# A Theoretical Framework for Self-Supervised MR Image Reconstruction Using Sub-Sampling via Variable Density Noisier2Noise

Charles Millard and Mark Chiew

***Abstract*—In recent years, there has been attention on leveraging the statistical modeling capabilities of neural networks for reconstructing sub-sampled Magnetic Resonance Imaging (MRI) data. Most proposed methods assume the existence of a representative fully-sampled dataset and use fully-supervised training. However, for many applications, fully sampled training data is not available, and may be highly impractical to acquire. The development and understanding of self-supervised methods, which use only sub-sampled data for training, are therefore highly desirable. This work extends the Noisier2Noise framework, which was originally constructed for self-supervised denoising tasks, to variable density sub-sampled MRI data. We use the Noisier2Noise framework to analytically explain the performance of Self-Supervised Learning via Data Undersampling (SSDU), a recently proposed method that performs well in practice but until now lacked theoretical justification. Further, we propose two modifications of SSDU that arise as a consequence of the theoretical developments. Firstly, we propose partitioning the sampling set so that the subsets have the same type of distribution as the original sampling mask. Secondly, we propose a loss weighting that compensates for the sampling and partitioning densities. On the fastMRI dataset we show that these changes significantly improve SSDU's image restoration quality and robustness to the partitioning parameters.**

***Index Terms*—Deep learning, image reconstruction, magnetic resonance imaging.**

## I. Introduction

THE data acquisition process in Magnetic Resonance Imaging (MRI) consists of traversing a sequence of smooth paths through the Fourier representation of the image, referred to as "k-space", which is inherently time-consuming. Images can be reconstructed from accelerated, sub-sampled acquisitions by leveraging the non-uniformity of receiver coil sensitivities, referred to as "parallel imaging" [1], [2], [3], [4]. Compressed sensing [5], [6], which uses sparse models to reconstruct incoherently sampled data, has also been widely applied to MRI [7], [8], [9].

There has been significant research attention in recent years on methods that reconstruct sub-sampled MRI data with neural networks [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. The majority of these works use fully-supervised training. To train a network in a fully-supervised manner, there must be a dataset comprised of fully sampled k-space data $y_{0,t} \in \mathbb{C}^N$, where $N$ is the dimension of k-space multiplied by the number of coils, and paired sub-sampled data $y_t = M_{\Omega_t} y_{0,t}$. Here, $t$ indexes the training set and $M_{\Omega_t} \in \mathbb{R}^{N \times N}$ is a sub-sampling mask with sampling set $\Omega_t$, so that the $j$th diagonal of $M_{\Omega_t}$ is 1 if $j \in \Omega_t$ and zero otherwise. Then a network $f_\theta$ with parameters $\theta$ is trained by seeking a minimum of a non-convex loss function:

$$\hat{\theta} = \arg\min_{\theta} \sum_{t} L(f_\theta(y_t), y_{0,t}), \tag{1}$$

which could be, for example, an $\ell_p$ norm in the image domain after coil combination [25]. The network $f_{\hat{\theta}}$ estimates the ground truth in the image domain or k-space depending on the choice of loss function. For a k-space to k-space network, $y_{0,s}$ can be estimated with $\hat{y}_s = f_{\hat{\theta}}(y_s)$, where $s$ indexes the test set.

Given sufficient representative training data, fully-supervised networks can yield substantial reconstruction quality gains over sparsity-based compressed sensing methods. There are a number of large datasets available for fully supervised training, such as the fastMRI knee and brain data [25]. However, for many contrasts, orientations, or anatomical regions of interest, fully sampled datasets are not publicly available. Fully sampled data is rarely acquired as part of a normal scanning protocol, so acquiring sufficient training data for a specific application is highly resource intensive. In some cases, it may not even be technically feasible to acquire such data [26], [27], [28]. Therefore, for MRI reconstruction with deep learning to be applicable to datasets acquired using only standard protocols, a training method that uses solely sub-sampled data is required.

Manuscript received 7 June 2022; revised 19 December 2022, 25 April 2023, 21 June 2023, and 29 June 2023; accepted 14 July 2023. Date of publication 26 July 2023; date of current version 9 August 2023. This work was supported in part by Engineering and Physical Sciences Research Council under Grant EP/T013133/1, in part by the Royal Academy of Engineering under Grant RF201617/16/23, in part by Wellcome Trust under Grant 203139/Z/16/Z, and in part by Canada Research Chairs Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Se Young Chun. *(Corresponding author: Charles Millard.)*

Charles Millard is with the Wellcome Centre for Integrative Neuroimaging, FMRIB, Nuffield Department of Clinical Neurosciences, University of Oxford, OX3 9DU Oxford, U.K. (e-mail: charles.millard@ndcn.ox.ac.uk).

Mark Chiew is with the Wellcome Centre for Integrative Neuroimaging, FMRIB, Nuffield Department of Clinical Neurosciences, University of Oxford, OX3 9DU Oxford, U.K., and with the Department of Medical Biophysics, University of Toronto, Toronto, ON M5S 1A1, Canada, and also with the Canada and Physical Sciences, Sunnybrook Research Institute, Toronto, ON M4N 3M5, Canada (e-mail: mark.chiew@utoronto.ca).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2023.3299212, provided by the authors.

Digital Object Identifier 10.1109/TCI.2023.3299212

There have been several attempts to train networks with only sub-sampled MRI data [29], [30], [31], [32], [33], [34], [35], [36], [37], some of which are based on methods from the denoising literature [38], [39], [40], [41], [42], [43], [44]. One such approach is Noise2Noise [38]. Rather than mapping $y_t$ to $y_{0,t}$, Noise2Noise trains a network to map $y_t$ to another sub-sampled k-space $y_T = M_{\Omega_T} y_{0,T}$ where $\Omega_T$ and $\Omega_t$ are independent and $y_{0,T} = y_{0,t}$ when $t = T$ [31]. A limitation of Noise2Noise is that it requires paired data, so the dataset must contain two independently sampled scans of the same k-space [14], which is not part of standard protocols. Further, unless compensated for [45], any motion and phase drifts between scans would cause the paired data to be inconsistent, violating the central assumption that underlies the method.

SSDU [33] is a recently proposed method for ground-truth free training that does not require paired data. SSDU partitions the sampling set $\Omega_t$ into two disjoint sets: $\Omega_t = A_t \cup B_t$, where $A_t \cap B_t = \emptyset$. Then the network is trained to recover $M_{A_t} y_t$ from $M_{B_t} y_t$:

$$\hat{\theta} = \arg\min_{\theta} \sum_t L(M_{A_t} f_\theta(M_{B_t} y_t), M_{A_t} y_t). \tag{2}$$

At inference, the estimate $f_{\hat{\theta}}(y_s)$ is used. With a physics-guided network architecture, SSDU was found to have a reconstruction quality comparable with fully supervised training given certain empirically selected choices of $A_t$ and $B_t$. However, it was presented without theoretical justification. Although SSDU has similarities with Noise2Self [40], Noise2Self's analysis has a strong requirement on independent noise, so do not apply to k-space sampling in general.

### A. Contributions

This article considers the recently proposed Noisier2Noise framework [41], which was originally constructed for denoising problems. We modify Noisier2Noise so that it can be applied to variable density sub-sampled MRI data. To our knowledge, this is the first work that applies Noisier2Noise to image reconstruction. Like SSDU, the proposed modification of Noisier2Noise does not require paired data, and involves training a network to map from one subset of $\Omega_t$ to another. While SSDU recovers one disjoint set from the other, Noisier2Noise applies a second sub-sampling mask to the data, $\widetilde{y}_t = M_{\Lambda_t} y_t = M_{\Lambda_t} M_{\Omega_t} y_{0,t}$, and the network is trained to recover $y_t$ from $\widetilde{y}_t$ with an $\ell_2$ loss. Then, at inference, the fully sampled data is estimated via a correction term based on the distributions of $\Lambda_t$ and $\Omega_t$ that ensures that the estimate is correct in expectation.

Despite their superficial differences, we show that, in fact, SSDU and Noisier2Noise are closely related. Specifically, we demonstrate that SSDU is a version of Noisier2Noise with a particular loss function modification that removes the need for the correction term at inference. The primary contribution of this article is the use of Noisier2Noise to theoretically explain SSDU's excellent empirical performance. Specifically, we show that SSDU with an $\ell_2$ loss correctly estimates fully sampled k-space in expectation: see Section II-D.

The second contribution of this article is the proposal of two modifications of SSDU that significantly improve its reconstruction quality and robustness to the parameters of $M_{\Lambda_t}$, both of which arise as a consequence of SSDU's connection to Noisier2Noise. Firstly, we use Noisier2Noise to inform SSDU's sampling set partition: we show that SSDU's performance improves when $B_t$ has the same type of distribution as the original mask $\Omega_t$, but not necessarily with the same parameters. Secondly, we show that SSDU's performance improves when a particular weighting is employed in the loss function. This non-trivial weighting, which arises as a consequence of the novel theoretical analysis of SSDU, depends on the distributions of $\Lambda_t$ and $\Omega_t$ and has minimal additional computational cost: see Section II-F.

Although this paper focuses on MRI reconstruction, we emphasize that none of the theoretical developments are specific to k-space. This framework is therefore applicable to any image reconstruction problem with a forward model that involves random sub-sampling, such as low dose x-ray computed tomography [46] or astronomical imaging [47].

## II. Theory

This section describes how the Noisier2Noise framework can be applied to sub-sampled data. Additive and multiplicative noise versions of Noisier2Noise are proposed in [41]. Based on the observation that a k-space sub-sampling mask can be considered as multiplicative "noise", we extend Noisier2Noise to image reconstruction by modifying the latter. It is standard practice in MRI to sub-sample k-space with variable density, so that low frequencies, where the spectral density is larger, are sampled with higher probability [7]. Since the multiplicative noise version of standard Noisier2Noise assumes uniformity, this requires a modification of the framework to variable density sampling.

### A. Variable Density Noisier2Noise for Reconstruction

The terms in the measurement model $y_t = M_{\Omega_t} y_{0,t}$ can be considered as instances of random variables. We denote $Y = M_\Omega Y_0$, where $Y$, $M_\Omega$ and $Y_0$ are the random variables corresponding to $y_t$, $M_{\Omega_t}$, and $y_{0,t}$ respectively. Now consider the multiplication of $Y$ by a second mask represented by the random variable $M_\Lambda$,

$$\widetilde{Y} = M_\Lambda Y = M_\Lambda M_\Omega Y_0,$$

so that $\widetilde{Y}$ is a further sub-sampled random variable. The following result states how the expectation of $Y_0$ can be computed from $\widetilde{Y}$ and $Y$. Here, and throughout this article, $\mathbb{E}[\cdot]$ is used to denote the expectation over all random variables within the brackets.

*Claim 1:* When $\mathbb{E}[M_{\Omega,jj}] = p_j > 0$ and $\mathbb{E}[M_{\Lambda,jj}] = \widetilde{p}_j < 1$ for all $j$, the expectation of $Y_0$ given $\widetilde{Y}$ is

$$\mathbb{E}[Y_0|\widetilde{Y}] = (\mathbb{1} - K)^{-1}(\mathbb{E}[Y|\widetilde{Y}] - K\widetilde{Y}), \tag{3}$$

where $K$ is a diagonal matrix defined as

$$K = (\mathbb{1} - \widetilde{P}P)^{-1}(\mathbb{1} - P) \tag{4}$$

for $P = \mathbb{E}[M_\Omega]$ and $\widetilde{P} = \mathbb{E}[M_\Lambda]$.

*Proof:* See Section A of the Appendix, which is based on the proof given in Section III.D of [41]. □

Equation (3) generalizes the version of Noisier2Noise proposed for uniform, multiplicative noise in [41] to variable density sampling. The difference between the uniform and variable density versions is the matrix $K$, which is a scalar in [41]. For the special case where $M_\Omega$ and $M_\Lambda$ are uniformly random sub-sampling masks, $P$, $\widetilde{P}$ and therefore $K$ are proportional to the identity matrix, and (3) simplifies to the uniform version. The mathematical requirement that $p_j > 0$ and $\widetilde{p}_j < 1$ for all $j$ simply ensures that $(\mathbb{1} - K)$ is invertible: see Section A of the Appendix.

Equation (3) implies that $\mathbb{E}[Y_0|\widetilde{Y}]$ can be estimated without fully sampled data by training a network to estimate $\mathbb{E}[Y|\widetilde{Y}]$. To do this, a network can be trained to minimize

$$\theta^* = \arg\min_\theta \mathbb{E}[\|W(f_\theta(\widetilde{Y}) - Y)\|_2^2|\widetilde{Y}] \tag{5}$$

for a full-rank matrix $W$. The minimum occurs when the gradient with respect to $\theta$ is zero:

$$\nabla_\theta \mathbb{E}[\|W(f_\theta(\widetilde{Y}) - Y)\|_2^2|\widetilde{Y}] = \mathbb{E}[2JW^HW(f_\theta(\widetilde{Y}) - Y)|\widetilde{Y}] = 0,$$

where $J$ is the Jacobian matrix with entries $J_{ij} = \partial f_\theta(\widetilde{Y})_j/\partial\theta_i$. The number of parameters is typically much greater than $N$, so $J$ has far more rows than columns. Assuming that the rows of $J$ are maximally linearly independent, so the row space is $N$-dimensional, the only solution is

$$\mathbb{E}[W^HW(f_\theta(\widetilde{Y}) - Y)|\widetilde{Y}] = 0. \tag{6}$$

If $W$ is full-rank, $W^HW$ is also full rank, so left-multiplying by $(W^HW)^{-1}$ and using $\mathbb{E}[f_\theta(\widetilde{Y})|\widetilde{Y}] = f_\theta(\widetilde{Y})$,

$$f_\theta(\widetilde{Y}) = \mathbb{E}[Y|\widetilde{Y}].$$

Therefore, by (3), a candidate for estimating fully sampled k-space with sub-sampled data only is

$$\mathbb{E}[Y_0|\widetilde{Y}] = (\mathbb{1} - K)^{-1}(f_{\theta^*}(\widetilde{Y}) - K\widetilde{Y}).$$

This expression does not use $Y$, so does not use all available data. Two candidate approaches for using all available data at inference are considered in this article. Firstly, one can overwrite known entries of the network output with $Y$:

$$\begin{aligned}\hat{Y}^{dc} &= (\mathbb{1} - M_\Omega)\mathbb{E}[Y_0|\widetilde{Y}] + Y \\ &= (\mathbb{1} - M_\Omega)(\mathbb{1} - K)^{-1}(f_{\theta^*}(\widetilde{Y}) - K\widetilde{Y}) + Y \\ &= (\mathbb{1} - M_\Omega)(\mathbb{1} - K)^{-1}f_{\theta^*}(\widetilde{Y}) + Y,\end{aligned}$$

where the final step uses $(\mathbb{1} - M_\Omega)\widetilde{Y} = (\mathbb{1} - M_\Omega)M_\Lambda M_\Omega Y_0 = 0$. Here, the superscript refers to "data consistent", since the estimate is exactly consistent with $Y$. We emphasize that $\hat{Y}^{dc}$ is consistent with *all* available data $Y$, not just the data in $\widetilde{Y}$. Alternatively, similar to the approaches suggested in both SSDU [33] and the additive noise examples in Noisier2Noise [41], one can use singly sub-sampled k-space $Y$ as the network input at inference:

$$\hat{Y} = (\mathbb{1} - K)^{-1}(f_{\theta^*}(Y) - KY) \tag{7}$$

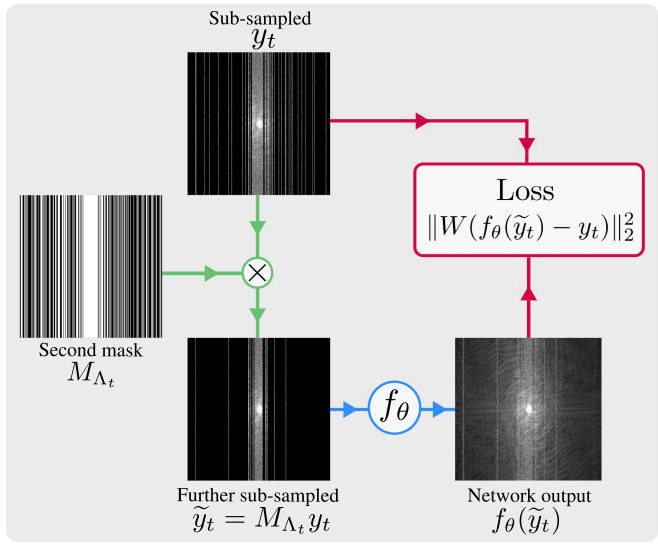



Fig. 1. Schematic of the self-supervised training methods in this article. If the loss weighting $W$ is full rank, the training method is variable density Noisier2Noise, as proposed in Section II-A, whereas if $W = (\mathbb{1} - M_{\Lambda_t})M_{\Omega_t}$ the training method is SSDU: see Section II-D.

Since Claim 1 applies to $f_{\theta^*}(\widetilde{Y})$, not $f_{\theta^*}(Y)$, (7) is not guaranteed to be correct in expectation. However, it has the advantage that all available data is used by the network. Hence, despite deviating from strict theory, we have found that it performs well in practice: see Section IV.

This suggests the following procedure, illustrated in Fig. 1, for training a network without fully-sampled data. For each sub-sampled k-space in the training set $y_t = M_{\Omega_t}y_{0,t}$, generate a further sub-sampled k-space $\widetilde{y}_t = M_{\Lambda_t}y_t = M_{\Lambda_t}M_{\Omega_t}y_{0,t}$, where $M_{\Lambda_t}$ is an instance of $M_\Lambda$. Then, approximate (5) by training a network to minimize the loss function

$$\hat{\theta} = \arg\min_\theta \sum_t \|W(f_\theta(\widetilde{y}_t) - y_t)\|_2^2, \tag{8}$$

for some full-rank matrix $W$. During inference, estimate fully-sampled k-space with either

$$\hat{y}_s^{dc} = (\mathbb{1} - M_{\Omega_s})(\mathbb{1} - K)^{-1}f_{\hat{\theta}}(\widetilde{y}_s) + y_s \tag{9}$$

or

$$\hat{y}_s = (\mathbb{1} - K)^{-1}(f_{\hat{\theta}}(y_s) - Ky_s), \tag{10}$$

where $s$ indexes the test set.

In other words, we train a network to estimate the "singly" sub-sampled k-space $y_t$ from "doubly" sub-sampled k-space $\widetilde{y}_t$ and then, during inference, apply a correction based on the diagonal matrix $K$ to estimate the fully sampled data. The correction term only needs to be applied during inference and has minimal computational cost.

In [41], only the version with $W = \mathbb{1}$ was presented. Here we present a version with non-trivial $W$ because it provides a theoretical link to SSDU; Section II-D shows that Noisier2Noise with the rank-deficient $W = (\mathbb{1} - M_\Lambda)M_\Omega$ is SSDU exactly.

Noisier2Noise and SSDU work because the network cannot deduce from $\widetilde{y}_t$ which entries of $y_t$ are non-zero [41]. Therefore, the loss is minimized when the network learns to recover *all* of k-space: see Section V for a detailed discussion.

### B. Choice of Mask Distributions

The only condition on the first mask $M_\Omega$ from Claim 1 is that $p_j > 0$ for all $j$. In other words, the guarantee only applies when there is a non-zero probability that there are sampled examples of all k-space locations in the training set.

Claim 1 also states that the second mask $M_\Lambda$ must obey $\widetilde{p}_j < 1$ for all $j$. This ensures that there is a non-zero probability that any entry of $\widetilde{Y}$ is masked. Unlike $M_\Omega$, whose distribution is determined by the acquisition protocol, the $M_\Lambda$ is chosen freely during training. Following [41], we suggest using a distribution of $M_\Lambda$ that is the same type as $M_\Omega$, but not necessarily with the same parameters. For instance, if $M_\Omega$ is column-wise sampling with variable density, such as in Fig. 1, an appropriate $M_\Lambda$ is one that is also column-wise, but possibly with a different variable density distribution.

### C. Choice of Network

Noisier2Noise is agnostic to the network architecture. We have found that using the data consistent function

$$f_\theta(\widetilde{y}_t) = (\mathbb{1} - M_{\Lambda_t} M_{\Omega_t}) g_\theta(\widetilde{y}_t) + \widetilde{y}_t, \tag{11}$$

where $g_\theta(\widetilde{y}_t)$ is a network with arbitrary architecture, may improve the performance of Noisier2Noise. This is because the $g_\theta(\widetilde{y}_t)$ in (11) only recovers regions of k-space that are not already sampled in $\widetilde{y}_t$, so the network does not need to learn to map sampled k-space locations to themselves. We emphasize that (11) ensures that $f_\theta(\widetilde{y}_t)$ is consistent with $\widetilde{y}_t$, while (9) ensures the estimate $\hat{y}_s^{dc}$ is consistent with $y_s$, which is only applied at inference and cannot be part of the network architecture when $\widetilde{y}_s$ is used as the input.

Many popular network architectures for MRI reconstruction are based on a sequence of "unrolled" iterations of a optimization algorithm [48] such as the Iterative Shrinkage Thresholding Algorithm (ISTA) [49] or the Alternating Direction Method of Multipliers (ADMM) [50]. These are variously known as "physics-guided", "physics-based" or "model-based" methods due to their explicit use of the MRI forward model. These architectures typically alternate between a module that recovers missing k-space entries by removing aliasing in the image domain and a module that ensures consistency with the k-space data. This implies that (11), or possibly a "soft" version of it where the data is not forced to be exactly consistent, may already be implemented as part of the network architecture. In the experimental evaluation of the methods in this article we used the Variational Network (VarNet) [12], [51], which is one such architecture where (11) is not necessary. However, in preliminary studies not presented in this article we found that a U-net [52], which does not already employ data consistency, benefited considerably from (11).

### D. Relationship to SSDU

This section shows that SSDU [33] with an $\ell_2$ loss is a version of Noisier2Noise with a particular rank-deficient loss weighting matrix $W$.

To see the connection between SSDU and Noisier2Noise, it is instructive to see the relationship between Noisier2Noise's $\Lambda_t$ and SSDU's disjoint subsets $A_t$ and $B_t$. Disjoint subsets of $\Omega_t$ can be formed in terms of $\Omega_t$ and $\Lambda_t$ by setting $A_t = \Omega_t \backslash \Lambda_t$ and $B_t = \Omega_t \cap \Lambda_t$. The distribution of $A_t$ and $B_t$ are defined by the distributions of $\Omega_t$ and $\Lambda_t$ and always satisfy $A_t \cup B_t = \Omega_t$ and $A_t \cap B_t = \emptyset$ as required. In terms of sampling masks, this is written as $M_{A_t} = (\mathbb{1} - M_{\Lambda_t}) M_{\Omega_t}$ and $M_{B_t} = M_{\Lambda_t} M_{\Omega_t}$. Therefore, SSDU's loss (2) with a squared $\ell_2$ norm is

$$\sum_t \|M_{A_t} f_\theta(M_{B_t} y_t) - M_{A_t} y_t\|_2^2 = \sum_t \|(\mathbb{1} - M_{\Lambda_t}) \cdot M_{\Omega_t}(f_\theta(\widetilde{y}_t) - y_t)\|_2^2,$$

so is exactly Noisier2Noise with $W = (\mathbb{1} - M_{\Lambda_t}) M_{\Omega_t}$. In other words, while Noiser2Noise's loss is computed over all k-space, SSDU's loss is computed only on indices that are in $\Omega_t$ but not in $\Lambda_t$.

SSDU's weighting ensures that any indices not sampled in $Y$ are ignored in the loss. One might think that the correct choice for this goal would be $W = M_{\Omega_t}$. However, if a data consistent network is employed, as in (11), the contribution to the loss from indices in both $\Omega_t$ and $\Lambda_t$ would be zero because they are consistent by construction. Therefore the loss for $W = M_{\Omega_t}$ and $W = (\mathbb{1} - M_{\Lambda_t}) M_{\Omega_t}$ would be identical. A similar idea was presented for fully supervised learning in [53], where a mask is applied to the training data multiple times.

### E. Proof of SSDU

This section shows that SSDU's loss weighting causes the correction $(\mathbb{1} - K)^{-1}$ at inference to no longer be necessary. When the weighting matrix $W$ is the random variable $(\mathbb{1} - M_\Lambda) M_\Omega$, the network parameters are trained to seek a minimum of

$$\theta^* = \arg\min_\theta \mathbb{E}[\|(\mathbb{1} - M_\Lambda) M_\Omega (f_\theta(\widetilde{Y}) - Y)\|_2^2 | \widetilde{Y}]. \tag{12}$$

Unlike Noisier2Noise, $W = (\mathbb{1} - M_\Lambda) M_\Omega$ is not full-rank, so $f_{\theta^*}(\widetilde{Y}) \neq \mathbb{E}[Y|\widetilde{Y}]$. The usual theoretical goal for self-supervised methods is to prove that the network is correct in expectation [38], [39], [40], [41], [42], [43], [44], as in Claim 1 for variable density Noisier2Noise. In the following we state, to our knowledge, the first similar result for SSDU.

*Claim 2:* A network with parameters that minimizes (12) satisfies

$$(\mathbb{1} - K)(\mathbb{1} - M_\Lambda M_\Omega)(f_{\theta^*}(\widetilde{Y}) - \mathbb{E}[Y_0|\widetilde{Y}]) = 0. \tag{13}$$

*Proof:* See Section B of the Appendix. □

If $\mathbb{1} - K$ is invertible, which holds when $p_j > 0$ and $\widetilde{p}_j < 1$ for all $j$,

$$(\mathbb{1} - M_\Lambda M_\Omega) f_{\theta^*}(\widetilde{Y}) = (\mathbb{1} - M_\Lambda M_\Omega) \mathbb{E}[Y_0|\widetilde{Y}].$$

Therefore, in general, $f_{\theta^*}(\widetilde{Y})$ is correct in expectation, but only in regions of k-space that are not sampled in $\widetilde{Y}$. This contrasts

with the variable density Noisier2Noise method presented in Section II-A, which is correct in expectation for all k-space indices. However, as described in the following, this apparent shortcoming can easily be circumvented by using all available data at inference.

Similarly to Noisier2Noise's (9) and (10), we consider two options for the k-space estimate at inference, both of which use all available data. Firstly, similarly to (9), the data consistent estimate

$$\hat{Y}^{dc} = (\mathbb{1} - M_\Omega) f_{\theta^*}(\widetilde{Y}) + Y \tag{14}$$

can be used, which is correct in expectation everywhere in k-space for any network architecture. Alternatively, the SSDU paper [33] suggests using

$$\hat{Y} = f_{\theta^*}(Y) \tag{15}$$

and a physics-guided network architecture. Like (10) for Noisier2Noise, the network input for (15) is singly sub-sampled, so Claim 2 does not apply and the estimate is not guaranteed to be correct in expectation. Nonetheless, it has the advantage over (14) that it uses all available data in the input to the network. As in [33], we have found that (15) performs well in practice when the network architecture includes a data consistency module: see Section IV.

We emphasize that unlike Noisier2Noise, SSDU does not require the correction term $(\mathbb{1} - K)^{-1}$ at inference. This implies that SSDU is less sensitive to inaccuracies in $f_{\theta^*}(\widetilde{Y})$, and we have found that SSDU outperforms Noisier2Noise in general: see Section IV.

### F. K-Weighted SSDU

Since we train on a finite number of instances of the random variables $Y$, $\widetilde{Y}$, $\Omega$ and $\Lambda$, the network parameters we obtain in practice, which we denote $\hat{\theta}$, are an approximation of the ideal $\theta^*$ from (12). In this case, the right-hand-side of (13) is not exactly zero. Rather,

$$(\mathbb{1} - K)(\mathbb{1} - M_\Lambda M_\Omega)(f_{\hat{\theta}}(\widetilde{Y}) - \mathbb{E}[Y_0|\widetilde{Y}]) = \mathcal{E}, \tag{16}$$

where $\mathcal{E}$ is a vector random variable. The vector $\mathcal{E}$ characterizes the difference between a true expectation and the network's estimate of it, which is non-zero for finite data. In other words, $\mathcal{E}$ is a statistical error due to finite sampling. The difference between the trained network's output and the expectation of interest, $\mathbb{E}[Y_0|\widetilde{Y}]$, is $(\mathbb{1} - K)^{-1}\mathcal{E}$. This implies that the network is more sensitive to errors in k-space locations where $(\mathbb{1} - K)^{-1}$ is large.

To compensate for this, we propose minimizing the following weighted version of SSDU's loss as an alternative to (12):

$$\arg\min_\theta \mathbb{E}[\|(\mathbb{1} - K)^{-\frac{1}{2}}(\mathbb{1} - M_\Lambda)M_\Omega(f_\theta(\widetilde{Y}) - Y)\|_2^2|\widetilde{Y}].$$

Introducing $(\mathbb{1} - K)^{-\frac{1}{2}}$ in the loss cancels the $\mathbb{1} - K$ in (16), so mitigates the error amplification caused by $\theta^*$ approximation. We find that this version of SSDU, which we refer to as "K-weighted SSDU" throughout the remainder of this article, substantially improves the image restoration quality and robustness to training hyperparameters: see Section IV. We chose the power $(\mathbb{1} - K)^{-\frac{1}{2}}$ because it exactly cancels the $\mathbb{1} - K$ on the left-hand-side of (16) when the squared $\ell_2$ loss is used; we also tried power $(\mathbb{1} - K)^{-1}$ and found that, as expected, it did not perform as well in practice.

### G. Understanding the Need for Correction

This section intuitively explains why Noisier2Noise requires correction at inference but SSDU does not. We can write the weighted loss as

$$\begin{aligned}\|W(f_\theta(\widetilde{Y}) - Y)\|_2^2 = \|W[&M_\Omega M_\Lambda + (\mathbb{1} - M_\Lambda)M_\Omega \\ &+ (\mathbb{1} - M_\Omega)](f_\theta(\widetilde{Y}) - Y)\|_2^2,\end{aligned}$$

where we have used that the term is square brackets equals the identity matrix. When $f_\theta(\widetilde{Y})$ is consistent with $\widetilde{Y}$, such as in (11), $M_\Omega M_\Lambda(f_\theta(\widetilde{Y}) - Y) = 0$. Therefore

$$\begin{aligned}\|W(f_\theta(\widetilde{Y}) - Y)\|_2^2 = &\|W(\mathbb{1} - M_\Lambda)M_\Omega(f_\theta(\widetilde{Y}) - Y)\|_2^2 \\ &+ \|W(\mathbb{1} - M_\Omega)f_\theta(\widetilde{Y})\|_2^2,\end{aligned} \tag{17}$$

where we have used $(\mathbb{1} - M_\Omega)Y = 0$. In (17) is SSDU's loss function (12) plus a contribution from all $j \in \Omega_i^c$.

Intuitively, the second term on the right-hand-side of (17) causes the proposed method to underestimate regions of k-space with index $j \in \Omega_j^c$. This underestimation is compensated for with $(\mathbb{1} - K)^{-1}$ at inference. For SSDU, where $W = (\mathbb{1} - M_\Lambda)M_\Omega$, the second term on the right-hand-side of (17) is zero, k-space is not underestimated anywhere, and there is no need for a correction term at inference.

## III. Experimental Method

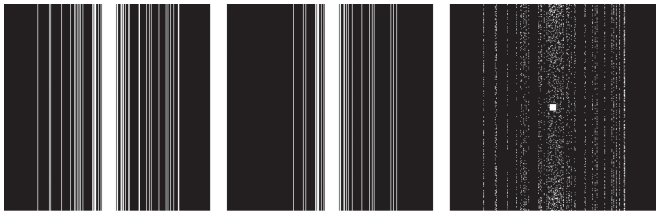
### A. Description of Data

We used the multi-coil brain and knee data from the fastMRI dataset [25], which is comprised of multi-channel raw k-space MRI data. The reference fastMRI test set data is magnitude images only, without fully sampled k-space data. Since we also require phase, we discarded the data allocated for testing and generated our own partition into training, validation and test sets. For the brain data, we only used data that was acquired on 16 coils, and used training, validation and test set sizes of 127, 19 and 14 volumes (2020, 302, and 224 slices) respectively. For the knee data, the training, validation and test sets consisted of 166, 19 and 14 volumes (5977, 665, and 493 slices) respectively. We set the network output to be zero in regions of k-space where the reference data had zero padding.

### B. Network Architecture

For $f_\theta$, we used the variant of the VarNet [12] that estimates coil sensitivities on-the-fly [51], which performs competitively on the fastMRI leaderboard and is available as part of the fastMRI package.[1] After a coil sensitivity estimation module, VarNet uses multiple repetitions of a module based on gradient

[1] [Online]. Available: https://github.com/facebookresearch/fastMRI



(a) An example $M_{\Omega_t}$ (b) $M_{\Lambda_t} M_{\Omega_t}$ for 1D partitioned SSDU and Noisier2Noise. (c) $M_{\Lambda_t} M_{\Omega_t}$ for 2D partitioned SSDU

Fig. 2. Example of the singly sub-sampled mask $M_{\Omega_t}$, and doubly sub-sampled $M_{\Lambda_t} M_{\Omega_t}$ with two $M_\Lambda$ distribution types. Here, the acceleration factor of the first mask is $R_\Omega = 4$ and the second is $R_\Lambda = 2$.

descent, which is comprised of a data consistency term in k-space and a prior based on a U-net [52] that acts in the image domain after an inverse Fourier transform and coil combination. The output of the neural network was in k-space. We used 6 repetitions of the main module, so that our model had around $1.5 \times 10^7$ parameters. Note that in [25], the Structural Similarity Index (SSIM) [54] was used as the loss, while in this article we use an $\ell_2$ loss.

The only additional operations SSDU and Noisier2Noise require compared to fully-supervised training are simple entry-wise masks, so all methods had similar memory requirements and training time. We trained for 50 epochs, which took around 17 hours on a GTX 1080 Ti GPU with 11 GB of RAM for the brain data. For all methods we used the Adam optimizer [55] with a fixed learning rate of $10^{-3}$. Our PyTorch implementation is publicly available on GitHub.[2]

### C. Distribution of Masks

So that the distribution of the sampling masks were known exactly, we generated our own masks rather than using those suggested in fastMRI. Unless stated otherwise, the distribution of the first mask $M_\Omega$ was 1D column-wise. We fully sampled the central 10 columns and sampled the remainder with polynomial variable density. We used polynomial order 8, and scaled the probability density $P$ so that it matched a desired acceleration factor. We ran each method with $R_\Omega \in \{4, 8\}$, where $R_\Omega = N/\sum_j p_j$ is the expected acceleration factor. An example at $R_\Omega = 4$ is shown in Fig. 2(a).

In [41], it is suggested that the distribution of Noisier2Noise's second random variable is the same as the first, but not necessarily with the same distribution parameters. Therefore, for Noisier2Noise's second mask $M_\Lambda$, we used the same type of distribution as $M_\Omega$ with a different variable density. An example with $R_\Omega = 4$ and $R_\Lambda = N/\sum_j \widetilde{p}_j = 2$ is shown in Fig. 2(b). Concretely, we define two masks as having the same 'type' of distribution when the conditional dependence of the sampling set indices is the same. Let $p_{j|k} = \mathbb{P}[j \in \Omega | k \in \Omega]$. If $p_{j|k} = p_j$ for all $j$ and $k$, the entries are independent and the mask is the type '2D Bernoulli'. If $p_{j|k} = 1$ when $j$ and $k$ are in the same k-space column and $p_{j|k} = p_j$ otherwise, the mask is the type '1D column-wise'. The experiments in this article focus on these two types of masks; other types are discussed in Section V. We emphasize that constraining a mask to a type does not constrain the $p_j$s, which define the variable sampling density.

To ensure that $\widetilde{p}_j < 1$ everywhere, we set $\widetilde{p}_j = 1 - \epsilon$ in the central 10 columns of k-space, where $\epsilon$ is a small real constant. The network architecture ensures that the central region is consistent with the input, so $\epsilon$ can be small without penalty. We used $\epsilon = 10^{-3}$.

In order to be a realistic simulation of prospectively sub-sampled data, the sampling set $\Omega_t$ must be fixed for all epochs. However, $\Lambda_t$ need not be. Therefore, we re-generated $M_{\Lambda_t}$ from the distribution of $M_\Lambda$ once per epoch. Since the network sees more samples from the distribution of $M_\Lambda$, the loss function is closer to (5), so $f_{\hat{\theta}}$ is expected to be a more accurate approximation of $\mathbb{E}[Y|\widetilde{Y}]$. This has similarities with training data augmentation, as each slice is used to generate several inputs to the network [56].

### D. Comparative Methods

We trained Noisier2Noise using different weightings of the $\ell_2$ loss stated in (8). For each self-supervised method, we considered two possible estimates at inference: one with the doubly sub-sampled $\widetilde{y}_s$ as the network input and the other with the singly sub-sampled $y_s$. The methods and their two estimates at inference are summarized in Table I.

We trained with $W = \mathbb{1}$, referred to as "Unweighted Noisier2Noise". By Claim 1, Unweighted Noisier2Noise requires a $(\mathbb{1} - K)^{-1}$ correction at inference: see Table I. We have found that the need for correction substantially reduces the image quality compared to SSDU, so do not recommend using Unweighted Noisier2Noise in practice. Nonetheless, we include some Unweighted Noisier2Noise results to illustrate the value of SSDU's loss weighting.

We also trained Noisier2Noise with $W = (\mathbb{1} - M_\Lambda) M_\Omega$ which, based on the relationship described in Section II-D, we refer to as "SSDU", despite some differences between our implementation and [33]. In [33], a mixture of an $\ell_1$ and $\ell_2$ loss was used, whereas here, so that it can be directly compared with Unweighted Noisier2Noise, we used an $\ell_2$ loss. We also used a different $M_\Omega$ distribution, dataset and network architecture to [33].

SSDU [33] was originally applied to an architecture that requires pre-computed sensitivity maps. It was suggested that $M_{B_t}$ has a fully sampled $4 \times 4$ central region and 2D Gaussian variable density otherwise, so that high frequencies are sampled with higher probability. For the architecture considered in this article, which has a coil sensitivity estimation module, we found that increasing the size of the fully sampled central region considerably improved the method's performance. Since $M_\Omega$ has 10 fully sampled central columns, we increased the size of the central region of $M_\Lambda$ to $10 \times 10$.

As the probability of sampling each location in k-space is independent, the sampling set partition proposed in [33] is

[2][Online]. Available: https://github.com/charlesmillard/Noisier2Noise_for_recon

TABLE I
THE SELF-SUPERVISED METHODS EVALUATED IN THIS PAPER

| NAME | LOSS WEIGHTING $W$ | $M_\Lambda$ DISTRIBUTION | ESTIMATE WITH $\widetilde{y}_s$ INPUT | ESTIMATE WITH $y_s$ INPUT |
|---|---|---|---|---|
| Unweighted Noisier2Noise | $\mathbb{1}$ | 1D column-wise | $(\mathbb{1}-M_{\Omega_s})(\mathbb{1}-K)^{-1}f_{\hat{\theta}}(\widetilde{y}_s)+y_s$ | $(\mathbb{1}-K)^{-1}(f_{\hat{\theta}}(y_s)-Ky_s)$ |
| 2D partitioned SSDU | $(\mathbb{1}-M_{\Lambda_t})M_{\Omega_t}$ | 2D Bernoulli | $(\mathbb{1}-M_{\Omega_s})f_{\hat{\theta}}(\widetilde{y}_s)+y_s$ | $f_{\hat{\theta}}(y_s)$ |
| 1D partitioned SSDU | $(\mathbb{1}-M_{\Lambda_t})M_{\Omega_t}$ | 1D column-wise | $(\mathbb{1}-M_{\Omega_s})f_{\hat{\theta}}(\widetilde{y}_s)+y_s$ | $f_{\hat{\theta}}(y_s)$ |
| K-weighted 1D partitioned SSDU | $(\mathbb{1}-K)^{-\frac{1}{2}}(\mathbb{1}-M_{\Lambda_t})M_{\Omega_t}$ | 1D column-wise | $(\mathbb{1}-M_{\Omega_s})f_{\hat{\theta}}(\widetilde{y}_s)+y_s$ | $f_{\hat{\theta}}(y_s)$ |

Here, and throughout this paper, the subscripts $t$ and $s$ index the training and test sets respectively. Examples of $M_{\Lambda_t}M_{\Omega_t}$ for 2D bernoulli and 1D column-wise $M_{\Lambda_t}$ are shown in fig. 2.

equivalent to a 2D variable density Bernoulli $M_\Lambda$ distribution. To estimate their variable density distribution $\widetilde{P}$ we ran the SSDU authors' set partitioning code[3] 1000 times on a fully sampled mask and averaged the result. We trained SSDU using a distribution of $M_\Lambda$ of this type, referred to as "2D partitioned SSDU", illustrated in Fig. 2(c). We also trained SSDU using the same distribution type of $M_\Lambda$ as $M_\Omega$, as in Fig. 2(b). We refer to this method as "1D partitioned SSDU", or "K-weighted 1D partitioned SSDU" when a $(\mathbb{1}-K)^{-\frac{1}{2}}$ weighting is used in the loss as described in Section II-F. Like Unweighted Noisier2Noise, $M_{\Lambda_t}$ was re-generated once per epoch [56]. We emphasize that although 2D partitioned SSDU has a similar $M_\Lambda$ distribution as in [33], the distribution of $M_\Omega$ here is random variable density columns, not equidistant columns as in [33]. Therefore, 2D partitioned SSDU is not necessarily expected to perform as well as SSDU in [33].

As a best-case target, we also trained using a fully supervised method with an (unweighted) $\ell_2$ loss. All deep learning methods had the same network architecture and training hyperparameters, as described in III-B.

Finally, as a comparative method that does not use deep learning, we ran a compressed sensing algorithm with a sparse model on wavelet coefficients, which we implemented via the Berkeley Advanced Reconstruction Toolbox (BART) [57]. We used BART's default settings with fourth-order Daubechies wavelets and a sparse weighting of $\lambda = 2\times10^{-3}$.

### E. Quality Metrics

To evaluate the reconstruction quality, we computed the Normalized Mean Squared Error (NMSE) in k-space on the test set: $\|\hat{y}_s - y_{0,s}\|_2^2/\|y_{0,s}\|_2^2$. We also computed the image-domain root-sum-of-squares (RSS), $\hat{x}_s = (\sum_c |F^H y_{s,c}|^2)^{1/2}$ where $y_{s,c}$ is the k-space entries on coil $c$ and $F$ is the discrete Fourier transform, cropped the RSS estimate to a central $320\times320$ region and computed the SSIM, as suggested in fastMRI [25].

## IV. RESULTS

For brevity, the results presented here focus on $R_\Omega = 8$. Similar results for the brain data at $R_\Omega = 4$ are shown in the supplementary material: see Figs. S1– S4.

For the brain data, we evaluated the dependence of the methods' performance on the distribution of $M_\Lambda$ by varying the parameters so that the sub-sampling factor $R_\Lambda$ changed. We trained with $R_\Lambda \in \{1.2, 1.6, 2, 4, 6\}$, except for 2D partitioned SSDU, which we found needed finer tuning and a smaller $R_\Lambda$ for the best performance, so we trained with $R_\Lambda \in \{1.1, 1.2, \ldots, 2, 3, 4, 6\}$.

[3][Online]. Available: https://github.com/byaman14/SSDU

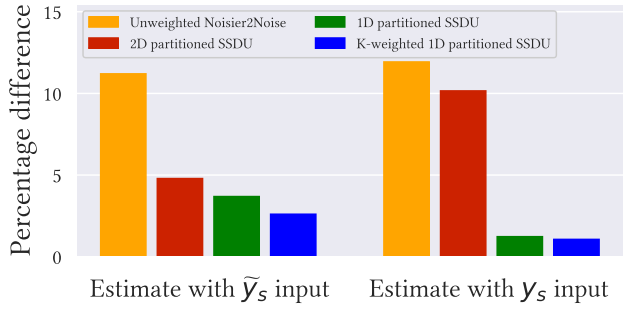



Fig. 3. Mean test set NMSE percentage difference between fully supervised and each methods at $R_\Omega = 8$ and a 1D distributed $M_\Omega$, where $R_\Lambda$ has been tuned to minimize the test set NMSE. Fig. S1 shows a similar plot for $R_\Omega = 4$.

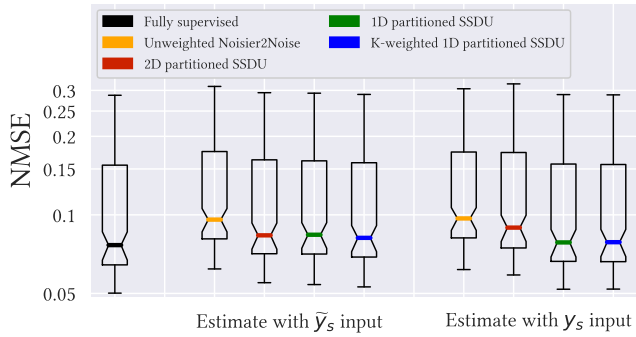



Fig. 4. NMSE for all methods at $R_\Omega = 8$ and a 1D distributed $M_\Omega$, where $R_\Lambda$ has been tuned to minimize the test set NMSE. Fig. S2 shows a similar plot for $R_\Omega = 4$ and the exact numerical values are in Table S1.

### A. Performance With Tuned $R_\Lambda$

This section focuses on the case where $R_\Lambda$ has been tuned to minimize the ground truth test set NMSE. Figs. 3 and S1 show bar charts of the percentage difference between fully supervised training and each method: $(\mu - \mu_{full})/\mu_{full}$ where $\mu$ and $\mu_{full}$ are the mean NMSE of interest and mean NMSE of fully supervised training respectively. The best performance was for K-weighted 1D partitioned SSDU with a $y_s$ input; its mean NMSE was only 1.1% and 0.8% larger than fully supervised for $R_\Omega = 8, 4$ respectively. Figs. 4 and S2 show box plots of the NMSE of each method for $R_\Omega = 8$ and $R_\Omega = 4$ respectively:

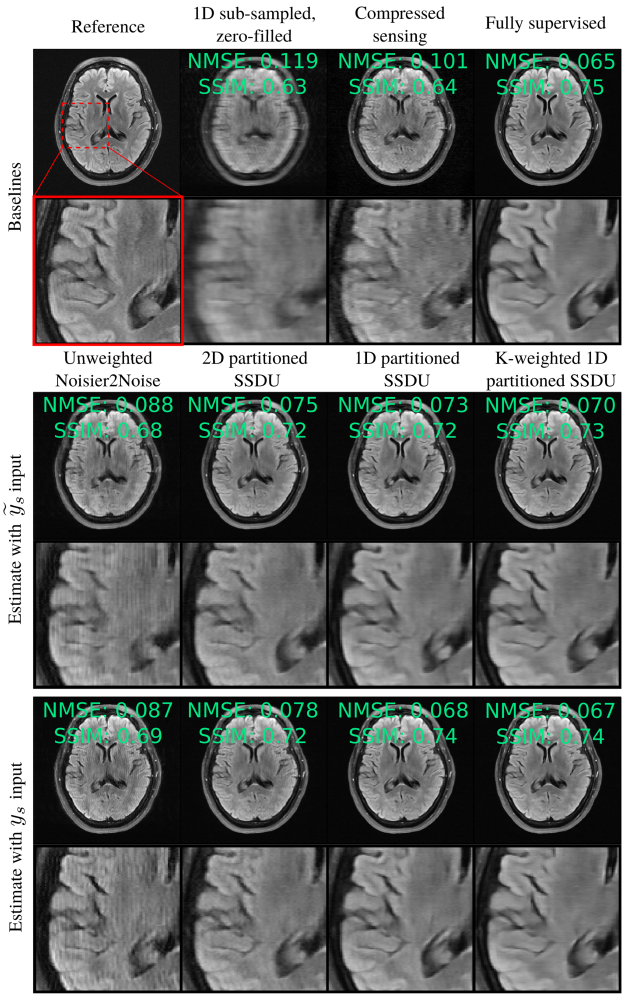



Fig. 5. Reconstruction example with a 1D sub-sampled $M_\Omega$ and $R_\Omega = 8$, with a $R_\Lambda$ tuned to minimize the test set NMSE. A similar figure for $R_\Omega = 4$ is in the supplementary material, Fig. S3.

see Table S1 of the supplementary material for the numerical values.

To evaluate whether the proposed changes to SSDU were statistically significant, we performed a one-sided Wilcoxon signed-rank test with $p$-value 0.01 on the test set NMSEs. For both the $y_s$ and $\widetilde{y}_s$ inputs, we found that there was a significant statistical difference between 2D and 1D partitioned SSDU. We also found that the difference between 1D partitioned SSDU and K-weighted 1D partitioned SSDU was statistically significant.

Figs. 5 and S3 show RSS estimates from the test set at $R_\Omega = 8$ and $R_\Omega = 4$ respectively. Qualitatively, K-weighted 1D partitioned SSDU performs the most similarly to fully supervised training. Although 2D partitioned SSDU has a competitive quantitative score for the estimate with $\widetilde{y}_s$ input, it exhibits some streaking artifacts.

Unweighted Noisier2Noise's performance was substantially worse than SSDU. Therefore we compare SSDU and its modifications only in the remainder of this article.

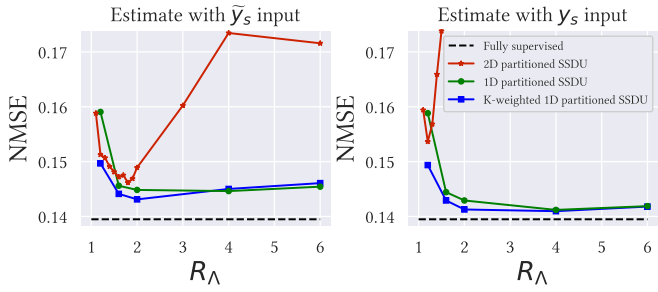



Fig. 6. Dependence of the test set NMSE on the acceleration factor of the second mask $M_\Lambda$, denoted as $R_\Lambda$, at $R_\Omega = 8$ for both outputs. 1D partitioned SSDU is far more robust to the tuning of $R_\Lambda$ than 2D partitioned SSDU. Fully supervised learning does not use a second mask $M_\Lambda$, so has the same performance for all $R_\Lambda$. A similar figure for $R_\Omega = 4$ is in the supplementary material, Fig. S3.

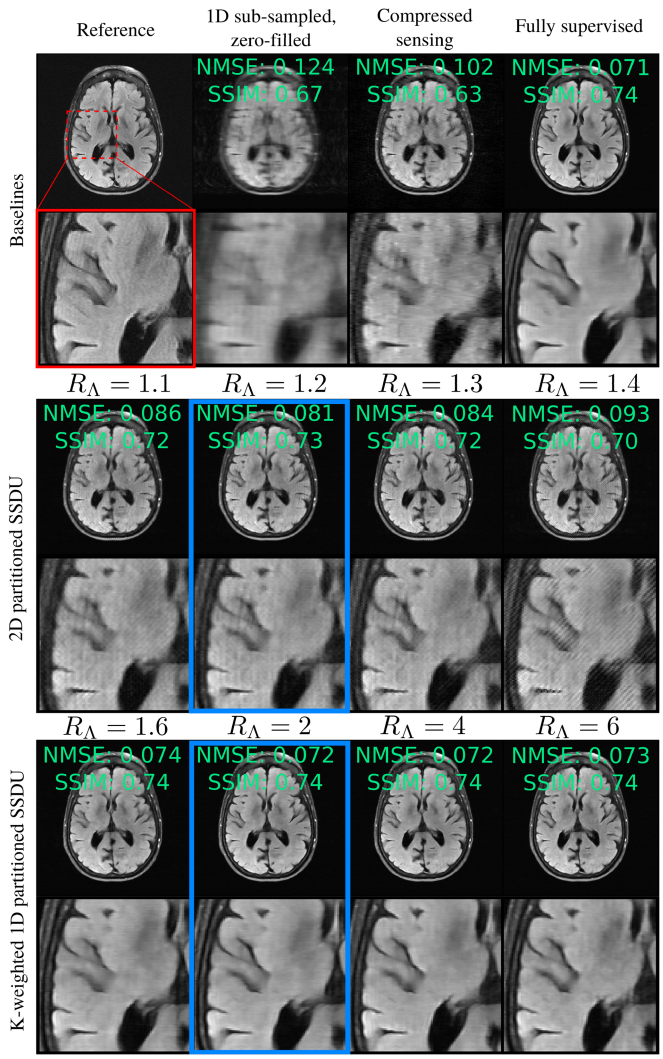



Fig. 7. Robustness to $R_\Lambda$, where the blue box highlights the case where $R_\Lambda$ is tuned. K-weighted 1D partitioned SSDU is very robust to $R_\Lambda$, with very similar restoration quality for all $R_\Lambda$ between 1.6 and 6. 2D partitioned SSDU is far more sensitive, with substantial degradation in image quality for mistunings as small as 0.1. Here, we show the estimate with $y_s$ input only.

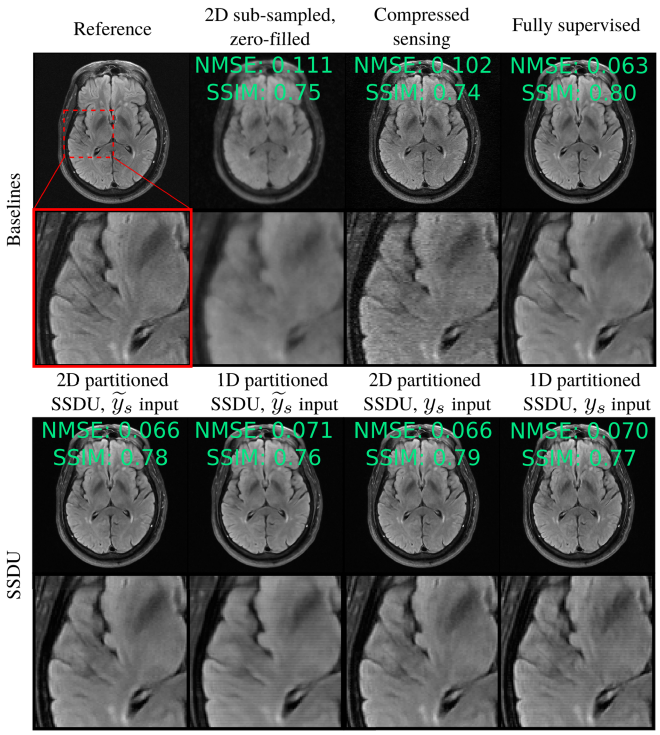



Fig. 8. Reconstruction example from the brain fastMRI dataset with a 2D Bernoulli distributed $M_\Omega$ and $R_\Omega = 8$. Compared to Fig. 5, the comparative performance of the SSDU algorithms are switched: here, 2D partitioned SSDU performs similarly to fully supervised training, while 1D partitioned SSDU suffers from streaking artifacts.

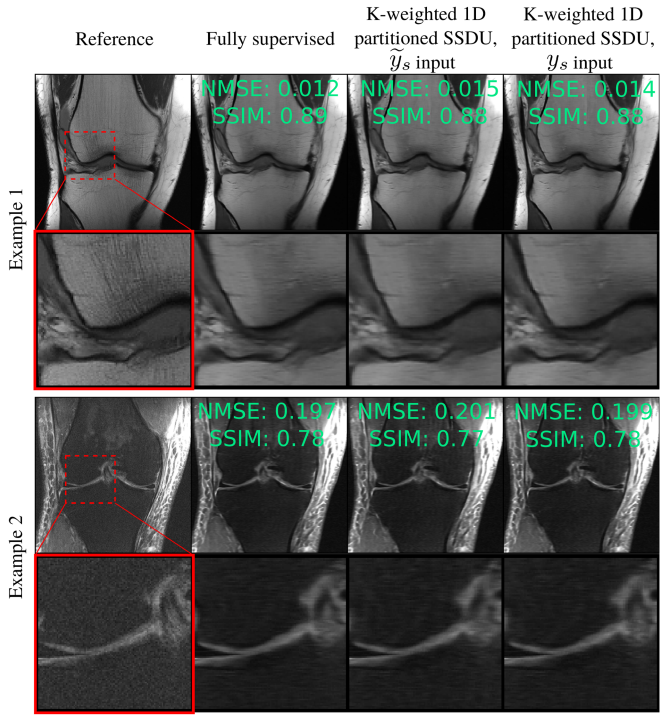



Fig. 9. Two reconstruction examples of K-weighted 1D partitioned SSDU from the knee fastMRI dataset, where $M_\Omega$ is 1D. As in Fig. 5, K-weighted 1D partitioned SSDU's restoration quality is very similar to fully supervised training.

### B. Robustness to $R_\Lambda$

For actual, prospectively sampled data, it would not be possible to tune $R_\Lambda$ on the ground truth test set NMSE. The practicality of SSDU therefore depends greatly on the robustness to $R_\Lambda$. Figs. 6 and S4 show the dependence of the mean test set NMSE on $R_\Lambda$ for $R_\Omega = 8$ and $R_\Omega = 4$ respectively. K-weighted 1D partitioned SSDU was the most robust to the tuning of $R_\Lambda$. 2D partitioned SSDU was the least robust, especially for the estimate with $y_s$ input. This is visualized in Fig. 7, which shows reconstruction examples for a number of $R_\Lambda$s. K-weighted 1D partitioned SSDU performs very similarly for all $R_\Lambda$s between 1.6 and 6, while 2D partitioned SSDU's restoration quality significantly degrades qualitatively and quantitatively for mistunings as small as 0.1.

### C. Performance on 2D Sampled Brain Data

To further evaluate the role of the partitioning distribution, we also ran 1D and 2D partitioned SSDU on the brain data with a 2D Bernoulli sampled $M_\Omega$. In this case, the type matching of the second mask to $M_\Omega$ is switched: 2D partitioned SSDU's second mask has the same type of distribution as the first, while 1D partitioned SSDU has a different type. For $M_\Omega$, we used a fully sampled $10 \times 10$ central region and a polynomial variable density that samples low frequencies with higher probability otherwise. We used $R_\Lambda = 1.2$ and $R_\Lambda = 4$ for 2D and 1D partitioned SSDU respectively. All other hyperparameters and network specifics were unchanged.

In this case, the best performance was 2D partitioned SSDU, which performed very similarly to fully supervised training: see Fig. 8. The $\widetilde{y}_s$ input had a mean test set NMSE of 0.141 and 0.144 for 2D and 1D partitioned SSDU respectively, and the $y_s$ input had 0.141 and 0.145, compared with 0.139 for fully supervised training. Although not shown in Fig. 8 for brevity, we also trained 2D partitioned SSDU with a $(\mathbb{1} - K)^{-\frac{1}{2}}$ loss weighting. As for 1D partitioned SSDU in Section IV-A, we found that this reduced the mean NMSE further to 0.140 for both the $y_s$ and $\widetilde{y}_s$ input.

### D. Performance on 1D Sampled Knee Data

We also trained K-weighted 1D partitioned SSDU on the fastMRI knee data with the same network architecture, training hyperparameters, and a 1D distributed $M_\Omega$. The sub-sampling factor of the first and second masks were $R_\Omega = 8$ and $R_\Lambda = 2$ respectively. The mean test set NMSE was 0.233 and 0.231 for the estimates with $\widetilde{y}_s$ and $y_s$ inputs respectively, compared with 0.230 for fully supervised training. Fig. 9 shows two example reconstructions from the test set, demonstrating competitive performance with fully supervised training qualitatively.

## V. Discussion

Due to its need for correction at inference, Unweighted Noisier2Noise had consistently the worst score. We therefore do

*not* recommend using Unweighted Noisier2Noise in practice. Rather, we suggest using a variant of SSDU, which has a loss weighting that removes the need for such a correction.

The hierarchy of 1D and 2D partitioned SSDU depends on the distribution of $M_\Omega$. In particular, the best performance was when they are both 1D or both 2D. It is conventional wisdom that better reconstruction quality is possible when k-space is randomly sub-sampled in both spatial dimensions (see, for instance, [58]). This is because the image-domain aliasing is incoherent in both dimensions, so is easier to remove. The superior performance of 1D partitioned SSDU compared with 2D partitioned SSDU when $M_\Omega$ is 1D shows that it is *not* necessarily true that the sampling set partition should also ideally be two-dimensional. Rather, better performance is possible when the distribution of $M_\Omega$ and $M_\Lambda M_\Omega$ are of the same type.

To see why, consider the nature of the aliasing caused by sub-sampling and further sub-sampling k-space, focusing on the example of a random 1D column sampled $M_\Omega$. Such sampling causes the image-domain aliasing to be horizontally incoherent and vertically coherent. With a 1D column-wise $\Lambda_t$, further horizontal aliasing is introduced. Since the network cannot distinguish between the horizontal aliasing caused by $\Omega_t$ or $\Lambda_t$, the loss is minimized when the aliasing due to *both* is removed. On the other hand, a 2D $\Lambda_t$ introduces some aliasing that is orthogonal to the original aliasing, which is distinguishable in principle. In this case, the loss is minimized when the network removes the aliasing caused by $\Lambda_s$, but not necessarily the original aliasing caused by $\Omega_s$. This is visible in Figs. 5 and 8, where SSDU fails to completely remove artifacts caused by $M_\Omega$ when $M_\Lambda$ does not have the same type of distribution.

This implies that, in general, better performance is possible when the distribution of the aliasing of $\widetilde{y}_t$ and $y_t$ are of the same type. For both the independent 1D column sampling and 2D Bernoulli sampling considered here, this can be achieved by choosing a $M_\Lambda$ with the same type of distribution as $M_\Omega$. Recently, in [59], this was also observed empirically for SSDU with random spoke sampling. However, such a procedure does not always achieve this goal. For instance, while the SSDU paper [33] considers a fully sampled central region and equidistant column sampling, recovery of images with regular undersampling is not currently considered in the proposed framework. In this case, a $\Lambda_t$ of the same type would not give a $\widetilde{y}_t$ with the same aliasing type as $y_t$. The 2D Gaussian variable density partition employed in this article was originally constructed to handle such sampling patterns, and was found to perform very well in this context. Future work includes establishing the correct sampling set partitions for $M_\Omega$ distributions not in [33] or covered by the approach suggested here.

We found that K-weighted SSDU further improved the image quality and robustness to $R_\Lambda$. Consider the $j$th entry of the (squared) weighting $(\mathbb{1} - K)^{-1}$ in terms of sampling probabilities:

$$(1 - k_j)^{-1} = \frac{1 - \widetilde{p}_j p_j}{p_j(1 - \widetilde{p}_j)} = \frac{\mathbb{P}(j \notin \Lambda \cap \Omega)}{\mathbb{P}(j \in \Omega \setminus \Lambda)}.$$

This leads to the following intuitive interpretation of the proposed loss weighting as compensation for the variable density of $\Omega$ and $\Lambda$. A smaller denominator $\mathbb{P}(j \in \Omega \setminus \Lambda)$ implies that the $j$th location occurs less frequently in the loss, which is compensated for by an increased weighting. A smaller numerator $\mathbb{P}(j \notin \Lambda \cap \Omega)$ implies that the $j$th location is estimated by the network less frequently, so has a decreased weighting.

The benefit of the $(\mathbb{1} - K)^{-1}$ weighting highlights and addresses a general challenge of self-supervised learning with variable density sampling: regions of k-space sampled with lower probability are underrepresented in the loss. This issue has been noted in other works. For instance, for variable density reconstruction with Noise2Noise, [60] suggests weighting the loss function by the sampling density. An alternative approach was suggested in [61], which suggests passing the training target through the network before it is employed in the loss function. We note that if the sampling and partitioning had uniform density, such as in [56], $K$ would also be uniform, so the proposed weighting would not be required. This may explain in part the empirical performance observed in [56].

When $M_\Omega$ was 1D, with the exception of 2D partitioned SSDU, Fig. 6 shows that the estimate with $y_s$ input performed similarly or better than with $\widetilde{y}_s$ input when $R_\Lambda$ is tuned. This indicates that, for these methods, the advantage of using all the data in the input to the network outweighs the disadvantage that the input data has a different sampling distribution to the training data so is not guaranteed by Claim 1 or 2 to be correct in expectation. Heuristically, when $M_\Omega$ and $M_\Lambda M_\Omega$ are both variable density column-wise sampled, a network trained on doubly sub-sampled data is likely to also be able to handle singly sub-sampled data. However, for 2D partitioned SSDU, $M_\Lambda M_\Omega$ is no longer column-wise, see Fig. 2(c). Accordingly, 2D partitioned SSDU was the only method that had a higher NMSE for the $y_s$ input compared to the $\widetilde{y}_s$ input.

The best $R_\Lambda$ for 2D partitioned SSDU was lower than competing methods: $R_\Lambda = 1.8$ and $R_\Lambda = 1.2$ for the $y_s$ and $\widetilde{y}_s$ inputs respectively. In [33], the sampling set partition was quantified in terms of the ratio $\rho = |A_t|/|B_t|$, and it was found that $\rho = 0.4$ offered the best performance. Since the $M_\Omega$ distributions are different here, the optimal $\rho$ is not expected to necessarily be the same. For 2D partitioned SSDU $R_\Lambda = 1.8$ and $R_\Lambda = 1.2$ corresponds to $\rho = 0.52$ and $\rho = 0.21$ respectively, while for the other methods's best performance at $R_\Lambda = 4$ corresponded to $\rho = 0.57$. Therefore the $\rho$ were reasonably similar despite the substantial difference in $R_\Lambda$.

Since the network architecture uses $\widetilde{y}_t$ in its coil sensitivity estimation module, not $y_t$, it is plausible that the differences between 1D and 2D partitioning could be due to poorer coil sensitivity estimation rather than an intrinsic property of the partition change. To examine this, we re-trained tuned 1D and 2D partitioned SSDU on the 1D sampled brain data with k-space masked to a central $10 \times 10$ region in the coil sensitivity estimation module. We found that the test set NMSE was within 1% of the usual approach. This verifies that the performance improvement was indeed a consequence of the partition change, not simply a consequence of specifics of the architecture.

Unweighted Noisier2Noise's correction at inference $(\mathbb{1} - K)^{-1}$ is only valid when an $\ell_2$ loss is used; we have found that other loss functions do not perform well in practice. This loss

leads to smoothing artifacts, even for fully supervised training. For SSDU, since there is no correction term, loss functions other than $\ell_2$ are possible. For instance, in [33], a mixture of $\ell_2$ and $\ell_1$ was used. Better visual quality may be achievable when SSDU is implemented with a different loss; we do not suggest using an $\ell_2$ loss in general, it is only required here so that it can be compared directly with Noisier2Noise.

For all self-supervised methods in this work, we re-generated $\Lambda_t$ once per epoch. This has similarities to the multi-mask SSDU approach proposed in [56]. However, in [56], a fixed number $n_\Lambda$ of $\Lambda_t$s were generated for each $\Omega_t$, each of which were treated as an additional member of the training set. Therefore, unlike in this article, each epoch was $n_\Lambda$ times as long. Future work includes establishing whether it is also advantageous to limit the number of unique $\Lambda_t$s per $\Omega_t$ for the approach considered in this article.

All methods in this article were trained without taking measurement noise into account [62], [63]. Recent work by the present authors has shown that the additive and multiplicative versions of Noisier2Noise can be combined to recover higher fidelity images than SSDU in the presence of noise [64].

## VI. Conclusions and Future Work

Based on the observation that SSDU is a version of Noisier2Noise with a particular rank-deficient loss weighting, we proved that SSDU correctly estimates $Y_0$ in expectation. This analysis led to two proposals that we found significantly improved SSDU's performance in practice. Firstly, we propose employing a distribution of $M_\Lambda M_\Omega$ that is the same type as the original mask $M_\Omega$. Secondly, we propose introducing a weighting of $(\mathbb{1} - K)^{-\frac{1}{2}}$ in SSDU's loss. We found that that each of these modifications significantly improved SSDU's test set NMSE and robustness to $R_\Lambda$.

There are a number of other self-supervised learning methods that also use sampling set partitioning [37], [56], [65], some of which are variants of SSDU. For instance, [37], [65], [66] propose training two networks in parallel, one for each sampling subset, with a loss function that includes the difference between the outputs of the two networks. Another recent development is zero-shot SSDU [67], which shows that sampling set partitioning can also be applied to recover images without a training dataset [68]. Future work includes determining whether the theoretical and practical developments of this article can be extended to these methods.

## Appendix

### A. Proof of Variable Density Noisier2Noise

This section of the Appendix proves that when $p_j \neq 0$ and $\widetilde{p}_j \neq 1$ for all $j$,

$$\mathbb{E}[Y_0|\widetilde{Y}] = (\mathbb{1} - K)^{-1}(\mathbb{E}[Y|\widetilde{Y}] - K\widetilde{Y}), \tag{18}$$

where $K = (\mathbb{1} - \widetilde{P}P)^{-1}(\mathbb{1} - P)$ for $P = \mathbb{E}[M_\Omega]$ and $\widetilde{P} = \mathbb{E}[M_\Lambda]$.

*Proof:* This proof is based on Section III-D of Noisier2Noise [41], but with more mathematical detail and generalized to variable density sampling. Following the compressed sensing literature, this article uses $p_j$ to refer to the probability that the $j$th location in k-space is sampled. This differs to [41], which uses $p$ to denote the probability that a pixel is *zeroed*.

We wish to compute $\mathbb{E}[Y_j|\widetilde{Y}_j]$ as a function of $\mathbb{E}[Y_{0,j}|\widetilde{Y}_j]$. To do this, we split $\mathbb{E}[Y_j|\widetilde{Y}_j]$ into two cases, for conditions $\widetilde{Y}_j \neq 0$ or $\widetilde{Y}_j = 0$, and subsequently construct an expression that is consistent with both.

*Case 1.* ($\mathbb{E}[Y_j|\widetilde{Y}_j \neq 0]$)*:* By the measurement model $\widetilde{Y} = M_\Lambda Y = M_\Lambda M_\Omega Y_0$, the singly sub-sampled $Y_j$ must take the same value as $\widetilde{Y}_j$ when $\widetilde{Y}_j \neq 0$. Therefore

$$\mathbb{E}[Y_j|\widetilde{Y}_j \neq 0] = \widetilde{Y}_j. \tag{19}$$

*Case 2.* ($\mathbb{E}[Y_j|\widetilde{Y}_j = 0]$)*:* Using the partition theorem for expectations, we write $\mathbb{E}[Y_j|\widetilde{Y}_j = 0]$ as the weighted sum of $\mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j = 0]$ and $\mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j \neq 0]$:

$$\begin{aligned}\mathbb{E}[Y_j|\widetilde{Y}_j = 0] = \mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j = 0] \cdot k_j \\ + \mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j \neq 0] \cdot (1 - k_j),\end{aligned} \tag{20}$$

where we define $k_j = \mathbb{P}[Y_j = 0|\widetilde{Y}_j = 0]$. Evaluating each of the terms on the right-hand-side of (20) in turn:

- $\mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j = 0]$: Since the random variable $Y_j$ is conditionally zero, its expectation is also zero:
$$\mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j = 0] = 0.$$
- $\mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j \neq 0]$: The measurement model implies that when $Y_j$ is non-zero, and therefore unmasked, it takes the value of $Y_{0,j}$. Therefore its expectation can be written in terms of the expectation of $Y_{0,j}$:
$$\mathbb{E}[Y_j|\widetilde{Y}_j = 0 \cap Y_j \neq 0] = \mathbb{E}[Y_{0,j}|\widetilde{Y}_j = 0]. \tag{21}$$
- $k_j$: By the definition of conditional expectation:
$$k_j = \mathbb{P}[Y_j = 0|\widetilde{Y}_j = 0] = \frac{\mathbb{P}[Y_j = 0 \cap \widetilde{Y}_j = 0]}{\mathbb{P}[\widetilde{Y}_j = 0]}.$$
The numerator is
$$\begin{aligned}\mathbb{P}[Y_j = 0 \cap \widetilde{Y}_j = 0] &= \mathbb{P}[Y_j = 0] \\ &= 1 - p_j,\end{aligned}$$
where $p_j = \mathbb{P}[Y_j \neq 0] = \mathbb{E}[M_{\Omega,jj}]$ is the probability that $j \in \Omega$. By the partition theorem, the denominator is
$$\begin{aligned}\mathbb{P}[\widetilde{Y}_j = 0] &= \mathbb{P}[\widetilde{Y}_j = 0|Y_j = 0]\mathbb{P}[Y_j = 0] \\ &\quad + \mathbb{P}[\widetilde{Y}_j = 0|Y_j \neq 0]\mathbb{P}[Y_j \neq 0] \\ &= 1 \cdot (1 - p_j) + (1 - \widetilde{p}_j)p_j \\ &= 1 - \widetilde{p}_j p_j,\end{aligned}$$
where $\widetilde{p}_j = \mathbb{P}[\widetilde{Y} \neq 0] = \mathbb{E}[M_{\Lambda,jj}]$. Therefore
$$k_j = \mathbb{P}[Y_j = 0|\widetilde{Y}_j = 0] = \frac{1 - p_j}{1 - \widetilde{p}_j p_j}. \tag{22}$$

Substituting the above results into (20) gives

$$\mathbb{E}[Y_j|\widetilde{Y}_j = 0] = \mathbb{E}[Y_{0,j}|\widetilde{Y}_j = 0](1 - k_j), \tag{23}$$

where $k_j$ is defined in (22).

*Combining Cases 1 and 2.* ($\mathbb{E}[Y_j|\widetilde{Y}_j]$)*:* To find $\mathbb{E}[Y_j|\widetilde{Y}_j]$, one must construct an expression that holds for both (19) and (23). Consider the following candidate:

$$\mathbb{E}[Y_j|\widetilde{Y}_j] = (1-k_j)\mathbb{E}[Y_{0,j}|\widetilde{Y}_j] + k_j\widetilde{Y}_j. \tag{24}$$

This expression can be verified as consistent with (19) by setting $\widetilde{Y}_j \neq 0$:

$$\begin{aligned}\mathbb{E}[Y_j|\widetilde{Y}_j \neq 0] &= (1-k_j)\mathbb{E}[Y_{0,j}|\widetilde{Y}_j \neq 0] + k_j\widetilde{Y}_j \\ &= (1-k_j)\widetilde{Y}_j + k_j\widetilde{Y}_j \\ &= \widetilde{Y}_j,\end{aligned}$$

as required. Secondly, setting $\widetilde{Y}_j = 0$ gives

$$\begin{aligned}\mathbb{E}[Y_j|\widetilde{Y}_j = 0] &= (1-k_j)\mathbb{E}[Y_{0,j}|\widetilde{Y}_j = 0] + k_j \cdot 0 \\ &= (1-k_j)\mathbb{E}[Y_{0,j}|\widetilde{Y}_j = 0],\end{aligned}$$

as required by (23). Therefore (24) is consistent with both (19) and (23), so is a correct expression for $\mathbb{E}[Y_j|\widetilde{Y}_j]$.

When $1-k_j \neq 0$ we can rearrange (24) for $\mathbb{E}[Y_{0,j}|\widetilde{Y}_j]$:

$$\mathbb{E}[Y_{0,j}|\widetilde{Y}_j] = (1-k_j)^{-1}(\mathbb{E}[Y_j|\widetilde{Y}_j] - k_j\widetilde{Y}_j). \tag{25}$$

By the expression for $k_j$ given in (22), $1-k_j$ is

$$1-k_j = 1 - \frac{1-p_j}{1-\widetilde{p}_j p_j} = \frac{p_j(1-\widetilde{p}_j)}{1-\widetilde{p}_j p_j},$$

so is non-zero when $p_j \neq 0$ and $\widetilde{p}_j \neq 1$. Writing (25) in terms of vectors and matrices yields (18), as required.

### B. Proof of SSDU

This section of the Appendix proves that a network trained with SSDU's loss weighting $(\mathbb{1} - M_\Lambda)M_\Omega$ satisfies

$$(\mathbb{1} - K)(\mathbb{1} - M_\Lambda M_\Omega)(f_{\theta^*}(\widetilde{Y}) - \mathbb{E}[Y_0|\widetilde{Y}]) = 0. \tag{26}$$

*Proof:* By (6), the minimum of SSDU's loss function (12) gives a function that satisfies

$$\mathbb{E}[(\mathbb{1} - M_\Lambda)M_\Omega(f_{\theta^*}(\widetilde{Y}) - Y)|\widetilde{Y}] = 0 \tag{27}$$

Similarly to Section A of the Appendix, the following derives expressions for $\mathbb{E}[(\mathbb{1} - M_\Lambda)M_\Omega(f_{\theta^*}(\widetilde{Y}) - Y)|\widetilde{Y}]$ under two conditions, $\widetilde{Y}_j \neq 0$ and $\widetilde{Y}_j = 0$, and subsequently find an expression that is true for both. In the following, $\widetilde{m}_j$ and $m_j$ are the $j$th diagonals of $M_\Lambda$ and $M_\Omega$ respectively.

*Case 1.* ($\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j \neq 0]$)*:* When $\widetilde{Y}_j \neq 0$, the $j$th entry is not masked: $\widetilde{m}_j = 1$. Therefore $(1-\widetilde{m}_j)m_j = 0$ and the expression is zero:

$$\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j \neq 0] = 0. \tag{28}$$

*Case 2.* ($\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0]$)*:* When $\widetilde{Y}_j = 0$, $\widetilde{m}_j m_j = 0$, so $(1-\widetilde{m}_j)m_j = m_j$. Therefore

$$\begin{aligned}&\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0] \\ &\quad= \mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0].\end{aligned} \tag{29}$$

As for Case 2 of Section A of the Appendix, we can use the partition theorem to express (29) as a weighted sum:

$$\begin{aligned}&\mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0] \\ &\quad= \mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0 \cap Y_j = 0] \cdot k_j \\ &\qquad+ \mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0 \cap Y_j \neq 0] \cdot (1-k_j),\end{aligned} \tag{30}$$

where $k_j = \mathbb{P}[Y_j = 0|\widetilde{Y}_j = 0]$ as in Section A of the Appendix, given in (22). Taking each term in turn:

- $\mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0 \cap Y_j = 0]$: Since $Y_j = 0$ when it is zeroed by the mask, $m_j = 0$. Therefore
$$\mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0 \cap Y_j = 0] = 0.$$
- $\mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0 \cap Y_j \neq 0]$: When $Y_j \neq 0$, it is not zeroed by the mask, so $m_j = 1$:
$$\begin{aligned}&\mathbb{E}[m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0 \cap Y_j \neq 0] \\ &\quad= \mathbb{E}[f_{\theta^*}(\widetilde{Y})_j - Y_j|\widetilde{Y}_j = 0 \cap Y_j \neq 0].\end{aligned}$$
Further, since $Y_j = Y_{0,j}$ when $Y_j \neq 0$ by the measurement model,
$$\begin{aligned}&\mathbb{E}[f_{\theta^*}(\widetilde{Y})_j - Y_j|\widetilde{Y}_j = 0 \cap Y_j \neq 0] \\ &\quad= \mathbb{E}[f_{\theta^*}(\widetilde{Y})_j - Y_{0,j}|\widetilde{Y}_j = 0].\end{aligned}$$

Substituting the above results in to (30) gives

$$\begin{aligned}&\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0] \\ &\quad= \mathbb{E}[f_{\theta^*}(\widetilde{Y})_j - Y_{0,j}|\widetilde{Y}_j = 0] \cdot (1-k_j).\end{aligned} \tag{31}$$

*Combining Cases 1 and 2:* A correct expression for $\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_j)|\widetilde{Y}_j = 0]$ must be true for both Case 1 and 2, so consistent with both (28) and (31). Consider the candidate

$$\begin{aligned}&\mathbb{E}[(1-\widetilde{m}_j)m_j(f_{\theta^*}(\widetilde{Y})_j - Y_{0,j})|\widetilde{Y}_j] \\ &\quad= (1-k_j)(1-\widetilde{m}_j m_j)\mathbb{E}[f_{\theta^*}(\widetilde{Y})_j - Y_{0,j}|\widetilde{Y}_j].\end{aligned} \tag{32}$$

Equation (32) is consistent with (28) because $(1-\widetilde{m}_j m_j) = 0$ when $\widetilde{Y}_j \neq 0$, and consistent with (31) because $(1-\widetilde{m}_j m_j) = 1$ when $\widetilde{Y}_j = 0$. Using the vector form of (32) and setting $\mathbb{E}[f_{\theta^*}(\widetilde{Y})|\widetilde{Y}] = f_{\theta^*}(\widetilde{Y})$ gives

$$\begin{aligned}&\mathbb{E}[(\mathbb{1} - M_\Lambda)M_\Omega(f_{\theta^*}(\widetilde{Y}) - Y)|\widetilde{Y}] \\ &\quad= (\mathbb{1} - K)(\mathbb{1} - M_\Lambda M_\Omega)(f_{\theta^*}(\widetilde{Y}) - \mathbb{E}[Y_0|\widetilde{Y}]) = 0,\end{aligned} \tag{33}$$

as required.

## ACKNOWLEDGMENT

For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. The computational aspects of this research were supported by the Wellcome Trust Core Award Grant Number 203141/Z/16/Z and the NIHR Oxford BRC. The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health.

## REFERENCES


[1] J. B. Ra and C. Y. Rim, "Fast imaging using subencoding data sets from multiple detectors," *Magn. Reson. Med.*, vol. 30, no. 1, pp. 142–145, 1993.

[2] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: Sensitivity Encoding for Fast MRI," *Magn. Reson. Med.: An Official J. Int. Soc. Magn. Reson. Med.*, vol. 42, pp. 952–62, Nov. 1999.

[3] M. A. Griswold et al., "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magn. Reson. Med.*, vol. 47, pp. 1202–1210, Jun. 2002.

[4] M. Uecker et al., "ESPIRiT-an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA," *Magn. Reson. Med.*, vol. 71, pp. 990–1001, Mar. 2014.

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

[6] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[7] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, pp. 1182–1195, Dec. 2007.

[8] J. C. Ye, "Compressed sensing MRI: A review from signal processing perspective," *BMC Biomed. Eng.*, vol. 1, Dec. 2019, Art. no. 8.

[9] O. N. Jaspan, R. Fleysher, and M. L. Lipton, "Compressed sensing MRI: A review of the clinical literature," *Brit. J. Radiol.*, vol. 88, Dec. 2015, Art. no. 20150487.

[10] S. Wang et al., "Accelerating magnetic resonance imaging via deep learning," in *Proc. IEEE 13th Int. Symp. Biomed. Imag.*, 2016, pp. 514–517.

[11] K. Kwon, D. Kim, and H. Park, "A parallel MR imaging method using multilayer perceptron," *Med. Phys.*, vol. 44, no. 12, pp. 6209–6224, 2017.

[12] K. Hammernik et al., "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.*, vol. 79, no. 6, pp. 3055–3071, 2018.

[13] A. P. Yazdanpanah, O. Afacan, and S. Warfield, "Deep plug-and-play prior for parallel MRI reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3952–3958.

[14] J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov, "RARE: Image reconstruction using deep priors learned without groundtruth," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 6, pp. 1088–1099, Oct. 2020.

[15] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 10–18.

[16] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A. Deep Learning Approach for Image Compressive Sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.

[17] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1828–1837.

[18] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018.

[19] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018.

[20] M. Mardani et al., "Deep generative adversarial neural networks for compressive sensing MRI," *IEEE Trans. Med. Imag.*, vol. 38, no. 1, pp. 167–179, Jan. 2019.

[21] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, Feb. 2019.

[22] R. Ahmad et al., "Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 105–116, Jan. 2020.

[23] S. Wang et al., "DIMENSION: Dynamic MR imaging with both k-space and spatial prior knowledge obtained via multi-supervised network training," *NMR Biomed.*, vol. 35, no. 4, 2022, Art. no. e4131.

[24] Y. Chen et al., "AI-based reconstruction for fast MRI–A systematic review and meta-analysis," *Proc. IEEE*, vol. 110, no. 2, pp. 224–245, Feb. 2022.

[25] J. Zbontar et al., "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018, *arXiv:1811.08839*.

[26] M. Uecker, S. Zhang, D. Voit, A. Karaus, K.-D. Merboldt, and J. Frahm, "Real-time MRI at a resolution of 20 ms," *NMR Biomed.*, vol. 23, no. 8, pp. 986–994, 2010.

[27] H. Haji-Valizadeh et al., "Validation of highly accelerated real-time cardiac cine MRI with radial k-space sampling and compressed sensing in patients at 1.5 T and 3T," *Magn. Reson. Med.*, vol. 79, no. 5, pp. 2745–2751, 2018.

[28] Y. Lim, Y. Zhu, S. G. Lingala, D. Byrd, S. Narayanan, and K. S. Nayak, "3D dynamic MRI of the vocal tract during natural speech," *Magn. Reson. Med.*, vol. 81, no. 3, pp. 1511–1520, 2019.

[29] J. Yoo, K. H. Jin, H. Gupta, J. Yerly, M. Stuber, and M. Unser, "Time-dependent deep image prior for dynamic MRI," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3337–3348, Dec. 2021.

[30] J. I. Tamir, X. Y. Stella, and M. Lustig, "Unsupervised deep basis pursuit: Learning reconstruction without ground-truth data," in *Proc. ISMRM Annu. Meeting*, 2019, Art. no. 0660.

[31] P. Huang et al., "Deep MRI reconstruction without ground truth for training," in *Proc. 27th Annu. Meeting ISMRM*, 2019. [Online]. Available: https://archive.ismrm.org/2019/4668.html

[32] E. K. Cole, J. M. Pauly, S. S. Vasanawala, and F. Ong, "Unsupervised MRI reconstruction with generative adversarial networks," 2020, *arXiv:2008.13065*.

[33] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, "Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data," *Magn. Reson. Med.*, vol. 84, no. 6, pp. 3172–3191, 2020.

[34] S. Liu, P. Schniter, and R. Ahmad, "MRI recovery with a self-calibrated denoiser," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 1351–1355.

[35] H. K. Aggarwal, A. Pramanik, and M. Jacob, "Ensure: Ensemble Stein's unbiased risk estimator for unsupervised learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 1160–1164.

[36] G. Zeng et al., "A review on deep learning MRI reconstruction without fully sampled k-space," *BMC Med. Imag.*, vol. 21, no. 1, pp. 1–11, 2021.

[37] C. Hu, C. Li, H. Wang, Q. Liu, H. Zheng, and S. Wang, "Self-supervised learning for MRI reconstruction with a parallel network training framework," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2021, pp. 382–391.

[38] J. Lehtinen et al., "Noise2noise: Learning image restoration without clean data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2965–2974.

[39] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2129–2137.

[40] J. Batson and L. Royer, "Noise2self: Blind denoising by self-supervision," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 524–533.

[41] N. Moran, D. Schmidt, Y. Zhong, and P. Coady, "Noisier2noise: Learning to denoise from unpaired noisy data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12064–12072.

[42] Y. Xie, Z. Wang, and S. Ji, "Noise2same: Optimizing a self-supervised bound for image denoising," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20320–20330.

[43] A. A. Hendriksen, D. M. Pelt, and K. J. Batenburg, "Noise2Inverse: Self-supervised deep convolutional denoising for tomography," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1320–1335, 2020.

[44] K. Kim and J. C. Ye, "Noise2Score: Tweedie's approach to self-supervised image denoising without clean images," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 864–874, 2021.

[45] W. Gan, Y. Sun, C. Eldeniz, J. Liu, H. An, and U. S. Kamilov, "Deformation-compensated learning for image reconstruction without ground truth," *IEEE Trans. Med. Imag.*, vol. 41, no. 9, pp. 2371–2384, Sep. 2022.

[46] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction," *Med. Phys.*, vol. 44, no. 10, pp. e360–e375, 2017.

[47] R. Flamary, "Astronomical image reconstruction with convolutional neural networks," in *Proc. IEEE 25th Eur. Signal Process. Conf.*, 2017, pp. 2468–2472.

[48] K. Hammernik et al., "Physics-driven deep learning for computational magnetic resonance imaging: Combining physics and machine learning for improved medical imaging," *IEEE Signal Process. Mag.*, vol. 40, no. 1, pp. 98–114, 2023.

[49] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, Nov. 2004.

[50] 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, 2010.

[51] A. Sriram et al., "End-to-end variational networks for accelerated MRI reconstruction," in *Proc. 23rd Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2020, pp. 64–73.
[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.
[53] B. Yaman, S. A. H. Hosseini, S. Moeller, and M. Akçakaya, "Improved supervised training of physics-guided deep learning image reconstruction with multi-masking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 1150–1154.
[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[56] B. Yaman et al., "Multi-mask self-supervised learning for physics-guided neural networks in highly accelerated magnetic resonance imaging," *NMR Biomed.*, vol. 35, no. 12, 2022, Art. no. e4798.
[57] M. Uecker, J. I. Tamir, F. Ong, and M. Lustig, "The BART Toolbox for Computational Magnetic Resonance Imaging," in *Proc. Int. Soc. Magn. Reson. Med.*, vol. 24, 2016. [Online]. Available: https://wwwuser.gwdg.de/∼muecker1/basp-uecker2.pdf
[58] V. Deshpande, D. Nickel, R. Kroeker, S. Kannengiesser, and G. Laub, "Optimized caipirinha acceleration patterns for routine clinical 3D imaging," in *Proc. 20th Annu. Meeting ISMRM*, 2012. [Online]. Available: https://archive.ismrm.org/2012/0104.html
[59] M. Blumenthal, G. Luo, M. Schilling, M. Haltmeier, and M. Uecker, "NLINV-Net: Self-supervised End-2-End learning for reconstructing undersampled radial cardiac real-time data," in *Proc. ISMRM Annu. Meeting*, 2022. [Online]. Available: https://archive.ismrm.org/2022/0499.html
[60] W. Gan et al., "Self-supervised deep equilibrium models for inverse problems with theoretical guarantees," 2022, *arXiv:2210.03837*.
[61] X. Liu, J. Zou, X. Zheng, C. Li, H. Zheng, and S. Wang, "Iterative data refinement for self-supervised MR image reconstruction," 2022, *arXiv:2211.13440*.
[62] A. D. Desai et al., "Noise2Recon: Enabling SNR-robust MRI reconstruction with semi-supervised and self-supervised learning," *Magn. Reson. Med.*, to be published, doi: 10.1002/mrm.29759.
[63] D. Chen, J. Tachella, and M. E. Davies, "Robust equivariant imaging: A fully unsupervised framework for learning to image from noisy and partial measurements," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5647–5656.
[64] C. Millard and M. Chiew, "Simultaneous self-supervised reconstruction and denoising of sub-sampled MRI data with Noisier2Noise," 2022, *arXiv:2210.01696*.
[65] J. Zou et al., "SelfCoLearn: Self-supervised collaborative learning for accelerating dynamic MR imaging," *Bioengineering*, vol. 9, no. 11, 2022, Art. no. 650.
[66] S. Wang et al., "PARCEL: Physics-based unsupervised contrastive representation learning for multi-coil MR imaging," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, Oct. 11, 2022, doi: 10.1109/TCBB.2022.3213669.
[67] B. Yaman, S. A. H. Hosseini, and M. Akcakaya, "Zero-shot physics-guided deep learning for subject-specific MRI reconstruction," in *Proc. Neural Inf. Process. Syst. Workshop Deep Learn. Inverse Problems*, 2021. [Online]. Available: https://openreview.net/forum?id=Nzv2jICkWV7
[68] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2018, pp. 9446–9454.

**Charles Millard** received the M.Sc. degree in physics from Imperial College London, London, U.K. and the doctorate degree in mathematics with biomedical imaging from the University of Oxford, Oxford, U.K. He is currently a Postdoctoral Researcher with the Wellcome Centre for Integrative Neuroimaging, University of Oxford. His research focuses on methods for reconstructing accelerated magnetic resonance imaging acquisitions with compressed sensing and deep learning.

**Mark Chiew** received the B.ASc. degree in engineering physics from the University of British Columbia, Vancouver, BC, Canada, and the Ph.D. degree in medical biophysics from the University of Toronto, Toronto, ON, Canada. From 2012 to 2022, he was a Postdoctoral Researcher and then the Royal Academy of Engineering Research Fellow with the University of Oxford, Oxford, U.K. Since 2022, he has been an Associate Professor with the University of Toronto, and the Scientist with Sunnybrook Research Institute, Toronto, ON. His research interests include the development of acquisition and image reconstruction strategies for magnetic resonance imaging.