Estimation Of The Mean Of A Multivariate Normal Distribution,” The annals of Statistics, pp. 1135–1151, 1981.

[16]. Donoho DL and Johnstone IM, “Adapting To Unknown Smoothness Via Wavelet Shrinkage,” Journal of the American Statistical Association, vol. 90, no. 432, pp. 1200–1224, 1995.

[17]. Blu T and Luisier F, “The SURE-LET Approach To Image Denoising,” IEEE Trans. Image Process, vol. 16, no. 11, pp. 2778–2786, 2007. [PubMed: 17990754]

[18]. Luisier F, Vonesch C, Blu T et al. , “Fast Interscale Wavelet Denoising Of Poisson-Corrupted Images,” Signal processing, vol. 90, no. 2, pp. 415–427, 2010.

[19]. Metzler CA, Mousavi A, Heckel R et al. , “Unsupervised Learning with Stein’s Unbiased Risk Estimator,” arXiv preprint arXiv:1805.10531, 2018.

[20]. Zhussip M, Soltanayev S, and Chun SY, “Extending Stein’s Unbiased Risk Estimator to Train Deep Denoisers with Correlated pPairs of Noisy Images,” in Advances in Neural Information Processing Systems, 2019.

[21]. Zhussip M, Soltanayev S, and Chun SY, “Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10255–10264.

[22]. Eldar YC, “Generalized SURE for Exponential Families: Applications to Regularization,” IEEE Trans. Signal Process, vol. 57, no. 2, pp. 471–481, 2008.

[23]. Abu-Hussein S, Tirer T, Chun SY et al., “Image restoration by deep projected GSURE,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 3602–3611.

[24]. Lehtinen J, Munkberg J, Hasselgren J et al., “ Noise2Noise: Learning Image Restoration without Clean Data,” in International Conference on Machine Learning, 2018, pp. 2965–2974.

[25]. Krull A, Buchholz T-O, and Jug F, “ Noise2Void-Learning Denoising From Single Noisy Images,” in IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2129–2137.

[26]. Yaman B, Hosseini SAH, Moeller S et al. , “Self-Supervised Learning Of Physics-Guided Reconstruction Neural Networks Without Fully Sampled Reference Data,” Magn. Reson. Med, pp. 3172–3191, 2020. [PubMed: 32614100]

[27]. Krull A, Vičar T, Prakash M et al. , “Probabilistic Noise2Void: Unsupervised Content-Aware Denoising,” Frontiers in Computer Science, vol. 2, p. 5, 2020.

[28]. Ramani S, Blu T, and Unser M, “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. [PubMed: 18701393]

[29]. Ulyanov D, Vedaldi A, and Lempitsky V, “Deep Image Prior,” in IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.

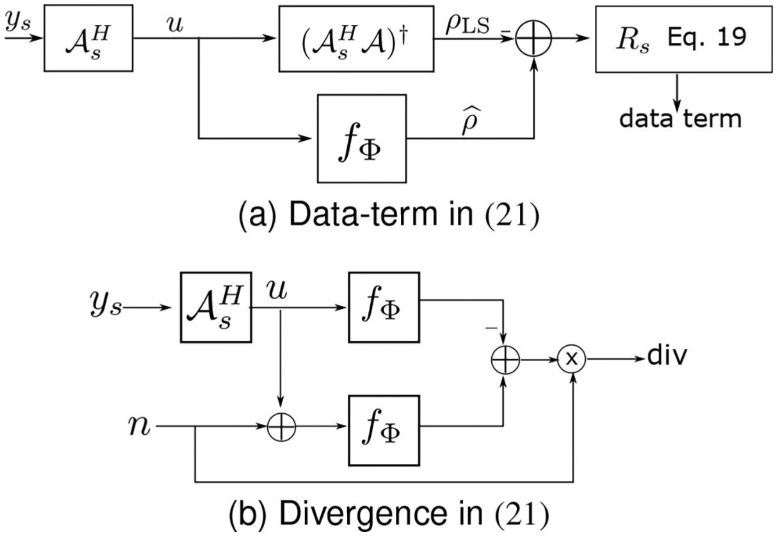
[30]. Pruessmann KP, Weiger M, Scheidegger MB et al. , “SENSE: Sensitivity encoding for fast MRI,” Magnetic Resonance in Medicine, vol. 42, no. 5, pp. 952–962, 1999. [PubMed: 10542355]

[31]. Uecker M, Lai P, Murphy MJ et al. , “ESPIRiT - An eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA,” Magnetic Resonance in Medicine, vol. 71, no. 3, pp. 990–1001, 2014. [PubMed: 23649942]

[32]. Wang Z, Bovik AC, Sheikh HR et al. , “Image Quality Assessment: From Error Visibility To Structural Similarity,” IEEE Trans. Image Process, vol. 13, no. 4, pp. 600–612, 2004. [PubMed: 15376593]

[33]. Aggarwal HK, Mani MP, and Jacob M, “MoDL-MUSSELS: Model-based deep learning for multishot sensitivity-encoded diffusion MRI,” IEEE Trans. Med. Imag, 2019.

[34]. Edupuganti V, Mardani M, Vasanawala S et al. , “Uncertainty quantification in Deep MRI reconstruction,” IEEE Transactions on Medical Imaging, vol. 40, no. 1, pp. 239–250, 2021. [PubMed: 32956045]



(a) Data-term in (21)

(b) Divergence in (21)

Fig. 1.

Visual representation of the computation of the data and divergence terms in the proposed ENSURE estimate in (21). Here, ⊕ and ⊗ represent the addition and inner product, respectively.

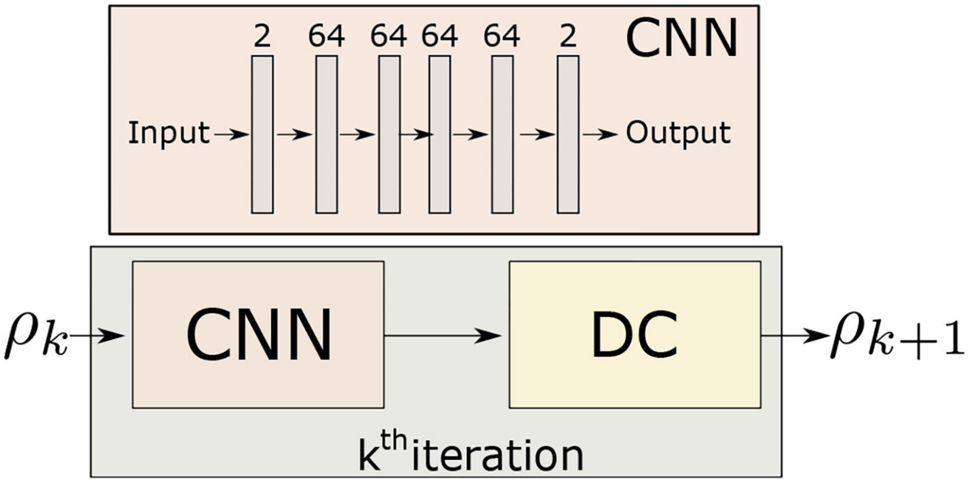


Fig. 2.

The specific network architecture used in the experiments. The CNN is a five-layer model that takes complex data as input. Each of the middle layers has 64 feature maps. It concatenates real and imaginary components as channels in the first layer. Similarly, the last layer output is converted to the complex type before it enters the DC step.

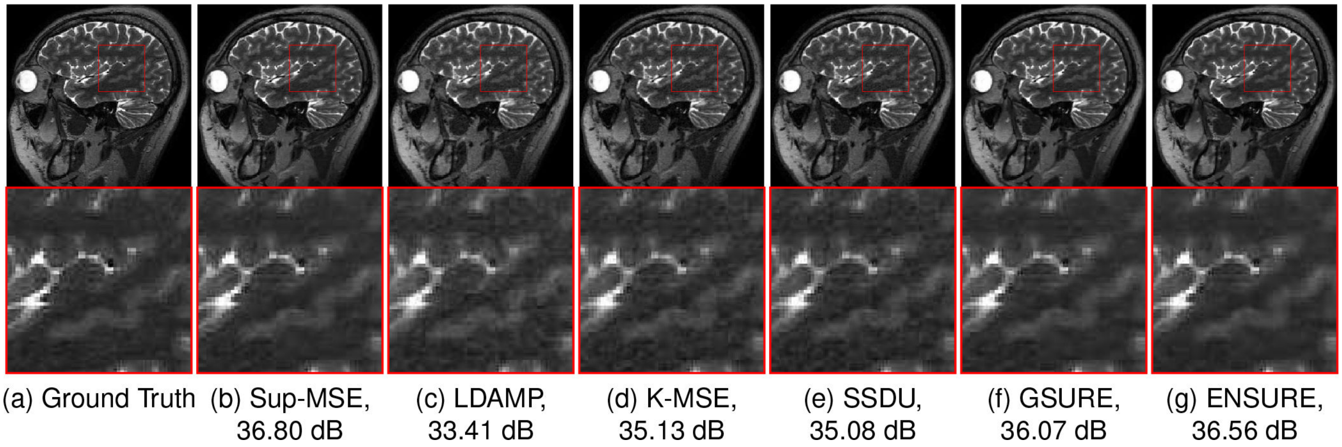


Fig. 3.

This figure compares reconstruction results at 4x acceleration on a test slice. We can observe from the zoomed region that the proposed ENSURE approach results in comparable reconstruction quality to that of supervised training with MSE (Sup-MSE). Here, LDAMP refers to the LDAMP-SURE algorithm.

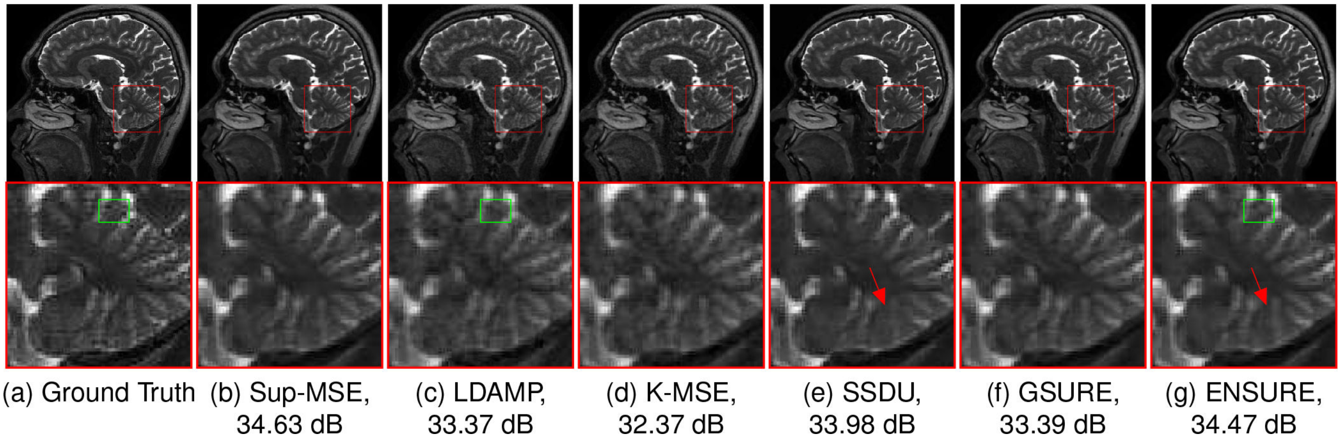


Fig. 4.

Reconstruction results at a higher acceleration factor of 6x. We observe that, even at this high acceleration, the proposed ENSURE approach results in similar quantitative and qualitative reconstruction quality to that of supervised MSE training. The arrow in the zoomed region points to a feature well captured using ENSURE as compared to SSDU.

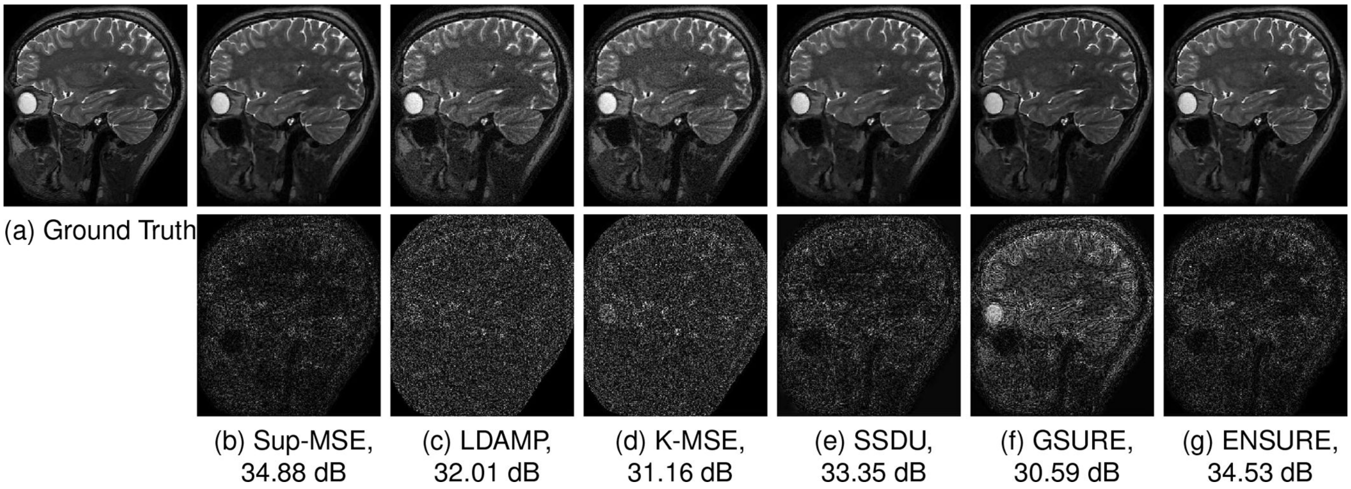


Fig. 5.

Reconstruction results at 4x acceleration with added Gaussian noise of $\sigma = 0.03$ in the measurements. The bottom row shows the corresponding error maps. The magnitude of error images is multiplied by four for visualization purposes.

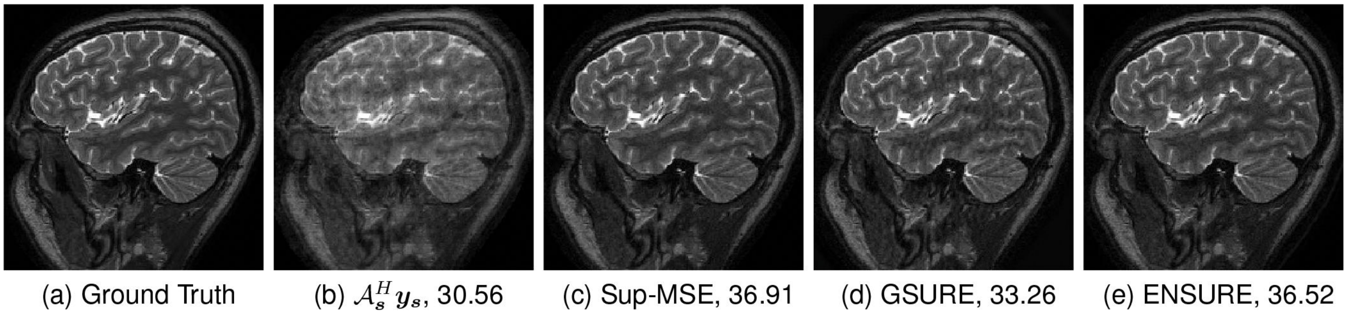
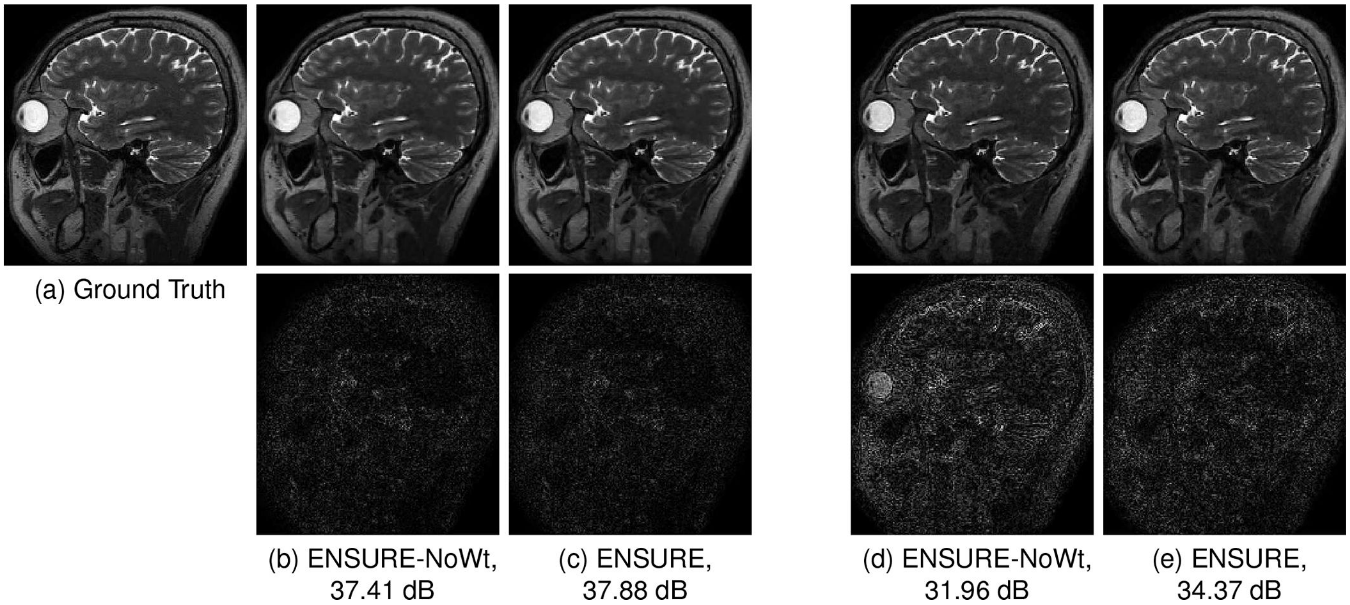

(a) Ground Truth
(b) $\mathcal{A}_{\boldsymbol{s}}^{H}\boldsymbol{y}_{\boldsymbol{s}}$, 30.56
(c) Sup-MSE, 36.91
(d) GSURE, 33.26
(e) ENSURE, 36.52

Fig. 6.

This example demonstrates the benefit of sampling an image with multiple measurement operators in the one-dimensional sampling case at a 2X acceleration factor with added Gaussian noise of $\boldsymbol{\sigma} = 0.02$. The numbers in the sub-captions show PSNR (dB) values.



(a) Ground Truth

(b) ENSURE-NoWt, 37.41 dB

(c) ENSURE, 37.88 dB

(d) ENSURE-NoWt, 31.96 dB

(e) ENSURE, 34.37 dB

Fig. 7.

This figure shows the benefit of the proposed weighting strategy. It visually compares the reconstruction quality of ENSURE with and without the proposed weighting scheme. The magnitude of error images is multiplied by four for visualization purposes. (b) and (c) refers to the case when no simulated noise is added to the measurements. (d) and (e) refers to the experimental setup when we added Gaussian noise of $\sigma = 0.03$ to the measurements.

TABLE I

Quantitative comparison ofpeak signal-to-noise ratio (PSNR) (dB) and structural similarity index (SSIM) values. Each value shows mean ± std.

	Peak Signal to Noise Ratio (PSNR)		
acceleration	4x	4x, $\sigma = 0.03$	6x
E-to-E-Sup	37.71 ± 0.97	35.34 ± 0.70	35.39 ± 0.92
L-by-L-Sup	36.64 ± 1.07	35.29 ± 1.29	35.26 ± 0.99
GSURE	36.80 ± 1.04	31.10 ± 1.40	33.68 ± 1.04
K-MSE	35.29 ± 1.12	31.67 ± 0.76	33.16 ± 1.04
SSDU	35.46 ± 1.03	33.48 ± 0.71	33.92 ± 1.15
LDAMP-SURE	34.97 ± 1.23	33.56 ± 1.39	32.87 ± 1.43
ENSURE	37.36 ± 0.92	34.13 ± 0.90	34.98 ± 0.93
	Structural Similarity Index (SSIM)		
E-to-E-Sup	0.97 ± 0.00	0.95 ± 0.01	0.94 ± 0.01
L-by-L- Sup	0.95 ± 0.00	0.95 ± 0.00	0.94 ± 0.00
GSURE	0.93 ± 0.01	0.91 ± 0.01	0.92 ± 0.02
K-MSE	0.95 ± 0.01	0.86 ± 0.02	0.93 ± 0.01
SSDU	0.93 ± 0.02	0.88 ± 0.04	0.92 ± 0.01
LDAMP-SURE	0.96 ± 0.00	0.94 ± 0.01	0.93 ± 0.01
ENSURE	0.97 ± 0.00	0.93 ± 0.01	0.94 ± 0.01

TABLE II

Quantitative results on fast MRI dataset. The values represent the average of PSNR(dB) ± standard deviations on the test Dataset.

	SENSE: U	ADMM-net: U	FISTA-net: U	MoDL: U	MoDL-MS	MoDL VD1	MoDL-VD2
MSE	38.95 4.2 0.965 0.02	42.09 3.8 0.983 0.01	42.04 3.8 0.981 0.01	42.30 3.8 0.985 0.01	42.38 3.9 0.985 0.01	43.90 3.7 0.99 0.01	44.45 3.9 0.990 0.01
ENSURE		39.99 4.0 0.972 0.02	40.24 3.8 0.972 0.02	40.75 3.7 0.973 0.02	40.67 3.9 0.973 0.02	41.69 3.9 0.980.01	42.17 4.0 0.983 0.01