


# Joint Supervised and Self-supervised Learning for MRI Reconstruction

George Yiasemis<sup>1,2</sup> 

G.YIASEMIS@NKI.NL

Nikita Moriakov<sup>1,2</sup>

N.MORIAKOV@NKI.NL

Clara I. Sánchez<sup>2</sup>

C.I.SANCHEZGUTIERREZ@UVA.NL

Jan-Jakob Sonke<sup>1,2</sup>

J.SONKE@NKI.NL

Jonas Teuwen<sup>1,2,3</sup>

J.TEUWEN@NKI.NL

<sup>1</sup> Netherlands Cancer Institute <sup>2</sup> University of Amsterdam <sup>3</sup> Radboud University Medical Center

**Editors:** Under Review for MIDL 2025

## Abstract

Magnetic Resonance Imaging (MRI) is a crucial modality but, its inherently slow acquisition process poses challenges in obtaining fully-sampled  $k$ -space data under motion. The lack of fully-sampled acquisitions, serving as ground truths, complicates the training of deep learning (DL) algorithms in a supervised manner. To address this limitation, self-supervised learning (SSL) methods have emerged as a viable alternative, leveraging available subsampled  $k$ -space data to train neural networks for MRI reconstruction. Nevertheless, these approaches often fall short when compared to supervised learning (SL). We propose Joint Supervised and Self-supervised Learning (JSSL), a novel training approach for DL-based MRI reconstruction algorithms aimed at enhancing reconstruction quality in cases where target datasets containing fully-sampled  $k$ -space measurements are unavailable. JSSL operates by simultaneously training a model in a SSL setting, using subsampled data from the target dataset(s), and in a SL manner, utilizing proxy datasets with fully-sampled  $k$ -space data. We demonstrate JSSL’s efficacy using two distinct combinations of target and proxy data. Quantitative and qualitative results showcase substantial improvements over conventional SSL methods. Furthermore, we provide “rule-of-thumb” guidelines for training MRI reconstruction models. Our code is available at <https://github.com/NKI-AI/direct>.

**Keywords:** MRI Reconstruction, Inverse Problems, Deep Learning, Self-supervised Learning

## 1. Introduction

Magnetic Resonance Imaging (MRI) is widely used in clinical practice for its ability to non-invasively visualize detailed anatomical and physiological information. However, the physics of MRI data acquisition, or  $k$ -space, often makes it time-consuming, limiting its utility in scenarios requiring fast imaging, such as image-guided tasks. Accelerating MRI acquisition through subsampled  $k$ -space data can reduce scan times but results in lower-quality reconstructed images with potential artifacts and aliasing (Zbontar et al., 2019).

Recently, numerous MRI reconstruction techniques have employed Deep Learning (DL)-based algorithms (Fessler, 2019; Pal and Rathi, 2022). Trained to produce high-quality images from subsampled  $k$ -space measurements, these methods outperform conventional techniques like Parallel Imaging (Pruessmann et al., 1999) and Compressed Sensing (Candes et al., 2006). Typically, they rely on fully supervised learning (SL) using retrospectively subsampled  $k$ -space data as inputs and fully-sampled  $k$ -space data as ground truth.

However, acquiring fully-sampled datasets, essential for SL training, is often infeasible or costly in certain clinical scenarios (Uecker et al., 2010; Sarma et al., 2014; Kim et al., 2021), such as abdomen, cardiac cine, chest, or prostate imaging, where motion complicates adherence to the Nyquist-Shannon sampling theorem (Shannon, 1949). To address this challenge, recent approaches train DL-based algorithms using self-supervised learning (SSL) on subsampled  $k$ -space data without ground truth, fully-sampled datasets (Yaman et al., 2020; Hu et al., 2021; Zhou et al., 2022a; Millard and Chiew, 2022). These methods leverage SSL-based mechanisms to reconstruct subsampled MRI data. Appendix A reviews related work on SSL-based MRI reconstruction, providing the background for our approach.

In this work, we propose Joint Supervised and Self-supervised Learning (JSSL), a novel method for training DL-based MRI reconstruction models without ground truth, fully-sampled  $k$ -space data for the target domain for SL. JSSL leverages fully-sampled data from *proxy* dataset(s) and subsampled data from *target* dataset(s) to jointly train models using both supervised and self-supervised paradigms. Our contributions include:

- JSSL presents the first approach to combine SL and SSL-based training in proxy and target organ domains within a unified pipeline for accelerated MRI reconstruction.
- We experimentally demonstrate that JSSL outperforms SSL-based MRI reconstruction approaches, with a specific focus on subsampled prostate and cardiac datasets.
- We offer practical “rule-of-thumb” guidelines for selecting appropriate training frameworks for accelerated MRI reconstruction models.

## 2. Materials and Methods

### 2.1. MRI Acquisition and Reconstruction

Recovering images from subsampled  $k$ -space measurements  $\tilde{\mathbf{y}}_{\mathbf{M}}$  in accelerated acquisitions is typically framed as a least-squares optimization problem solved numerically (Yiasemis et al., 2022b). Deep learning (DL) methods streamline this process by learning reconstruction directly from data, utilizing diverse configurations in operation domains, architectural designs, and physics-guided unrolling (Singh et al., 2023). For clarity, all mathematical formulations and operators are detailed in Appendix B.

### 2.2. MRI Reconstruction with Supervised Learning

In supervised learning settings, fully-sampled  $k$ -space datasets are assumed to be available. Let  $\mathcal{D}^{\text{SL}} = \{\mathbf{y}^{(i)}\}_{i=1}^N$  represent such a dataset, which is retrospectively subsampled during training:  $\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)} = \mathbf{U}_{\mathbf{M}_i}(\mathbf{y}^{(i)})$ , and let  $f_{\psi}$  denote a DL-based reconstruction network with parameters  $\psi$ . Note that the architecture of  $f_{\psi}$  can be configured to output image reconstructions,  $k$ -space data, or both, but here we assume that both input and output lie in the image domain. The objective in SL-based MRI reconstruction is to minimize the discrepancy between the fully-sampled and predicted  $k$ -spaces:

$$\psi^* = \arg \min_{\psi} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_K(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}), \quad \hat{\mathbf{y}}^{(i)} = \text{DC}_{\mathbf{M}_i} \left( \tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}, \mathcal{F} \circ \mathcal{E}_{\mathbf{S}} \circ f_{\psi} \left( \tilde{\mathbf{x}}_{\mathbf{M}_i}^{(i)} \right) \right), \quad \tilde{\mathbf{x}}_{\mathbf{M}_i}^{(i)} = \mathcal{A}_{\mathbf{M}_i, \mathbf{S}}^*(\mathbf{y}^{(i)}), \quad (1)$$

or the discrepancy between the fully-sampled and predicted images:

$$\psi^* = \arg \min_{\psi} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_I(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)}), \quad \mathbf{x}^{(i)} = \text{RSS} \circ \mathcal{F}^{-1}(\mathbf{y}^{(i)}), \quad \hat{\mathbf{x}}^{(i)} = \left| f_{\psi}(\tilde{\mathbf{x}}_{\mathbf{M}_i}^{(i)}) \right|, \quad (2)$$



where  $\mathcal{L}_K$  and  $\mathcal{L}_I$  denote arbitrary frequency and image domain loss functions. For unseen data  $\tilde{\mathbf{y}}_{\mathbf{M}}^{\text{inf}}$  a prediction is estimated as  $\hat{\mathbf{x}} = \left| f_{\psi^*}(\tilde{\mathbf{x}}_{\mathbf{M}}^{\text{inf}}) \right|$ ,  $\tilde{\mathbf{x}}_{\mathbf{M}}^{\text{inf}} = \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1}(\tilde{\mathbf{y}}_{\mathbf{M}}^{\text{inf}})$ .

### 2.3. MRI Reconstruction with Self-supervised Learning

When fully-sampled  $k$ -space data are unavailable, DL models can be trained using SSL. Let  $\mathcal{D}^{\text{SSL}} = \{\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}\}_{i=1}^N$  represent a dataset of subsampled acquisitions, where each instance  $\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}$  is sampled from a set  $\mathbf{M}_i$ . In SSL, training involves partitioning the acquired subsampled measurements (Yaman et al., 2020). For each sample  $\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}$ , the sampling pattern  $\mathbf{M}_i$  is divided into two disjoint subsets,  $\Theta_i$  and  $\Lambda_i$ , followed by projecting  $\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}$  onto both:

$$\Theta_i \cup \Lambda_i = \mathbf{M}_i, \quad \Theta_i \cap \Lambda_i = \emptyset, \quad \tilde{\mathbf{y}}_{\Theta_i}^{(i)} = \mathbf{U}_{\Theta_i}(\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}), \quad \tilde{\mathbf{y}}_{\Lambda_i}^{(i)} = \mathbf{U}_{\Lambda_i}(\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}). \quad (3)$$

Subsequently, one partition is used as input to the reconstruction network, while the other serves as the target. The objective loss function is formulated in the  $k$ -space domain:

$$\begin{aligned} \psi^* &= \arg \min_{\psi} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_K(\tilde{\mathbf{y}}_{\Theta_i}^{(i)}, \hat{\mathbf{y}}_{\Theta_i \Lambda_i}^{(i)}), \\ \hat{\mathbf{y}}_{\Theta_i \Lambda_i}^{(i)} &= \mathbf{U}_{\Theta_i} \left( \text{DC}_{\Lambda_i}(\tilde{\mathbf{y}}_{\Lambda_i}^{(i)}, \hat{\mathbf{y}}_{\Lambda_i}^{(i)}) \right), \quad \hat{\mathbf{y}}_{\Lambda_i}^{(i)} = \mathcal{F} \circ \mathcal{E}_{\mathbf{S}} \circ f_{\psi}(\tilde{\mathbf{x}}_{\Lambda_i}^{(i)}), \quad \tilde{\mathbf{x}}_{\Lambda_i}^{(i)} = \mathcal{A}_{\Lambda_i, \mathbf{S}}^*(\tilde{\mathbf{y}}_{\mathbf{M}_i}^{(i)}). \end{aligned} \quad (4)$$

The loss can equivalently be computed in the image domain as follows:

$$\psi^* = \arg \min_{\psi} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_I(\tilde{\mathbf{x}}_{\Theta_i}^{(i)}, \hat{\mathbf{x}}^{(i)}), \quad \tilde{\mathbf{x}}_{\Theta_i}^{(i)} = \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1}(\tilde{\mathbf{y}}_{\Theta_i}^{(i)}), \quad \hat{\mathbf{x}}^{(i)} = \left| \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1}(\hat{\mathbf{y}}_{\Theta_i \Lambda_i}^{(i)}) \right|. \quad (5)$$

While most SSL-based MRI reconstruction methods rely on loss calculations in the frequency domain (Yaman et al., 2020; Millard and Chiew, 2022; Hu et al., 2021, 2024), some studies have explored dual-domain losses (Zhou et al., 2022b; Yan et al., 2023).

For unseen data  $\tilde{\mathbf{y}}_{\mathbf{M}}^{\text{inf}}$  in SSL settings, a prediction is estimated as outlined below:

$$\hat{\mathbf{x}} = \left| \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1} \circ \text{DC}_{\mathbf{M}}(\tilde{\mathbf{y}}_{\mathbf{M}}^{\text{inf}}, \hat{\mathbf{y}}_{\mathbf{M}}^{\text{inf}}) \right|, \quad \hat{\mathbf{y}}_{\mathbf{M}}^{\text{inf}} = \mathcal{F} \circ \mathcal{E}_{\mathbf{S}} \circ f_{\psi^*}(\tilde{\mathbf{x}}_{\mathbf{M}}^{\text{inf}}), \quad \tilde{\mathbf{x}}_{\mathbf{M}}^{\text{inf}} = \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1}(\tilde{\mathbf{y}}_{\mathbf{M}}^{\text{inf}}). \quad (6)$$

### 2.4. Joint Supervised and Self-supervised Learning

We introduce Joint Supervised and Self-supervised Learning, a novel approach for training DL-based MRI reconstruction models when reference data are unavailable in the target organ domain. JSSL integrates SSL using subsampled measurements from target domain(s) with SL using fully-sampled acquisitions from proxy datasets in other organ domains. By leveraging knowledge from proxy datasets, JSSL aims to surpass the performance of conventional SSL methods that rely solely on subsampled target data. Figure 1 illustrates the end-to-end JSSL pipeline. A theoretical rationale for JSSL is provided in Appendix C.

**JSSL Training Framework** To implement JSSL, we construct the overall loss function with two components: one for supervised learning and another for self-supervised learning. For simplicity we assume a single target and a single proxy dataset.

**Supervised Learning Loss** The SL loss is calculated on the proxy dataset, which contains fully-sampled  $k$ -space data. It is formulated as follows:

$$\mathcal{L}_{\psi}^{\text{SL}} := \mathcal{L}_{I\psi}^{\text{SL}} + \mathcal{L}_{K\psi}^{\text{SL}} = \frac{1}{N_{\text{prx}}} \sum_{i=1}^{N_{\text{prx}}} \left[ \mathcal{L}_I(\mathbf{x}^{\text{prx},(i)}, \hat{\mathbf{x}}^{\text{prx},(i)}) + \mathcal{L}_K(\mathbf{y}^{\text{prx},(i)}, \hat{\mathbf{y}}^{\text{prx},(i)}) \right]. \quad (7)$$

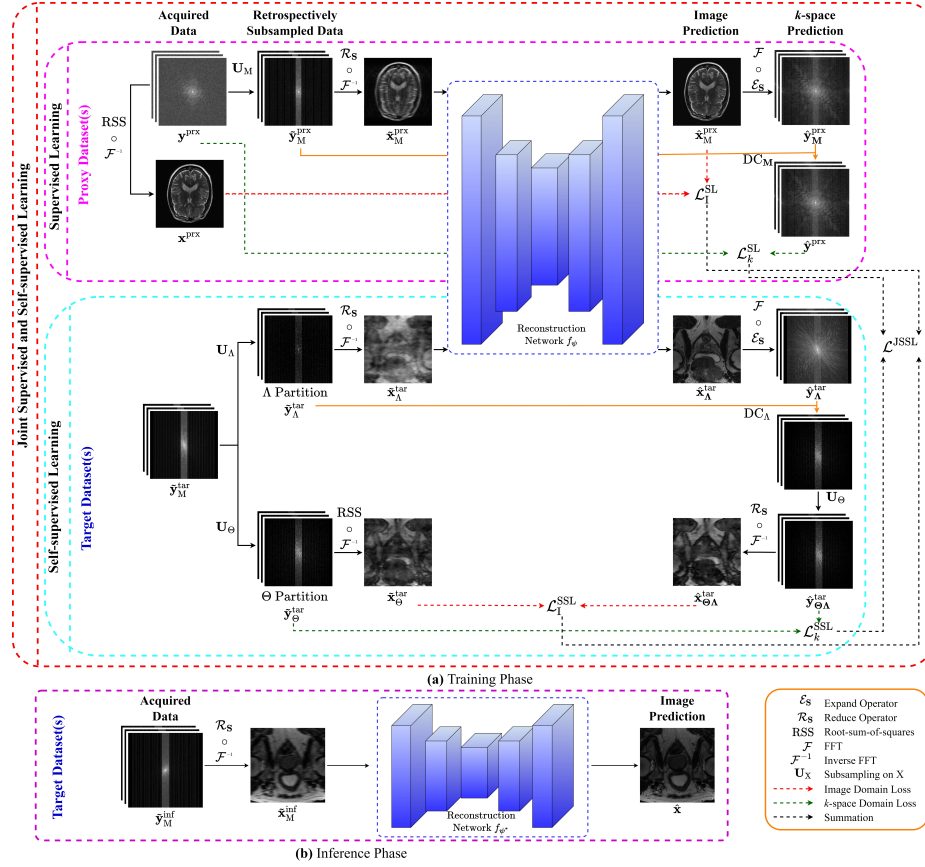


Figure 1: JSSL (a) training and (b) inference phases.

Here,  $\mathbf{x}^{\text{prx},(i)}$ ,  $\hat{\mathbf{x}}^{\text{prx},(i)}$  represent the ground truth and predicted images, respectively, for the  $i$ -th sample in the proxy dataset, while  $\mathbf{y}^{\text{prx},(i)}$ ,  $\hat{\mathbf{y}}^{\text{prx},(i)}$  represent the fully-sampled and predicted  $k$ -spaces, respectively, as defined in Sec. 2.2.

**Self-supervised Learning Loss** The SSL loss is calculated using the target dataset, consisting of subsampled  $k$ -space data without ground truth. Motivated by SSL-based methods (Zhou et al., 2022a,b) which established improved performance when using dual-domain loss, we calculate the SSL loss in both the image and  $k$ -space domains as follows:

$$\mathcal{L}_{\psi}^{\text{SSL}} := \mathcal{L}_{\text{I}\psi}^{\text{SSL}} + \mathcal{L}_{\text{K}\psi}^{\text{SSL}} = \frac{1}{N_{\text{tar}}} \sum_{i=1}^{N_{\text{tar}}} \left[ \mathcal{L}_{\text{K}} \left( \tilde{\mathbf{y}}_{\Theta_i}^{\text{tar},(i)}, \hat{\mathbf{y}}_{\Theta_i \Lambda_i}^{\text{tar},(i)} \right) + \mathcal{L}_{\text{I}} \left( \tilde{\mathbf{x}}_{\Theta_i}^{\text{tar},(i)}, \hat{\mathbf{x}}_{\Theta_i}^{\text{tar},(i)} \right) \right], \quad (8)$$

$$\tilde{\mathbf{x}}_{\Theta_i}^{\text{tar},(i)} = \text{RSS} \circ \mathcal{F}^{-1}(\tilde{\mathbf{y}}_{\Theta_i}^{\text{tar},(i)}), \quad \hat{\mathbf{x}}_{\Theta_i}^{\text{tar},(i)} = \left| f_{\psi}(\tilde{\mathbf{x}}_{\Lambda_i}^{\text{tar},(i)}) \right|,$$

where,  $\tilde{\mathbf{x}}_{\Lambda_i}^{\text{tar},(i)}$ ,  $\tilde{\mathbf{y}}_{\Theta_i}^{\text{tar},(i)}$ ,  $\hat{\mathbf{y}}_{\Theta_i \Lambda_i}^{\text{tar},(i)}$  are as defined in Sec. 2.3.

**JSSL Loss** The JSSL loss is a fundamental component of our approach, defined as the combination of the SL and SSL losses:  $\mathcal{L}_{\psi}^{\text{JSSL}} := \mathcal{L}_{\psi}^{\text{SL}} + \mathcal{L}_{\psi}^{\text{SSL}}$  and the model's parameters are updated during training such that  $\psi^* = \arg \min_{\psi} \mathcal{L}_{\psi}^{\text{JSSL}}$ .

**JSSL at Inference** During the inference phase, assuming  $\tilde{\mathbf{y}}_{\text{M}}^{\text{inf}}$  denotes the subsampled  $k$ -space data, for subsampled  $k$ -space data  $\tilde{\mathbf{y}}_{\text{M}}^{\text{inf}}$  the trained JSSL reconstruction model  $f_{\psi^*}$  estimates the underlying image as follows:  $\hat{\mathbf{x}} = \left| f_{\psi^*}(\tilde{\mathbf{x}}_{\text{M}}^{\text{inf}}) \right|$ , where  $\tilde{\mathbf{x}}_{\text{M}}^{\text{inf}} = \mathcal{R}_{\text{S}} \circ \mathcal{F}^{-1}(\tilde{\mathbf{y}}_{\text{M}}^{\text{inf}})$ .

## 2.5. Coil Sensitivity Prediction

An initial approximation of coil sensitivity maps is derived from the autocalibration signal (ACS) (McKenzie et al., 2002). While SSL-based approaches use this initial approximation (Yaman et al., 2020; Desai et al., 2023; Zhou et al., 2022b), or employ expensive algorithms like Espirit (Uecker et al., 2013), our JSSL approach takes this initial estimation and feeds it as input to a Sensitivity Map Estimator (SME) similarly to (Millard and Chiew, 2022; Zhang et al., 2024; Hu et al., 2024). The SME aims to refine the sensitivity maps and it is trained end-to-end with the reconstruction model and is integrated in all training setups.

## 3. Experiments

### 3.1. Datasets

We utilized fully-sampled multi-coil  $k$ -space data from the fastMRI brain, fastMRI knee (Zbontar et al., 2019), fastMRI prostate T2 (Tibrewala et al., 2023), and CMRxRecon 2023 cardiac cine MRI (Wang et al., 2024) datasets. Their characteristics and data-splitting parameters are summarized in Appendix D.1. To evaluate JSSL, we paired target and proxy datasets in two experimental sets: **(A) Target:** Prostate, **Proxy:** Brain + Knee; **(B) Target:** Cardiac, **Proxy:** Brain + Knee + Prostate. During training, target datasets were retrospectively subsampled, with fully-sampled target data reserved only for inference. Proxy datasets were also retrospectively subsampled for training, with their fully-sampled measurements used to compute the SL loss component of  $\mathcal{L}_\psi^{\text{JSSL}}$ .

### 3.2. Subsampling Schemes

For our experiments, we applied random uniform subsampling to brain data and used equispaced subsampling for the knee, prostate, and cardiac data, commonly employed in the literature (Zbontar et al., 2019; Wang et al., 2024). During training, we randomly selected acceleration factors of  $R = 4, 8, 12$  (only for **A**), and 16 (only for **B**). We retained 8%, 4%, 3%, and 2% of the fully-sampled ACS lines for  $R = 4, 8, 12$ , and 16, respectively. During inference, all these acceleration factors were tested.

**SSL Subsampling Partitioning** During the training of any SSL-based method, including JSSL, we split the subsampled data into two distinct sets, as explained in Sec. 2.3. Specifically,  $\Theta_i$  was obtained from  $\mathbf{M}_i$  using a 2D Gaussian sampling approach with a standard deviation of 3.5 pixels (see Appendix D.2), motivated by reports of Gaussian surpassing uniform partitioning (Yaman et al., 2020). Consequently, we set  $\Lambda_i = \mathbf{M}_i \setminus \Theta_i$ . Furthermore, the ratio  $q_i = \frac{|\Theta_i|}{|\mathbf{M}_i|}$  was chosen randomly between 0.3 and 0.8. Note that each  $\Lambda_i$  included a  $w \times w = 4 \times 4$  window in the ACS region to enhance SME module training.

### 3.3. Implementation & Optimization

**Model Architecture** We employed vSHARP, a physics-guided deep learning method unrolled across  $T$  iterations, previously utilized for accelerated MRI reconstruction (Yiasemis et al., 2023, 2025; Lyu et al., 2024). Details of the hyperparameters are provided in Appendix D.3, while the choice of loss functions and optimization parameters are outlined in Appendices D.4 and D.5, respectively. The SME module incorporated a 2D U-Net with

four scales and 16 filters in the first scale. Since training frameworks are model-agnostic, we extend our experiments using two additional architectures in Appendix E.1.

### 3.4. Training Setups Comparison

We conducted the following experiments: (1) SSL in the target domain, (2) SSL in both target and proxy domains (SSL ALL), (3) SL in the target domain, (4) SL in both proxy and target domains (SL ALL), (5) SL in proxy domains only (SL PROXY), and (6) JSSL. Our primary goal was to assess JSSL against SSL approaches in scenarios where fully-sampled target data are unavailable. To verify that JSSL’s performance does not simply result from using a larger dataset, we included SSL ALL, which incorporates all available data (target + proxy) under an SSL strategy. SL methods served as a reference, though their results are naturally expected to be superior when fully-sampled target data are accessible. For a deeper analysis of JSSL and its performance across various configurations, we provide additional experiments in Appendix E.2.

### 3.5. Evaluation

The performance of our experiments was evaluated on the target test sets using three metrics: SSIM, PSNR, NMSE (Yiasemis et al., 2024). Model checkpoints were selected based on their validation set performance. Statistical tests were performed to assess whether the top-performing method in each category achieved a significant improvement over the others (Appendix D.6). An asterisk (\*) denotes that the average best method (bold) was not found to be statistically significantly better than the corresponding method.

### 3.6. Results

Table 1: Results for fastMRI prostate (target) using brain and knee (proxy) datasets.

Setup	2x			4x			8x			16x		
	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE
SL	<b>0.974</b> $\pm 0.010$	<b>41.8</b> $\pm 2.3$	<b>0.002</b> $\pm 0.001$	<b>0.930</b> $\pm 0.022$	<b>37.5</b> $\pm 1.8$	<b>0.005</b> $\pm 0.002$	<b>0.868</b> $\pm 0.033$	<b>33.9</b> $\pm 1.6$	<b>0.011</b> $\pm 0.003$	<b>0.799</b> $\pm 0.045$	<b>31.0</b> $\pm 1.6$	<b>0.021</b> $\pm 0.005$
SL ALL	0.969 $\pm 0.012$	41.1 $\pm 2.3$	0.002 $\pm 0.001$	0.922 $\pm 0.024$	36.9 $\pm 1.8$	0.005 $\pm 0.002$	0.854 $\pm 0.035$	33.2 $\pm 1.5$	0.013 $\pm 0.003$	0.771 $\pm 0.049$	30.0 $\pm 1.6$	0.026 $\pm 0.006$
SL PROXY	0.961 $\pm 0.016$	39.8 $\pm 2.4$	0.003 $\pm 0.002$	0.914 $\pm 0.026$	36.4 $\pm 1.8$	0.006 $\pm 0.002$	0.839 $\pm 0.041$	32.5 $\pm 1.7$	0.015 $\pm 0.004$	0.733 $\pm 0.051$	28.6 $\pm 1.5$	0.035 $\pm 0.008$
SSL	0.956 $\pm 0.015$	38.8 $\pm 2.6$	0.004 $\pm 0.002$	0.891 $\pm 0.030$	34.7 $\pm 2.0$	0.009 $\pm 0.003$	0.801 $\pm 0.038$	31.1 $\pm 1.5$	0.020 $\pm 0.005$	0.707 $\pm 0.050$	28.0 $\pm 1.6$	0.041 $\pm 0.008$
SSL ALL	0.953 $\pm 0.016$	38.6 $\pm 2.5$	0.004 $\pm 0.002$	0.892 $\pm 0.031$	34.8 $\pm 2.0$	0.009 $\pm 0.004$	0.801 $\pm 0.041$	31.1 $\pm 1.6$	0.020 $\pm 0.006$	0.699 $\pm 0.052$	27.8 $\pm 1.6$	0.043 $\pm 0.010$
JSSL	<b>0.965</b> $\pm 0.015$	<b>39.5</b> $\pm 2.8$	<b>0.003</b> $\pm 0.002$	<b>0.918</b> $\pm 0.026$	<b>36.4</b> $\pm 1.9$	<b>0.006</b> $\pm 0.002$	<b>0.842</b> $\pm 0.038$	<b>32.5</b> $\pm 1.6$	<b>0.015</b> $\pm 0.004$	<b>0.752</b> $\pm 0.053$	<b>29.3</b> $\pm 1.6$	<b>0.030</b> $\pm 0.007$

Table 2: Results for cardiac (target) using brain, knee and prostate (proxy) datasets.

Setup	2x			4x			8x			12x		
	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE
SL	<b>0.991</b> $\pm 0.003$	<b>48.1</b> $\pm 2.5$	<b>0.004</b> $\pm 0.003$	<b>0.984</b> $\pm 0.005$	<b>45.7</b> $\pm 2.6$	<b>0.006</b> $\pm 0.002$	<b>0.965</b> $\pm 0.011$	<b>40.6</b> $\pm 2.2$	<b>0.018</b> $\pm 0.007$	<b>0.946</b> $\pm 0.018$	<b>37.8</b> $\pm 2.3$	<b>0.035</b> $\pm 0.015$
SL ALL	0.987 $\pm 0.004$	46.5 $\pm 2.6$	0.005 $\pm 0.004$	0.979 $\pm 0.006$	44.5 $\pm 1.9$	0.007 $\pm 0.003$	0.956 $\pm 0.012$	39.4 $\pm 1.9$	0.024 $\pm 0.008$	0.932 $\pm 0.019$	36.5 $\pm 2.0$	0.047 $\pm 0.016$
SL PROXY	0.875 $\pm 0.037$	39.8 $\pm 2.0$	0.022 $\pm 0.009$	0.880 $\pm 0.035$	37.6 $\pm 2.0$	0.036 $\pm 0.012$	0.848 $\pm 0.034$	33.1 $\pm 1.7$	0.099 $\pm 0.027$	0.810 $\pm 0.041$	30.0 $\pm 2.2$	0.211 $\pm 0.079$
SSL	0.944 $\pm 0.017$	41.2 $\pm 2.1$	0.016 $\pm 0.007$	0.902 $\pm 0.020$	36.2 $\pm 2.0$	0.049 $\pm 0.014$	0.854 $\pm 0.025$	33.2 $\pm 1.7$	0.097 $\pm 0.020$	0.817 $\pm 0.032$	<b>31.2</b> $\pm 1.9$	<b>0.153</b> $\pm 0.038$
SSL ALL	0.974 $\pm 0.006$	44.0 $\pm 1.9$	0.009 $\pm 0.005$	0.929 $\pm 0.016$	37.9 $\pm 1.9$	0.033 $\pm 0.011$	0.862 $\pm 0.026$	33.0 $\pm 1.7$	0.102 $\pm 0.026$	0.814 $\pm 0.034$	30.3 $\pm 2.0$	0.191 $\pm 0.059$
JSSL	<b>0.975</b> $\pm 0.007$	<b>45.5</b> $\pm 2.6$	<b>0.006</b> $\pm 0.004$	<b>0.944</b> $\pm 0.013$	<b>39.2</b> $\pm 2.6$	<b>0.025</b> $\pm 0.010$	<b>0.893</b> $\pm 0.022$	<b>34.3</b> $\pm 1.8$	<b>0.077</b> $\pm 0.023$	<b>0.848</b> $\pm 0.032$	31.1 $\pm 2.1$ *	0.161 $\pm 0.059$ *

**Training Setups Comparison** The quantitative results of our comparative studies are summarized in Tables 1 and 2, which detail metric averages and statistical significance. Box plots illustrating these results are presented in Appendix F (Figures S4 and S5). As



Figure 2: Example reconstructions (experiment set **A**) from each training setup.

expected, supervised methods consistently produced the best reconstruction results across both experimental setups.

From Tab. 1 and Fig. S4, it is evident that in experiment set **A** (prostate as target), JSSL demonstrated superior reconstruction performance across all acceleration factors and metrics compared to both SSL and SSL utilizing all data (SSL ALL). Notably, JSSL approached the performance of supervised methods (SL, SL ALL), particularly at  $R = 2, 4, 8$ . The use of proxy datasets in SSL settings (SSL ALL) did not enhance performance over SSL alone. Similarly, supervised training on all data (SL ALL) offered no significant advantage over SL alone. In SL PROXY, where training relied solely on proxy datasets, out-of-distribution inference on the prostate dataset resulted in better reconstruction quality than SSL. However, JSSL outperformed SL PROXY in SSIM across all acceleration factors and matched or exceeded pSNR and NMSE, except at  $R = 2$ , where SL PROXY showed a slight edge.

In experiment set **B** (cardiac as target), similar patterns were observed. JSSL consistently outperformed other SSL methods, except at  $R = 12$ , where SSL achieved slightly better (but non-significant) pSNR and NMSE, as shown in Tab. 2 and Fig. S5. Unlike in **A**, SSL ALL showed performance improvements over SSL for cardiac data. SL PROXY, however, performed worse than all other methods.

For qualitative analysis, Figures 2, 3, S6, and S7 display sample reconstructions. At lower accelerations ( $R = 2, 4$ ), all methods accurately reconstructed prostate data. At higher accelerations, supervised, SL PROXY, and JSSL setups exhibited fewer artifacts compared to SSL and SSL ALL (Figures 2, S6). A similar trend was observed for cardiac data, where SSL-based reconstructions were visually weaker, particularly at high accelerations ( $R = 8, 12$ ), yielding highly aliased images. Consistent with the quantitative results, out-of-distribution inference (SL PROXY) reconstructions exhibited noticeable artifacts.

#### 4. Discussion and Conclusion

This study introduces Joint Supervised and Self-supervised Learning, a novel training framework aimed at improving MRI reconstruction quality when fully-sampled  $k$ -space data are



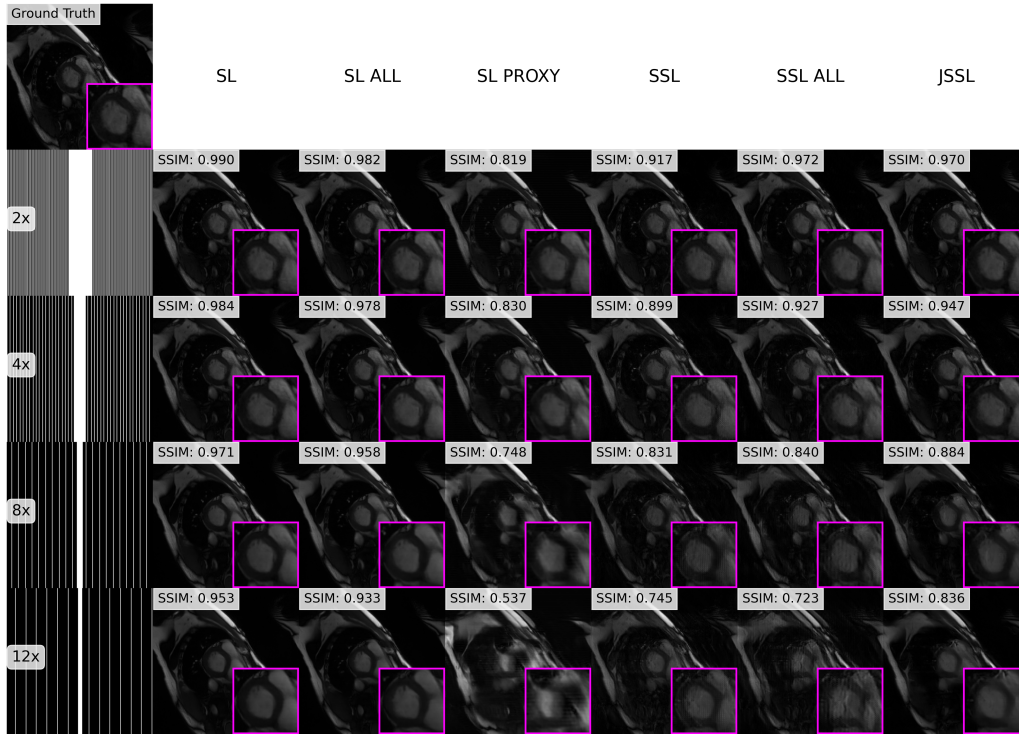


Figure 3: Example reconstructions (experiment set **B**) from each training setup.

unavailable for the target domain. By integrating SL on fully-sampled proxy datasets with SSL on subsampled target datasets, JSSL offers a practical alternative to SSL methods, achieving superior reconstruction quality when acquiring fully-sampled data is infeasible.

Our results demonstrate that JSSL consistently yields higher reconstruction quality across various accelerations, even when proxy datasets differ anatomically from the target dataset. Further experiments in Appendix E.2 confirm that JSSL maintains superior performance under different configurations as well. Additionally, as demonstrated in Appendix E.1, JSSL achieves consistent improvements across different model architectures, showcasing its robustness and independence from specific architectural choices.

While JSSL shows clear advantages over conventional SSL methods, certain limitations are discussed in Appendix G. Moreover, SL remains the preferred approach when fully-sampled ground truth data are available for the target domain, as it offers the highest reconstruction quality due to complete and accurate training data. However, the goal of this work is not to compete with SL but to show that JSSL improves SSL performance. Based on our findings, we propose the following “rule-of-thumb” training guidelines:

- (1) Use SL when fully-sampled ground truth data are available for the target dataset.
- (2) When only subsampled target data are present, and ground truth data are accessible from proxy datasets (e.g., fastMRI or CMRxRecon), adopt the JSSL approach.
- (3) If only proxy ground truth data exist, supervised training on proxy domains can be effective, particularly when proxies are anatomically similar to the target domain. Avoid this approach if proxies are dissimilar.
- (4) In scenarios with only subsampled target data proceed with SSL.



## Acknowledgments

This work was supported by institutional grants of the Dutch Cancer Society and of the Dutch Ministry of Health, Welfare and Sport. The authors would like to acknowledge the Research High Performance Computing (RHPC) facility of the Netherlands Cancer Institute (NKI).

## References

- Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019.
- E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. doi: 10.1109/TIT.2005.862083.
- William Jay Conover. Practical nonparametric statistics, 1999.
- Zhuo-Xu Cui, Chentao Cao, Shaonan Liu, et al. Self-score: Self-supervised learning on score-based models for mri reconstruction, 2022.
- Mohammad Zalbagi Darestani, Jiayu Liu, and Reinhard Heckel. Test-time training can close the natural distribution shift performance gap in deep learning based compressed sensing, 2022. URL <https://arxiv.org/abs/2204.07204>.
- Arjun D. Desai, Batu M. Ozturkler, Christopher M. Sandino, et al. Noise2recon: Enabling snr-robust mri reconstruction with semi-supervised and self-supervised learning. *Magnetic Resonance in Medicine*, 90(5):2052–2070, 2023. doi: <https://doi.org/10.1002/mrm.29759>.
- Jeffrey A Fessler. Optimization methods for mr image reconstruction, 2019.
- Chen Hu, Cheng Li, Haifeng Wang, Qiegen Liu, Hairong Zheng, and Shanshan Wang. Self-supervised learning for MRI reconstruction with a parallel network training framework. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 382–391. Springer International Publishing, 2021. doi: 10.1007/978-3-030-87231-1\_37. URL [https://doi.org/10.1007/978-3-030-87231-1\\_37](https://doi.org/10.1007/978-3-030-87231-1_37).
- Yuyang Hu, Weijie Gan, Chunwei Ying, et al. Spicer: Self-supervised learning for mri with automatic coil sensitivity estimation and reconstruction, 2024. URL <https://arxiv.org/abs/2210.02584>.
- Eu Hyun Kim, Moon Hyung Choi, Young Joon Lee, Dongyeob Han, Mahmoud Mostapha, and Dominik Nickel. Deep learning-accelerated t2-weighted imaging of the prostate: Impact of further acceleration with lower spatial resolution on image quality. *European Journal of Radiology*, 145:110012, December 2021. doi: 10.1016/j.ejrad.2021.110012. URL <https://doi.org/10.1016/j.ejrad.2021.110012>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

- Yilmaz Korkmaz, Tolga Cukur, and Vishal M Patel. Self-supervised mri reconstruction with unrolled diffusion models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 491–501. Springer, 2023.
- Jun Lyu, Chen Qin, Shuo Wang, et al. The state-of-the-art in cardiac mri reconstruction: Results of the cmrxrecon challenge in miccai 2023, 2024. URL <https://arxiv.org/abs/2404.01082>.
- Charles A McKenzie, Ernest N Yeh, Michael A Ohliger, Mark D Price, and Daniel K Sodickson. Self-calibrating parallel imaging with automatic coil sensitivity extraction. *Magnetic Resonance in Medicine*, 47(3):529–538, 2002.
- Charles Millard and Mark Chiew. A theoretical framework for self-supervised mr image reconstruction using sub-sampling via variable density noisier2noise, 2022.
- Arghya Pal and Yogesh Rathi. A review and experimental evaluation of deep learning methods for MRI reconstruction. *J. Mach. Learn. Biomed. Imaging*, 1, March 2022.
- Klaas P Pruessmann, Markus Weiger, Markus B Scheidegger, and Peter Boesiger. Sense: sensitivity encoding for fast mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 42(5):952–962, 1999.
- Saiprasad Ravishankar and Yoram Bresler. Mr image reconstruction from highly under-sampled k-space data by dictionary learning. *IEEE Transactions on Medical Imaging*, 30(5):1028–1041, 2011. doi: 10.1109/TMI.2010.2090538.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24574-4\_28. URL [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- Manoj Sarma, Peng Hu, Stanislas Rapacchi, et al. Accelerating dynamic magnetic resonance imaging (MRI) for lung tumor tracking based on low-rank decomposition in the spatial-temporal domain: a feasibility study based on simulation and preliminary prospective undersampled MRI. *Int. J. Radiat. Oncol. Biol. Phys.*, 88(3):723–731, March 2014.
- C.E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. doi: 10.1109/JRPROC.1949.232969.
- S. S. SHAPIRO and M. B. WILK. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4):591–611, December 1965. ISSN 1464-3510. doi: 10.1093/biomet/52.3-4.591. URL <http://dx.doi.org/10.1093/biomet/52.3-4.591>.
- Dilbag Singh, Anmol Monga, Hector L. de Moura, Xiaoxia Zhang, Marcelo Zibetti, and Ravinder Regatte. Emerging trends in fast mri using deep-learning reconstruction on undersampled k-space data: A systematic review. *Bioengineering*, 10(9), 2023. ISSN 2306-5354. doi: 10.3390/bioengineering10091012. URL <https://www.mdpi.com/2306-5354/10/9/1012>.

- Anuroop Sriram, Jure Zbontar, Tullie Murrell, et al. *End-to-End Variational Networks for Accelerated MRI Reconstruction*, page 64–73. Springer International Publishing, 2020. ISBN 9783030597139. doi: 10.1007/978-3-030-59713-9\\_7. URL [http://dx.doi.org/10.1007/978-3-030-59713-9\\_7](http://dx.doi.org/10.1007/978-3-030-59713-9_7).
- Radhika Tibrewala, Tarun Dutt, Angela Tong, Luke Ginocchio, Mahesh B Keerthivasan, Steven H Baete, Sumit Chopra, Yvonne W Lui, Daniel K Sodickson, Hersh Chandarana, and Patricia M Johnson. Fastmri prostate: A publicly available, biparametric mri dataset to advance machine learning for prostate cancer imaging, 2023.
- Martin Uecker, Shuo Zhang, Dirk Voit, Alexander Karaus, Klaus-Dietmar Merboldt, and Jens Frahm. Real-time MRI at a resolution of 20 ms. *NMR in Biomedicine*, 23(8):986–994, August 2010. doi: 10.1002/nbm.1585. URL <https://doi.org/10.1002/nbm.1585>.
- Martin Uecker, Peng Lai, Mark J. Murphy, et al. ESPIRiT-an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA. *Magnetic Resonance in Medicine*, 71(3):990–1001, May 2013. URL <https://doi.org/10.1002/mrm.24751>.
- Chengyan Wang, Jun Lyu, Shuo Wang, et al. Cmrrecon: A publicly available k-space dataset and benchmark to advance deep learning for cardiac mri. *Scientific Data*, 11(1):687, 2024.
- Burhaneddin Yaman, Seyed Amir Hossein Hosseini, Steen Moeller, Jutta Ellermann, Kâmil Uğurbil, and Mehmet Akçakaya. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. *Magnetic Resonance in Medicine*, 84(6):3172–3191, July 2020. doi: 10.1002/mrm.28378. URL <https://doi.org/10.1002/mrm.28378>.
- Yanghui Yan, Tiejun Yang, Xiang Zhao, Chunxia Jiao, Aolin Yang, and Jianyu Miao. Dc-siamnet: Deep contrastive siamese network for self-supervised mri reconstruction. *Computers in Biology and Medicine*, 167:107619, 2023.
- George Yiasemis, Nikita Moriakov, Dimitrios Karkaloulos, Matthan Caan, and Jonas Teuwen. Direct: Deep image reconstruction toolkit. *Journal of Open Source Software*, 7(73):4278, 2022a. doi: 10.21105/joss.04278. URL <https://doi.org/10.21105/joss.04278>.
- George Yiasemis, Jan-Jakob Sonke, Clarisa Sánchez, and Jonas Teuwen. Recurrent variational network: A deep learning inverse problem solver applied to the task of accelerated mri reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 732–741, June 2022b.
- George Yiasemis, Nikita Moriakov, Jan-Jakob Sonke, and Jonas Teuwen. Deep cardiac mri reconstruction with admm. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 479–490. Springer, 2023.
- George Yiasemis, Clara I Sánchez, Jan-Jakob Sonke, and Jonas Teuwen. On retrospective k-space subsampling schemes for deep mri reconstruction. *Magnetic Resonance Imaging*, 107:33–46, 2024.

- George Yiasemis, Nikita Moriakov, Jan-Jakob Sonke, and Jonas Teuwen. vsharp: Variable splitting half-quadratic admm algorithm for reconstruction of inverse-problems. *Magnetic Resonance Imaging*, 115:110266, 2025. ISSN 0730-725X. doi: <https://doi.org/10.1016/j.mri.2024.110266>. URL <https://www.sciencedirect.com/science/article/pii/S0730725X24002479>.
- Jure Zbontar, Florian Knoll, Anuroop Sriram, et al. fastmri: An open dataset and benchmarks for accelerated mri, 2019.
- Molin Zhang, Junshen Xu, Yamin Arefeen, and Elfar Adalsteinsson. Zero-shot self-supervised joint temporal image and sensitivity map reconstruction via linear latent space. In Ipek Oguz, Jack Noble, Xiaoxiao Li, Martin Styner, Christian Baumgartner, Mirabela Rusu, Tobias Heinmann, Despina Kontos, Bennett Landman, and Benoit Dawant, editors, *Medical Imaging with Deep Learning*, volume 227 of *Proceedings of Machine Learning Research*, pages 1713–1725. PMLR, 10–12 Jul 2024.
- Bo Zhou, Neel Dey, Jo Schlemper, et al. Dsformer: A dual-domain self-supervised transformer for accelerated multi-contrast mri reconstruction, 2022a.
- Bo Zhou, Jo Schlemper, Neel Dey, et al. Dual-domain self-supervised learning for accelerated non-cartesian mri reconstruction. *Medical Image Analysis*, 81:102538, 2022b.

## Appendix A. Related Work

In the realm of self-supervised learning-based MRI reconstruction, among the first works introduced was SSDU (Self-supervised learning via data undersampling) (Yaman et al., 2020). SSDU, inspired by SSL concepts from deep learning, particularly Noise2Self (Batson and Royer, 2019), proposed training a reconstruction model (ResNet CNN with conjugate gradient formulation) by partitioning the undersampled data into two subsets. One subset served as input, and the other as the target, with the loss estimated in the  $k$ -space domain.

An extension of SSDU was proposed in a parallel network framework (Hu et al., 2021), where two networks were trained on each partition of the subsampled  $k$ -space data. A consistency loss minimized the discrepancy between the two networks’ outputs, allowing either network to be used during inference since both networks were trained to produce consistent results.

Further building on SSDU, (Millard and Chiew, 2022) introduced a Noisier2Noise framework, where a second subsampling mask was applied to the already subsampled  $k$ -space data. The employed network, E2EVarNet (Sriram et al., 2020), was trained to recover singly subsampled data from the doubly subsampled version, showing that SSDU is a special case of this broader method. Furthermore, (Millard and Chiew, 2022) provided theoretical justifications for SSDU.

In the realm of diffusion-based MRI reconstruction, a fully-sampled-data-free score-based diffusion model was proposed in (Cui et al., 2022), where the model learned the prior of fully-sampled images from subsampled data in a self-supervised manner. Another diffusion-based approach, SSDiffRecon (Korkmaz et al., 2023), integrated cross-attention transformers with data-consistency blocks in an unrolled architecture. However, diffusion-based methods are outside the scope of our work.

Following the SSDU subsampled data splitting, in (Yan et al., 2023) the authors present DC-SiamNet, which employs two branches with shared weights in a Siamese architecture. Each branch reconstructs from a partition of the  $k$ -space data, and the training is guided by a dual-domain loss that includes image and frequency domain consistency which ensure reconstructed images/ $k$ -spaces are consistent across partitions, along with contrastive loss in the latent space.

A more recent work extended SSDU by introducing SPICER, which includes coil sensitivity estimation based on autocalibration signal (ACS) data and utilizes U-Net-based models for both sensitivity estimation and reconstruction (Hu et al., 2024). Similar sensitivity estimation was also employed in (Millard and Chiew, 2022) within the E2EVarNet framework.

Finally, SSDU has also been applied to reconstruct non-Cartesian MRI data, with the subsampled  $k$ -space split into disjoint parts (Zhou et al., 2022b). In this approach, a variational network is trained using a dual-domain loss similar to (Yan et al., 2023): frequency consistency ensures that reconstructed  $k$ -spaces from each partition match the input data, while image consistency ensures that the reconstructed images are consistent across partitions. Additionally, loss is computed by comparing the reconstructed  $k$ -spaces and images from each partition with those generated when subsampled data is used as input.

Most self-supervised MRI reconstruction methods can be seen as derivatives or extensions of SSDU, with partitioning of undersampled data into disjoint subsets as the funda-

mental idea. This partitioning approach underpins the SSL component of our method, and without loss of generality, SSDU can be considered a representative method in this domain. While recent techniques have incorporated different architectures or loss functions, they largely build upon this core strategy.

Our proposed method, Joint Supervised and Self-Supervised Learning, draws inspiration from these aforementioned approaches. Like most SSL-based methods, it seeks to overcome the challenge of training without fully-sampled  $k$ -space data for the target organ domain. However, JSSL extends the applicability of such techniques by leveraging fully-sampled data from proxy datasets while incorporating subsampled data from the target domain. This enables joint training through both supervised and self-supervised learning, providing a practical solution for scenarios where ground truth fully-sampled data is inaccessible, yet allowing for improved reconstruction performance through the combination of proxy and target datasets.

In the broader context of combining supervised and self-supervised learning, Noise2Recon (Desai et al., 2023) extended SSDU by leveraging both fully-sampled and subsampled data within a single organ domain for reconstruction and denoising, using the E2EVarNet model (Sriram et al., 2020). However, this method’s dependency on fully-sampled data restricts its applicability in scenarios where such data is unavailable.

Another recent approach utilized paired fully-sampled and subsampled data from different modalities for reconstruction of the target modality (Zhou et al., 2022a). While SSL was employed for training, this method still relied on fully-sampled data during both training and inference, which contrasts with our approach that focuses on cases where fully-sampled data is unavailable for the target domain.

Lastly, test-time training (Darestani et al., 2022) is a recent method proposed to handle domain shifts in MRI reconstruction. By re-training models at inference times using a SSL data-consistency loss, it aims to adjust to shifts in data distribution between training and testing, such as moving from one scanner to another. However, this technique operates at inference time, which limits its utility in real-time imaging applications.

## Appendix B. MRI Background

### B.1. MRI Acquisition and Reconstruction

Assuming a fully-sampled MRI acquisition, the ground truth image  $\mathbf{x} \in \mathbb{C}^n$  can be recovered from the fully-sampled multi-coil ( $n_c > 1$ )  $k$ -space  $\mathbf{y} \in \mathbb{C}^{n \times n_c}$  by applying the inverse Fast Fourier transform (FFT), denoted as  $\mathcal{F}^{-1}$ , followed by the RSS method:

$$\mathbf{x} = \text{RSS} \circ \mathcal{F}^{-1}(\mathbf{y}) = \left( \sum_{k=1}^{n_c} |\mathcal{F}^{-1}(\mathbf{y}^k)|^2 \right)^{1/2}, \quad (9)$$

or alternatively, with known sensitivity maps  $\mathbf{S}$ , the SENSE operator:

$$\mathbf{x} = |\text{SENSE} \circ \mathcal{F}^{-1}(\mathbf{y})| = \left| \sum_{k=1}^{n_c} \mathbf{S}^{k*} \mathcal{F}^{-1}(\mathbf{y}^k) \right|, \quad (10)$$



where,

$$\mathbf{x} = \text{RSS}(\mathbf{z}) = \left( \sum_{k=1}^{n_c} |\mathbf{z}^k|^2 \right)^{1/2}, \quad \mathbf{z} \in \mathbb{C}^{n \times n_c} \quad (11)$$

and

$$\text{SENSE}(\mathbf{z}) = \sum_{k=1}^{n_c} \mathbf{S}^{k*} \mathbf{z}^k, \quad \mathbf{z} \in \mathbb{C}^{n \times n_c}. \quad (12)$$

In accelerated acquisitions, the fully-sampled  $k$ -space is subsampled via an operator  $\mathbf{U}_{\mathbf{M}}$ , which selectively retains pixels present in the sampling set  $\mathbf{M}$  and sets others to zero:

$$\mathbf{U}_{\mathbf{M}}(\mathbf{y})_j = \mathbf{y}_j \cdot \mathbb{1}_{\mathbf{M}}(j) = \begin{cases} \mathbf{y}_j, & \text{if } j \in \mathbf{M} \\ 0, & \text{if } j \notin \mathbf{M}. \end{cases} \quad (13)$$

The forward problem of the accelerated acquisition is described by:

$$\tilde{\mathbf{y}}_{\mathbf{M}} = \mathcal{A}_{\mathbf{M}, \mathbf{S}}(\mathbf{x}) + \mathbf{e}, \quad (14)$$

where  $\mathbf{e} \in \mathbb{C}^{n \times n_c}$  represents noise and  $\mathcal{A}_{\mathbf{M}, \mathbf{S}} : \mathbb{C}^n \rightarrow \mathbb{C}^{n \times n_c}$  denotes the forward operator which maps an image to individual coil images using the coil sensitivity maps  $\mathbf{S}$  through the expand operator  $\mathcal{E}_{\mathbf{S}}$ , applies the FFT, denoted by  $\mathcal{F}$ , and applies subsampling via  $\mathbf{U}_{\mathbf{M}}$ :

$$\mathcal{A}_{\mathbf{M}, \mathbf{S}}(\mathbf{x}) = \mathbf{U}_{\mathbf{M}} \circ \mathcal{F} \circ \mathcal{E}_{\mathbf{S}}(\mathbf{x}) = \left\{ \mathbf{U}_{\mathbf{M}} \circ \mathcal{F}(\mathbf{S}^k \mathbf{x}) \right\}_{k=1}^{n_c}. \quad (15)$$

The adjoint operator of  $\mathcal{A}_{\mathbf{M}, \mathbf{S}}$ , denoted by  $\mathcal{A}_{\mathbf{M}, \mathbf{S}}^* : \mathbb{C}^{n \times n_c} \rightarrow \mathbb{C}^n$ , subsamples the input multi-coil data via  $\mathbf{U}_{\mathbf{M}}$ , maps them to the image domain using  $\mathcal{F}^{-1}$ , and reduces them to a single image using  $\mathbf{S}$  via the reduce operator  $\mathcal{R}_{\mathbf{S}}$ :

$$\mathcal{A}_{\mathbf{M}, \mathbf{S}}^*(\mathbf{y}) = \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1} \circ \mathbf{U}_{\mathbf{M}}(\mathbf{y}) = \sum_{k=1}^{n_c} \mathbf{S}^{k*} \mathcal{F}^{-1} \left( \mathbf{U}_{\mathbf{M}}(\mathbf{y}^k) \right). \quad (16)$$

Typically, the process of recovering an image from the subsampled  $k$ -space measurements  $\tilde{\mathbf{y}}_{\mathbf{M}}$  is formulated as a regularized least squares optimization problem:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}'} \frac{1}{2} \left\| \mathcal{A}_{\mathbf{M}, \mathbf{S}}(\mathbf{x}') - \tilde{\mathbf{y}}_{\mathbf{M}} \right\|_2^2 + \mathcal{G}(\mathbf{x}'), \quad (17)$$

where  $\mathcal{G} : \mathbb{C}^n \rightarrow \mathbb{R}$  represents an arbitrary regularization functional that incorporates prior reconstruction information. Equation 17 lacks a closed-form solution, and a solution can only be obtained numerically.

The operator  $\text{DC}_{\mathbf{M}}$  denotes the data consistency operator, which enforces consistency between the available and predicted measurements and is defined as:

$$\text{DC}_{\mathbf{M}}(\mathbf{w}_1, \mathbf{w}_2) = \mathbf{U}_{\mathbf{M}}(\mathbf{w}_1) + \mathbf{U}_{\mathbf{M}^c}(\mathbf{w}_2). \quad (18)$$

## Appendix C. JSSL Theoretical Motivation

The core concept behind JSSL is to leverage both supervised and self-supervised learning to enhance MRI reconstruction of a target dataset, even when the parameters optimized on supervised proxy tasks may not be the most optimal. We hypothesize that introducing a supervised proxy task serves as a form of regularization, reducing the variance of our estimators due to the proxy supervised training on a ‘less noisy’ task. We illustrate this intuition with two simplified examples in Proposition 1 (estimating means of distributions) and Proposition 2 (linear regression), where we assume two distributions - one that we wish to estimate, but we cannot obtain sufficient samples from, and a proxy distribution that is directly accessible. We demonstrate that drawing samples from both distributions (or using only the proxy distribution) can reduce our estimator’s variance and risk.

**Proposition 1** *Consider two distributions  $p_i$ ,  $i = 1, 2$  with means and variances  $\mu_i, \sigma_i$ ,  $i = 1, 2$ , with unknown  $\mu_1$ , and  $\mu_1 \neq \mu_2$ . Then if  $(\mu_1 - \mu_2)^2 < c \frac{\sigma_1^2}{N}$  for some  $c \in (0, 1)$  and  $N \in \mathbb{Z}^+$ , then  $\tilde{x} = \frac{1}{N+K} \sum_{i=1}^{N+K} x_i$  is a lower-variance estimator of  $\mu_1$  compared to  $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ , where  $\{x^{(i)} \sim p_1\}_{i=1}^N$  and  $\{x^{(N+i)} \sim p_2\}_{i=1}^K$  for a choice of a large  $K \in \mathbb{Z}^+$ .*

**Proof** Assume a mixture distribution:

$$p_\pi(x) = \pi \mathcal{N}(x|\mu_1, \sigma_1^2) + (1 - \pi) \mathcal{N}(x|\mu_2, \sigma_2^2).$$

It is then straightforward to compute:

$$\mathbb{E}[p_\pi] = \pi \mu_1 + (1 - \pi) \mu_2$$

and,

$$\mathbb{V}[p_\pi] = \pi \sigma_1^2 + (1 - \pi) \sigma_2^2 + \pi(1 - \pi)(\mu_2 - \mu_1)^2.$$

Drawing  $\{x^{(i)} \sim p_1\}_{i=1}^N$  and  $\{x^{(N+i)} \sim p_2\}_{i=1}^K$ , is approximately equivalent to drawing  $N + K$  samples from the mixture  $p_\pi$  with  $\pi = \frac{N}{N+K}$ . Using bias-variance decomposition, we can compute the expected mean squared errors for the two estimators:

$$\mathbb{E}[(\bar{x} - \mu_1)^2] = \frac{\sigma_1^2}{N},$$

and,

$$\mathbb{E}[(\tilde{x} - \mu_1)^2] = (1 - \pi)^2(\mu_1 - \mu_2)^2 + \frac{\pi \sigma_1^2 + (1 - \pi) \sigma_2^2 + \pi(1 - \pi)(\mu_1 - \mu_2)^2}{N + K}.$$

If  $(\mu_1 - \mu_2)^2 < c \frac{\sigma_1^2}{N}$  for some  $c \in (0, 1)$ , then taking the limit  $K \rightarrow \infty$  and thus  $\pi \rightarrow 0$ , we observe that

$$\mathbb{E}[(\tilde{x} - \mu_1)^2] \rightarrow (\mu_1 - \mu_2)^2 < c \frac{\sigma_1^2}{N} < \frac{\sigma_1^2}{N} = \mathbb{E}[(\bar{x} - \mu_1)^2].$$

■

**Proposition 2** Let  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_p)$  be  $\mathbb{R}^p$ -valued isotropic Gaussian random vector and  $y, \tilde{y}$  be random variables with  $p(y|\mathbf{x}) = \mathcal{N}(y|\mathbf{w}^T \mathbf{x}, \varepsilon^2)$  and  $p(\tilde{y}|\mathbf{x}) = \mathcal{N}(\tilde{y}|\tilde{\mathbf{w}}^T \mathbf{x}, \tilde{\varepsilon}^2)$  for some  $\mathbf{w}, \tilde{\mathbf{w}} \in \mathbb{R}^p$ . Let  $\mathcal{T} = \{(\mathbf{x}_1, \tilde{y}_1), \dots, (\mathbf{x}_K, \tilde{y}_K)\}$  be a training data set with  $K > p$  and consider a maximum likelihood estimator  $\hat{y}(\mathbf{x}; \mathcal{T})$  for  $y$  given  $\mathbf{x}$ , computed using  $\mathcal{T}$ . Then the following holds:

1.  $\text{Bias}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] = (\tilde{\mathbf{w}}^T - \mathbf{w}^T) \mathbf{x}$ .
2.  $\text{Var}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] = \frac{\tilde{\varepsilon}^2}{\sigma^2 K} \|\mathbf{x}\|_2^2$ .
3.  $\mathbb{E}_{(\mathbf{x}, y)}[\hat{y}(\mathbf{x}; \mathcal{T}) - y]^2 \leq p\sigma^2 \|\tilde{\mathbf{w}} - \mathbf{w}\|_2^2 + \frac{p\tilde{\varepsilon}^2}{K} + \varepsilon^2$

**Proof** Let  $\tilde{\mathbf{w}}_{\text{MLE}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \tilde{\mathbf{y}}$  be the MLE estimator for  $\tilde{\mathbf{w}}$ , where the  $K$  rows of  $\mathbf{X} \in \mathbb{R}^{K \times p}$  are given by  $\mathbf{x}_1^T, \dots, \mathbf{x}_K^T$  and the vector  $\tilde{\mathbf{y}}$  is defined as  $\tilde{\mathbf{y}} := (\tilde{y}_1, \dots, \tilde{y}_K) \in \mathbb{R}^K$ . Since  $K > p$ , matrix  $\mathbf{X}$  has full column rank almost surely and thus  $\mathbf{X}^T \mathbf{X}$  is almost surely invertible. Observe that

$$\mathbb{E}_{\mathcal{T}}[\tilde{\mathbf{w}}_{\text{MLE}}^T] = \mathbb{E}_{\mathcal{T}}[(\tilde{\varepsilon}^T + \tilde{\mathbf{w}}^T \mathbf{X}^T) \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1}] = \tilde{\mathbf{w}}^T,$$

since  $\tilde{\varepsilon} := \tilde{\mathbf{y}} - \mathbf{X} \tilde{\mathbf{w}}$  has zero mean, is independent from  $\mathbf{x}_i$ 's and the expectation  $\mathbb{E}_{\mathcal{T}}[\cdot]$  can be rewritten as  $\mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[\mathbb{E}_{\tilde{\varepsilon}}[\cdot]]$ . By definition of estimator bias,

$$\text{Bias}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] = \mathbb{E}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] - \mathbb{E}_{y|\mathbf{x}} y = \mathbb{E}_{\mathcal{T}}[\tilde{\mathbf{w}}_{\text{MLE}}^T \mathbf{x}] - \mathbf{w}^T \mathbf{x} = (\tilde{\mathbf{w}}^T - \mathbf{w}^T) \mathbf{x}.$$

Next,

$$\begin{aligned} \text{Var}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] &= \mathbb{E}_{\mathcal{T}}[\mathbb{E}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] - \hat{y}(\mathbf{x}; \mathcal{T})]^2 = \\ &= \mathbb{E}_{\mathcal{T}}[\tilde{\mathbf{w}}^T \mathbf{x} - (\tilde{\varepsilon}^T + \tilde{\mathbf{w}}^T \mathbf{X}^T) \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}]^2 = \mathbb{E}_{\mathcal{T}}[\tilde{\varepsilon}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}]^2. \end{aligned}$$

The scalar  $(\tilde{\varepsilon}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x})^2$  can be equivalently written as

$$(\tilde{\varepsilon}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x})^T (\tilde{\varepsilon}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}) = \mathbf{x}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \tilde{\varepsilon} \tilde{\varepsilon}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}.$$

Using that  $\mathbb{E}_{\mathcal{T}}[\cdot] = \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[\mathbb{E}_{\tilde{\varepsilon}}[\cdot]]$ , we deduce that

$$\begin{aligned} \mathbb{E}_{\mathcal{T}}[\tilde{\varepsilon}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}]^2 &= \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[\mathbf{x}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbb{E}_{\tilde{\varepsilon}}[\tilde{\varepsilon} \tilde{\varepsilon}^T] \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}] = \\ &= \tilde{\varepsilon}^2 \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[\mathbf{x}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}] = \tilde{\varepsilon}^2 \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[\text{tr}(\mathbf{x}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x})] = \\ &= \tilde{\varepsilon}^2 \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[\text{tr}(\mathbf{x} \mathbf{x}^T (\mathbf{X}^T \mathbf{X})^{-1})] = \tilde{\varepsilon}^2 \text{tr}(\mathbf{x} \mathbf{x}^T \mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[(\mathbf{X}^T \mathbf{X})^{-1}]), \end{aligned}$$

where we use cyclic property of the trace and the fact that  $z = \text{tr}(z)$  for a scalar  $z$ . To compute  $\mathbb{E}_{\mathbf{x}_1, \dots, \mathbf{x}_K}[(\mathbf{X}^T \mathbf{X})^{-1}]$ , we note that, by definition,  $\mathbf{X}^T \mathbf{X}$  follows Wishart distribution  $\mathcal{W}_p(\sigma^2 \mathbf{I}_p, K)$  with  $K$  degrees of freedom and thus  $(\mathbf{X}^T \mathbf{X})^{-1}$  follows inverse Wishart distribution  $\mathcal{W}_p^{-1}(\sigma^{-2} \mathbf{I}_p, K + p + 1)$ , whose mean equals  $\frac{\mathbf{I}_p}{\sigma^2 K}$ . Combining this with the previous results, we conclude

$$\text{Var}_{\mathcal{T}}[\hat{y}(\mathbf{x}; \mathcal{T})] = \frac{\tilde{\varepsilon}^2}{\sigma^2 K} \text{tr}(\mathbf{x} \mathbf{x}^T) = \frac{\tilde{\varepsilon}^2}{\sigma^2 K} \|\mathbf{x}\|_2^2.$$

The final estimate follows from the first two identities and the bias-variance decomposition. ■

Propositions 1 and 2 imply that leveraging a large number of samples from the proxy distribution ( $K \rightarrow \infty$ ) can lead to a significant reduction in the variance of estimators trained under both supervised and self-supervised learning paradigms. Moreover, it highlights how the introduction of bias through supervised learning can be a strategic trade-off to lower variance. Additionally, Proposition 2 sheds light on how the risk associated with our estimator can be influenced by the degree of similarity between the target and proxy distributions.

## Appendix D. Experiments

### D.1. Datasets Information

In our experiments we utilize the fastMRI Knee (Zbontar et al., 2019), fastMRI Brain (Zbontar et al., 2019), fastMRI Prostate (Lyu et al., 2024; Wang et al., 2024), and CMR-Recon Cine (Tibrewala et al., 2023) datasets. The characteristics and data splits are shown below in Table S1.

Table S1: Dataset characteristics and splits.

Dataset		fastMRI Knee	fastMRI Brain	fastMRI Prostate	CMR-Recon Cine
Field Strength		1.5 T, 3.0 T	1.5 T, 3.0 T	3.0 T	3.0 T
Sequence		Proton Density with and without fat suppression	T1-w pre and post contrast, T2-w, FLAIR	T2-w	Cine
Subjects		Healthy or Abnormality present	Healthy or Pathology present	Cancer Patients	Healthy
Acquisition		Cartesian	Cartesian	Cartesian	Cartesian
Fully Sampled or Subsampled		Fully Sampled	Fully Sampled	Three averages (2x) / GRAPPA reconstructed	One average (3x) / GRAPPA reconstructed
No. Coils		15	2-24	10-30	10
No. Volumes Used		973	2,991	312	473
No. Slices Used		34,742	47,426	9,508	3,185
Split Size (No. Volumes/ No. Slices)	Training	973 / 34,742	2,991 / 47,426	218 / 6,647	203 / 1,364
	Validation	-	-	48 / 1,462	111 / 731
	Test	-	-	46 / 1,399	159 / 1,090

In the comparative experiments outlined in Section 3.4 and the alternative configuration experiments (2) and (3) in Appendix E.2, we addressed the imbalance between proxy datasets (brain and knee in experiment set **A**; brain, knee, and prostate in experiment set **B**) and target datasets (prostate in experiment set **A**; cardiac cine in experiment set **B**) by oversampling the proxy data. Unless specified otherwise, this was achieved by duplicating each proxy dataset sample to ensure consistency across experiments.

## D.2. SSL Subsampling Partitioning

Let  $\mathbf{M}_i$  denote the sampling set. Here we describe  $\mathbf{M}_i$  as a sampling mask in the form of a squared array of size  $n = n_x \times n_y$  such that:

$$(\mathbf{M}_i)_{kj} = \begin{cases} 1, & \text{if } (k, j) \text{ is sampled} \\ 0, & \text{if } (k, j) \text{ is not sampled.} \end{cases}$$

The set  $\Theta_i$  is obtained by selecting elements from  $\mathbf{M}_i$  using a variable density 2D Gaussian scheme with a standard deviation of  $\sigma$  pixels and mean vector as the center of the sampling set  $\mathbf{M}_i$ , up to the number of elements determined by a ratio  $q_i$ , determined such that  $q_i = \frac{|\Theta_i|}{|\mathbf{M}_i|}$ , where  $|\cdot|$  here denotes the cardinality. Mathematically, the selection process for  $\Theta_i$  from  $\mathbf{M}_i$  can be described by the following algorithm:

**Data:** Square array  $\mathbf{M}_i$  of size  $n_x \times n_y$ , ratio  $0 < q_i < 1$ , standard deviation  $\sigma$

**Result:** Set  $\Theta_i$

Initialize  $\Theta_i$  as an array of zeros of the same size as  $\mathbf{M}_i$  **while**  $\frac{|\Theta_i|}{|\mathbf{M}_i|} < q_i$  **do**

Generate  $(k, j)$  from  $\mathcal{N}([\frac{n_x}{2}, \frac{n_y}{2}], \sigma^2 \mathbf{I}_2)$  **if**  $(\Theta_i)_{kj} == 0$  **then**  
     |  $(\Theta_i)_{kj} \leftarrow 1$   
**end**

**end**

**Algorithm 1:** Generation of  $\Theta_i$  using Gaussian Sampling

Subsequently, to partition  $\mathbf{M}_i$ , we set  $\Lambda_i = \mathbf{M}_i \setminus \Theta_i$ . Note that by selecting  $q_i = 0$  then  $\Theta_i = \emptyset$ , and for  $q_i = 1$  then  $\Theta_i = \mathbf{M}_i$ .

For our comparison study in Section 3.6 of the main paper for SSL and JSSL experiments we randomly selected the ratio  $q_i$  between 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8. For our alternative configurations study in Appendix E.1, we employed an identical partitioning ratio selection except for the case of a fixed ratio of  $q_i = 0.5$ . In all our JSSL and SSL experiments we used  $\sigma = 3.5$ .

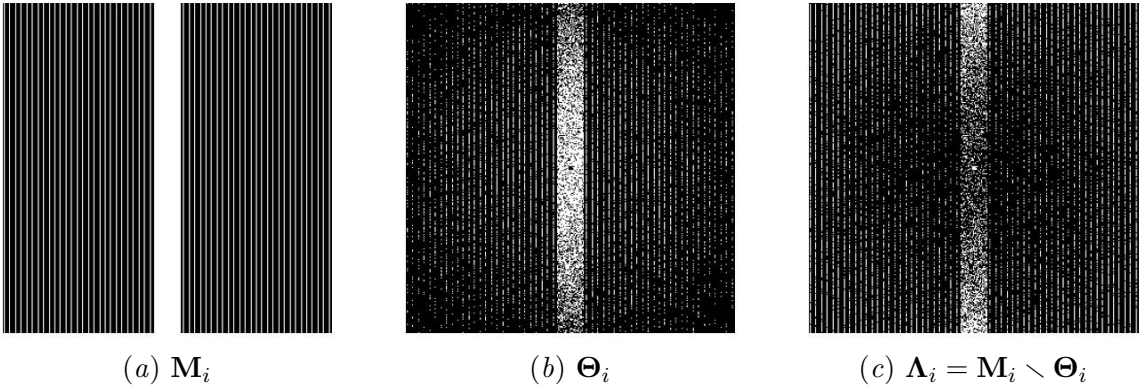


Figure S1: Example of SSL subsampling partitioning with a ratio  $q = \frac{1}{2}$  and  $w = 4$ .

### D.3. Reconstruction Network - vSHARP

In our main experiments, we employed the variable Splitting Half-quadratic ADMM algorithm for Reconstruction of inverse-Problems (vSHARP) as our reconstruction network, which is an unrolled physics-guided DL-based method (Yiasemis et al., 2025) that has previously been applied in accelerated brain, prostate and dynamic cardiac MRI reconstruction (Yiasemis et al., 2025, 2023). The vSHARP algorithm incorporates the half-quadratic variable splitting method to the optimization problem presented in (17), introducing an auxiliary variable  $\mathbf{z}$ :

$$\min_{\mathbf{x}', \mathbf{z}} \frac{1}{2} \left\| \mathcal{A}_{\mathbf{M}, \mathbf{S}}(\mathbf{x}') - \tilde{\mathbf{y}}_{\mathbf{M}} \right\|_2^2 + \mathcal{G}(\mathbf{z}) \quad \text{s.t. } \mathbf{x}' = \mathbf{z}. \quad (19)$$

Subsequently, (19) is iteratively unrolled over  $T$  iterations using the Alternating Direction Method of Multipliers (ADMM). The ADMM formulation consists of three key steps: (a) a denoising step to refine the auxiliary variable  $\mathbf{z}$ , (b) data consistency for the target image  $\mathbf{x}$ , and (c) an update for the Lagrange Multipliers  $\mathbf{u}$  introduced by ADMM:

$$\mathbf{z}_{t+1} = \mathcal{H}_{\psi_{t+1}}(\mathbf{z}_t, \mathbf{x}_t, \mathbf{u}_t / \mu_{t+1}), \quad (20a)$$

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x}'} \left\| \mathcal{A}_{\mathbf{M}, \mathbf{S}}(\mathbf{x}') - \tilde{\mathbf{y}}_{\mathbf{M}} \right\|_2^2 + \mu \left\| \mathbf{x}' - \mathbf{z}_{t+1} + \mathbf{u}_t / \mu_{t+1} \right\|_2^2, \quad (20b)$$

$$\mathbf{u}_{t+1} = \mathbf{u}_t + \mu_{t+1}(\mathbf{x}_{t+1} - \mathbf{z}_{t+1}). \quad (20c)$$

In (20a),  $\mathcal{H}_{\psi_{t+1}}$  denotes a convolutional based DL image denoiser with trainable parameters  $\psi_{t+1}$ , and  $\eta_{t+1}$  a trainable learning rate. At each iteration,  $\mathcal{H}_{\psi_{t+1}}$  takes as input the previous predictions of the three variables and outputs an estimation of the auxiliary variable  $\mathbf{z}$ . Equation 20b is solved numerically by unrolling further a gradient descent scheme over  $T_{\mathbf{x}}$  iterations. The last step in (20c), involves a straightforward computation. The initial approximations for  $\mathbf{x}$  and  $\mathbf{z}$  are taken as:  $\mathbf{x}_0, \mathbf{z}_0 = \mathcal{R}_{\mathbf{S}} \circ \mathcal{F}^{-1}(\tilde{\mathbf{y}}_{\mathbf{M}})$ . Moreover, for  $\mathbf{u}_0$ , vSHARP employs a trainable replication-padding and dilated convolutional-based network represented by  $\mathcal{U}_{\phi}$ :  $\mathbf{u}_0 = \mathcal{U}_{\phi}(\mathbf{x}_0)$ .

For all experiments, we adopted vSHARP with  $T = 12$  optimization steps, utilizing two-dimensional U-Nets (Ronneberger et al., 2015) composed of 4 scales and 32 filters (in the first scale) for  $\{\mathcal{H}_{\theta_t}\}_{t=0}^{T-1}$ . For the data consistency step, we set  $T_{\mathbf{x}} = 10$ .

### D.4. Choice of Loss Functions

In all our experiments, loss was computed as detailed in Section 2.4 employing the following combinations in the image and frequency domains, respectively:

$$\begin{aligned} \mathcal{L}_{\mathbf{I}}^{\text{SL}}, \mathcal{L}_{\mathbf{I}}^{\text{SSL}} &:= 2(1 - \text{SSIM} + \mathcal{L}_1) + \text{HFEN}_1 + \text{HFEN}_2, \\ \mathcal{L}_K^{\text{SL}}, \mathcal{L}_K^{\text{SSL}} &:= 2(\text{NMSE} + \text{NMAE}). \end{aligned}$$

Following, we provide the mathematical definitions of each component.

- Image Domain Loss Functions



- Structural Similarity Index Measure (SSIM) Loss

$$\mathcal{L}_{\text{SSIM}} := 1 - \text{SSIM}, \quad \text{SSIM}(\mathbf{a}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N \frac{(2\mu_{\mathbf{a}_i}\mu_{\mathbf{b}_i} + \gamma_1)(2\sigma_{\mathbf{a}_i\mathbf{b}_i} + \gamma_2)}{(\mu_{\mathbf{a}_i}^2 + \mu_{\mathbf{b}_i}^2 + \gamma_1)(\sigma_{\mathbf{a}_i}^2 + \sigma_{\mathbf{b}_i}^2 + \gamma_2)}, \quad (21)$$

where  $\mathbf{a}_i, \mathbf{b}_i, i = 1, \dots, N$  represent  $7 \times 7$  square windows of  $\mathbf{a}, \mathbf{b}$ , respectively, and  $\gamma_1 = 0.01, \gamma_2 = 0.03$ . Additionally,  $\mu_{\mathbf{a}_i}, \mu_{\mathbf{b}_i}$  denote the means of each window,  $\sigma_{\mathbf{a}_i}$  and  $\sigma_{\mathbf{b}_i}$  represent the corresponding standard deviations. Lastly,  $\sigma_{\mathbf{a}_i\mathbf{b}_i}$  represents the covariance between  $\mathbf{a}_i$  and  $\mathbf{b}_i$ .

- High Frequency Error Norm (HFEN)

$$\mathcal{L}_{\text{HFEN}_k} := \text{HFEN}_k, \quad \text{HFEN}_k(\mathbf{a}, \mathbf{b}) = \frac{\|\mathcal{G}(\mathbf{a}) - \mathcal{G}(\mathbf{b})\|_k}{\|\mathcal{G}(\mathbf{b})\|_k}, \quad (22)$$

where  $\mathcal{G}$  is a Laplacian-of-Gaussian filter (Ravishankar and Bresler, 2011) with kernel of size  $15 \times 15$  and with a standard deviation of 2.5, and  $k = 1$  or 2.

- Mean Average Error (MAE /  $L_1$ ) Loss

$$\mathcal{L}_1(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_1 = \sum_{i=1}^n |a_i - b_i| \quad (23)$$

- $k$ -space Domain Loss Functions

- Normalized Mean Squared Error (NMSE)

$$\mathcal{L}_{\text{NMSE}} := \text{NMSE}, \quad \text{NMSE}(\mathbf{a}, \mathbf{b}) = \frac{\|\mathbf{a} - \mathbf{b}\|_2^2}{\|\mathbf{a}\|_2^2} = \frac{\sum_{i=1}^n (a_i - b_i)^2}{\sum_{i=1}^n a_i^2}. \quad (24)$$

- Normalized Mean Average Error (NMAE)

$$\mathcal{L}_{\text{NMAE}} := \text{NMAE}, \quad \text{NMAE}(\mathbf{a}, \mathbf{b}) = \frac{\|\mathbf{a} - \mathbf{b}\|_1}{\|\mathbf{a}\|_1} = \frac{\sum_{i=1}^n |a_i - b_i|}{\sum_{i=1}^n |a_i|}. \quad (25)$$

The rationale for the loss function components is also drawn from the literature (Yiasemis et al., 2025). In the frequency domain,  $\mathcal{L}_{\text{NMSE}}$  and  $\mathcal{L}_{\text{NMAE}}$  are used to evaluate global similarity to the fully sampled  $k$ -space, with the former addressing larger deviations and the latter focusing on finer discrepancies. In the image domain,  $\mathcal{L}_1$  and  $\mathcal{L}_{\text{SSIM}}$  are commonly combined to optimize pixel-level accuracy and perceptual quality, while  $\mathcal{L}_{\text{HFEN}_k}$  emphasizes the preservation of edges and fine details.

## D.5. Parameter Optimization

We optimized the model parameters using the Adam optimizer (Kingma and Ba, 2017), with parameters  $\epsilon = 10^{-8}$ ,  $(\beta_1, \beta_2) = (0.99, 0.999)$  and initial learning rate (lr) set to 0.003. We also employed a lr scheduler which decayed the lr by a factor of 0.8 every 150,000 training iterations. Experiments were carried out on 2 A6000 RTX GPUs, with a batch size of 2 slices of multi-coil  $k$ -space data assigned to each GPU employing the DIRECT toolkit (Yiasemis et al., 2022a). All models were trained to convergence.

## D.6. Statistical Testing

To determine whether the top-performing method in each category (SL methods, SSL-based methods including JSSL, SSL-based methods with different configurations) significantly outperformed the others, we conducted statistical tests. Initially, we calculated the differences in performance between the best method and the other methods within each category. The Shapiro-Wilk test ([SHAPIRO and WILK, 1965](#)) was used to assess the normality of these differences. If the differences were normally distributed ( $p > \alpha$ ), a paired t-test was performed, alternatively the Wilcoxon signed-rank test ([Conover, 1999](#)) was used. In our reported results we denote with an asterisk instances which the average best method was not found to be statistically significantly better ( $p > \alpha$ ). Note that we set  $\alpha = 0.05$  as the significance level.

## Appendix E. Supplementary Experiments

### E.1. Robustness to Model Choice Experiments

Here, we present supplementary experiments aimed at further validating the efficacy of our proposed JSSL method. These experiments involve a comparative analysis between JSSL and traditional SSL MRI Reconstruction. We adapt the methodologies outlined in Section 3 of our main paper, utilizing two distinct deep MRI reconstruction models instead of the vSHARP architecture:

- Utilizing a plain image domain U-Net (Ronneberger et al., 2015), a non-physics-based model that takes an undersampled-reconstructed image as input and refines it. Specifically, we employ a U-Net with four scales and 64 filters in the first channel.
- Employing an End-to-end Variational Network (E2EVarNet) (Sriram et al., 2020), a physics-based model that executes a gradient descent-like optimization scheme in the  $k$ -space domain. For E2EVarNet, we perform 6 optimization steps using U-Nets with four scales and 16 filters in the first scale.

To estimate sensitivity maps for both architectures, an identical Sensitivity Map Estimation (SME) module was integrated, mirroring the experimental setup outlined in our primary paper.

Both models underwent training and evaluation on data subsampled with acceleration factors of 4, 8, and 16, with ACS ratios of 8%, 4%, and 2% of the data shape. Choices of hyperparameters for JSSL and SSL are the same as in the comparative experiments presented in Section 3.6. Additionally, choices for proxy and target datasets, as well as data splits, are also the same as in the main paper.

Experimental setups were executed on NVIDIA A100 80GB GPUs, utilizing 2 GPUs for U-Net and 1 GPU for E2EVarNet. We employed batch sizes of 2 and 4 for U-Net and E2EVarNet, respectively, on each GPU. The optimization procedures, initial learning rates, and the employed optimizers aligned with those utilized in the main paper.

Table S2: Model architectures parameters.

Model	Parameter	Physics Model	Training	Learning Rate	Inference Time (s)
	Count (millions)		Iterations (k)	Reduction Schedule	per volume
vSHARP	95	ADMM	700	150k	17.7
U-Net	33	-	375	75k	13.1
E2EVarNet	13.5	Gradient Descent in $k$ -space Domain	250	50k	13.9

Table S2 details the model specifics for all considered architectures presented in both the main paper and this section.

#### E.1.1. ROBUSTNESS TO MODEL CHOICE EXPERIMENTS RESULTS

The average results of our supplementary comparative studies to assess JSSL’s robustness to different architecture choices are depicted via box plots in Figures S2 and S3 for U-Net and E2EVarNet, respectively. Corresponding average metrics are provided in Table S3.

Table S3: Robustness to model choice experiments results. An asterisk (\*) denotes that the average best method (bold) was not found to be statistically significantly better than the corresponding method ( $p > 0.05$ ).

Architecture Setup		4x			8x			16x		
		SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE
U-Net	SSL	0.854 $\pm$ 0.031	33.0 $\pm$ 1.6	0.013 $\pm$ 0.004	0.742 $\pm$ 0.040	29.4 $\pm$ 1.4	0.030 $\pm$ 0.006	0.651 $\pm$ 0.051	<b>26.7</b> $\pm$ 1.5	<b>0.055</b> $\pm$ 0.009
	JSSL	<b>0.863</b> $\pm$ 0.031	<b>33.5</b> $\pm$ 1.5	<b>0.012</b> $\pm$ 0.002	<b>0.759</b> $\pm$ 0.042	<b>29.7</b> $\pm$ 1.4	<b>0.027</b> $\pm$ 0.005	<b>0.663</b> $\pm$ 0.051	26.7 $\pm$ 1.4*	0.055 $\pm$ 0.009*
E2EVarNet	SSL	0.874 $\pm$ 0.029	33.7 $\pm$ 1.7	0.011 $\pm$ 0.003	0.770 $\pm$ 0.039	30.0 $\pm$ 1.4	0.025 $\pm$ 0.006	0.670 $\pm$ 0.051	27.0 $\pm$ 1.5	0.051 $\pm$ 0.009
	JSSL	<b>0.888</b> $\pm$ 0.032	<b>34.9</b> $\pm$ 1.6	<b>0.008</b> $\pm$ 0.002	<b>0.784</b> $\pm$ 0.042	<b>30.5</b> $\pm$ 1.4	<b>0.023</b> $\pm$ 0.005	<b>0.678</b> $\pm$ 0.053	<b>27.1</b> $\pm$ 1.5	<b>0.050</b> $\pm$ 0.009

From these results we observe alignment with our original findings: JSSL-trained models consistently outperform SSL-trained models for both architecture choices.

Furthermore, the superior performance of vSHARP and E2EVarNet compared to the U-Net model in both SSL and JSSL settings across all acceleration factors highlights the advantage of adopting physics-guided unrolled models for reconstruction. It is also worth mentioning that vSHARP consistently outperformed E2EVarNet at all accelerations.

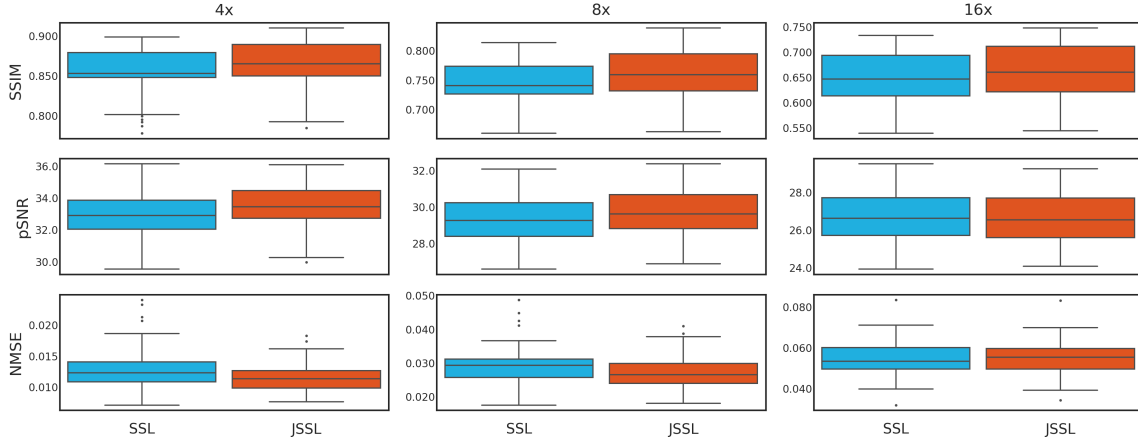


Figure S2: Evaluation metric results for SSL and JSSL methods using an image domain U-Net architecture as a reconstruction network.

## E.2. Alternative Configuration Experiments

To investigate JSSL further and evaluate its performance under different settings, we examine additional configurations for the JSSL and SSL setups. In particular, we perform the following experiments on the first set of experiments (prostate data as target):

- (1) JSSL and SSL in all domains by oversampling 10-fold the target dataset during training to balance better proxy and target data, in comparison to 2-fold in our original experiments in Section 3.4.

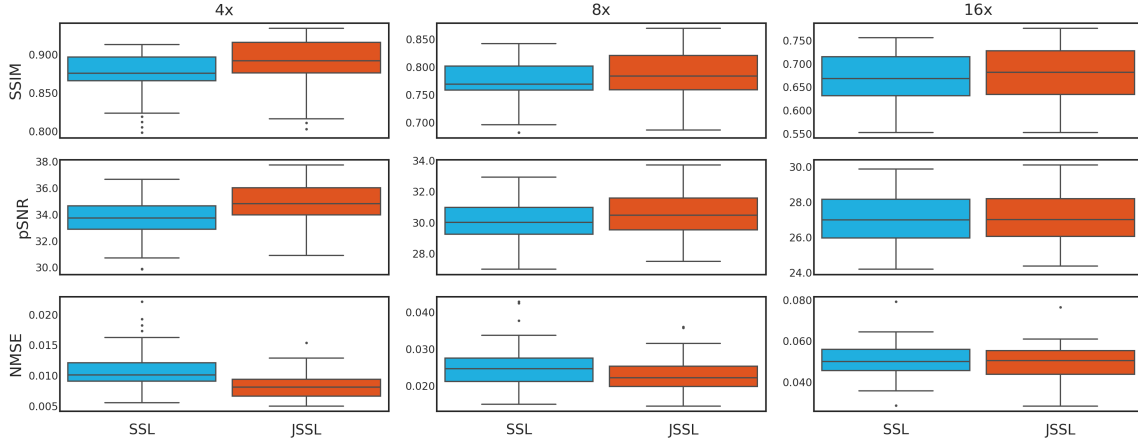


Figure S3: Evaluation metric results for SSL and JSSL methods using an E2EVarNet architecture as a reconstruction network.

- (2) JSSL and SSL using a constant partitioning ratio of  $q = 0.5$  instead of  $q \in (0.3 - 0.8)$  as in Section 3.4.
- (3) JSSL and SSL setting for the ACS window  $w \times w = 10 \times 10$  opposed to  $w \times w = 4 \times 4$  in our experiments in Section 3.4.

#### E.2.1. ALTERNATIVE CONFIGURATION EXPERIMENTS RESULTS

Table S4: Average results for alternative configurations with the fastMRI prostate as target.

Setup	2x			4x			8x			16x		
	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE	SSIM	pSNR	NMSE
SSL Original	0.956 $\pm$ 0.015	38.8 $\pm$ 2.6	0.004 $\pm$ 0.002	0.891 $\pm$ 0.030	34.7 $\pm$ 2.0	0.009 $\pm$ 0.003	0.801 $\pm$ 0.038	31.1 $\pm$ 1.5	0.020 $\pm$ 0.005	0.707 $\pm$ 0.050	28.0 $\pm$ 1.6	0.041 $\pm$ 0.008
SSL ALL Original	0.953 $\pm$ 0.016	38.6 $\pm$ 2.5	0.004 $\pm$ 0.002	0.892 $\pm$ 0.031	34.8 $\pm$ 2.0	0.009 $\pm$ 0.004	0.801 $\pm$ 0.041	31.1 $\pm$ 1.6	0.020 $\pm$ 0.006	0.699 $\pm$ 0.052	27.8 $\pm$ 1.6	0.043 $\pm$ 0.010
JSSL Original	<b>0.965<math>\pm</math>0.015</b>	<b>39.5<math>\pm</math>2.8</b>	<b>0.003<math>\pm</math>0.002</b>	<b>0.918<math>\pm</math>0.026</b>	<b>36.4<math>\pm</math>1.9</b>	<b>0.006<math>\pm</math>0.002</b>	<b>0.842<math>\pm</math>0.038</b>	<b>32.5<math>\pm</math>1.6</b>	<b>0.015<math>\pm</math>0.004</b>	<b>0.752<math>\pm</math>0.053</b>	<b>29.3<math>\pm</math>1.6</b>	<b>0.030<math>\pm</math>0.007</b>
SSL ALL Oversamp.	0.955 $\pm$ 0.015	38.5 $\pm$ 2.5	0.004 $\pm$ 0.002	0.891 $\pm$ 0.030	34.6 $\pm$ 2.0	0.009 $\pm$ 0.004	0.807 $\pm$ 0.039	31.3 $\pm$ 1.6	0.019 $\pm$ 0.006	0.712 $\pm$ 0.051	28.1 $\pm$ 1.6	0.040 $\pm$ 0.009
JSSL Oversamp.	<b>0.968<math>\pm</math>0.012</b>	<b>41.0<math>\pm</math>2.1</b>	<b>0.002<math>\pm</math>0.001</b>	<b>0.919<math>\pm</math>0.026</b>	<b>36.7<math>\pm</math>1.9</b>	<b>0.006<math>\pm</math>0.002</b>	<b>0.842<math>\pm</math>0.038</b>	<b>32.6<math>\pm</math>1.6</b>	<b>0.014<math>\pm</math>0.004</b>	<b>0.749<math>\pm</math>0.052</b>	<b>29.2<math>\pm</math>1.6</b>	<b>0.031<math>\pm</math>0.008</b>
SSL (q=0.5)	0.957 $\pm$ 0.014	39.1 $\pm$ 2.4	0.003 $\pm$ 0.002	0.895 $\pm$ 0.027	35.0 $\pm$ 1.9	0.008 $\pm$ 0.003	0.817 $\pm$ 0.038	31.7 $\pm$ 1.6	0.017 $\pm$ 0.005	0.733 $\pm$ 0.050	28.9 $\pm$ 1.6	0.033 $\pm$ 0.008
SSL ALL (q=0.5)	0.956 $\pm$ 0.015	39.0 $\pm$ 2.5	0.004 $\pm$ 0.002	0.897 $\pm$ 0.029	35.2 $\pm$ 2.0	0.008 $\pm$ 0.003	0.816 $\pm$ 0.039	31.7 $\pm$ 1.6	0.018 $\pm$ 0.005	0.715 $\pm$ 0.052	28.3 $\pm$ 1.6	0.038 $\pm$ 0.009
JSSL (q=0.5)	<b>0.965<math>\pm</math>0.015</b>	<b>39.7<math>\pm</math>2.8</b>	<b>0.003<math>\pm</math>0.002</b>	<b>0.919<math>\pm</math>0.026</b>	<b>36.6<math>\pm</math>1.8</b>	<b>0.006<math>\pm</math>0.002</b>	<b>0.842<math>\pm</math>0.037</b>	<b>32.6<math>\pm</math>1.6</b>	<b>0.014<math>\pm</math>0.004</b>	<b>0.742<math>\pm</math>0.052</b>	<b>29.0<math>\pm</math>1.6</b>	<b>0.032<math>\pm</math>0.007</b>
SSL (w=10)	0.954 $\pm$ 0.015	38.4 $\pm$ 2.5	0.004 $\pm$ 0.002	0.893 $\pm$ 0.029	34.7 $\pm$ 2.0	0.009 $\pm$ 0.004	0.815 $\pm$ 0.038	31.6 $\pm$ 1.6	0.018 $\pm$ 0.006	0.726 $\pm$ 0.048	28.5 $\pm$ 1.6	0.036 $\pm$ 0.008
SSL ALL (w=10)	0.954 $\pm$ 0.015	38.4 $\pm$ 2.5	0.004 $\pm$ 0.002	0.890 $\pm$ 0.029	34.6 $\pm$ 2.0	0.009 $\pm$ 0.004	0.805 $\pm$ 0.039	31.2 $\pm$ 1.6	0.020 $\pm$ 0.006	0.710 $\pm$ 0.052	28.1 $\pm$ 1.6	0.040 $\pm$ 0.009
JSSL (w=10)	<b>0.958<math>\pm</math>0.016</b>	<b>38.7<math>\pm</math>2.6</b>	<b>0.004<math>\pm</math>0.002</b>	<b>0.916<math>\pm</math>0.026</b>	<b>36.4<math>\pm</math>1.8</b>	<b>0.006<math>\pm</math>0.002</b>	<b>0.839<math>\pm</math>0.038</b>	<b>32.5<math>\pm</math>1.6</b>	<b>0.015<math>\pm</math>0.004</b>	<b>0.748<math>\pm</math>0.052</b>	<b>29.2<math>\pm</math>1.6</b>	<b>0.031<math>\pm</math>0.008</b>

Summarized in Table S4, we calculated the average evaluation metrics on the test set for our alternative configurations experiments, providing additional context to the JSSL approach. These experiments consistently showcased the superior performance of JSSL over SSL setups, in line with our prior observations. Interestingly, variations in the training hyper-parameters for JSSL, such as oversampling, partitioning ratio ( $q = 0.5$ ), and

ACS window size ( $w = 10$ ), did not yield significant improvements or deteriorations in performance, except for an observable improvement in average pSNR at  $R = 2$ .

Regarding the SSL setups, an observable enhancement was witnessed for  $8\times$  and  $16\times$  accelerated data when adopting a fixed partitioning ratio  $q = 0.5$  or a larger ACS window of  $w^2 = 10^2$  pixels. However, this improvement was particularly evident in the SSL setup using solely the (subsampled) proxy dataset. Furthermore, the inclusion of proxy datasets within SSL configurations (SSL ALL) did not yield improvements in reconstruction performance, consistent with our earlier findings in the comparative study.

For further assessment, we provide in Appendix F box plots illustrating comprehensively the performance metrics, as well as sample reconstructions for each setup considered in the alternative configurations study.



## Appendix F. Additional Figures

### F.1. Comparison Studies

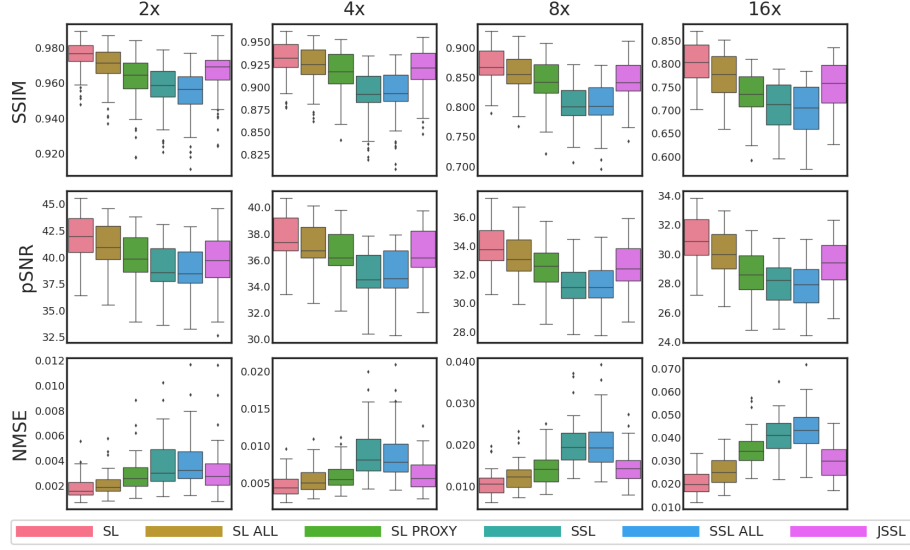


Figure S4: Evaluation results for different training setups with the prostate as target dataset, and the brain and knee datasets as proxy datasets (experiment set **A**).

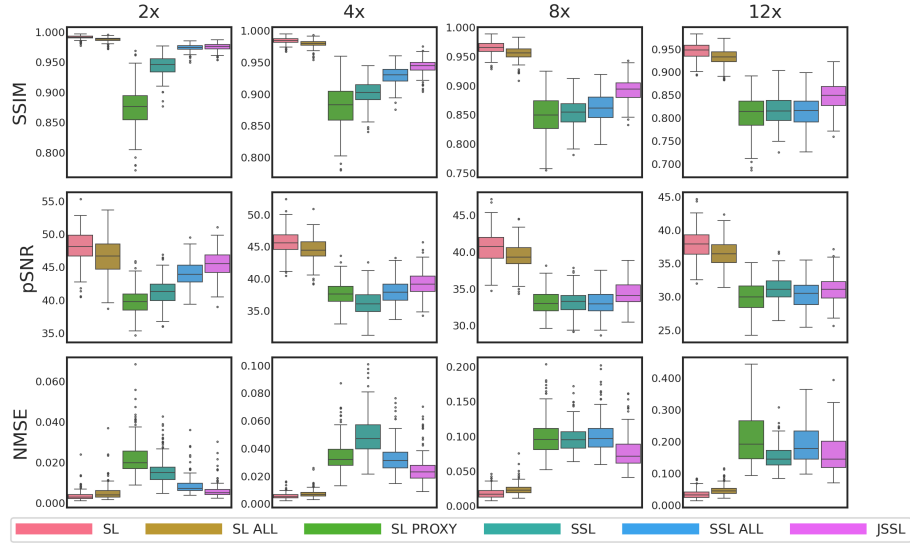


Figure S5: Evaluation results for different training setups with the cardiac as target dataset, and the brain, knee and prostate datasets as proxy datasets (experiment set **B**).

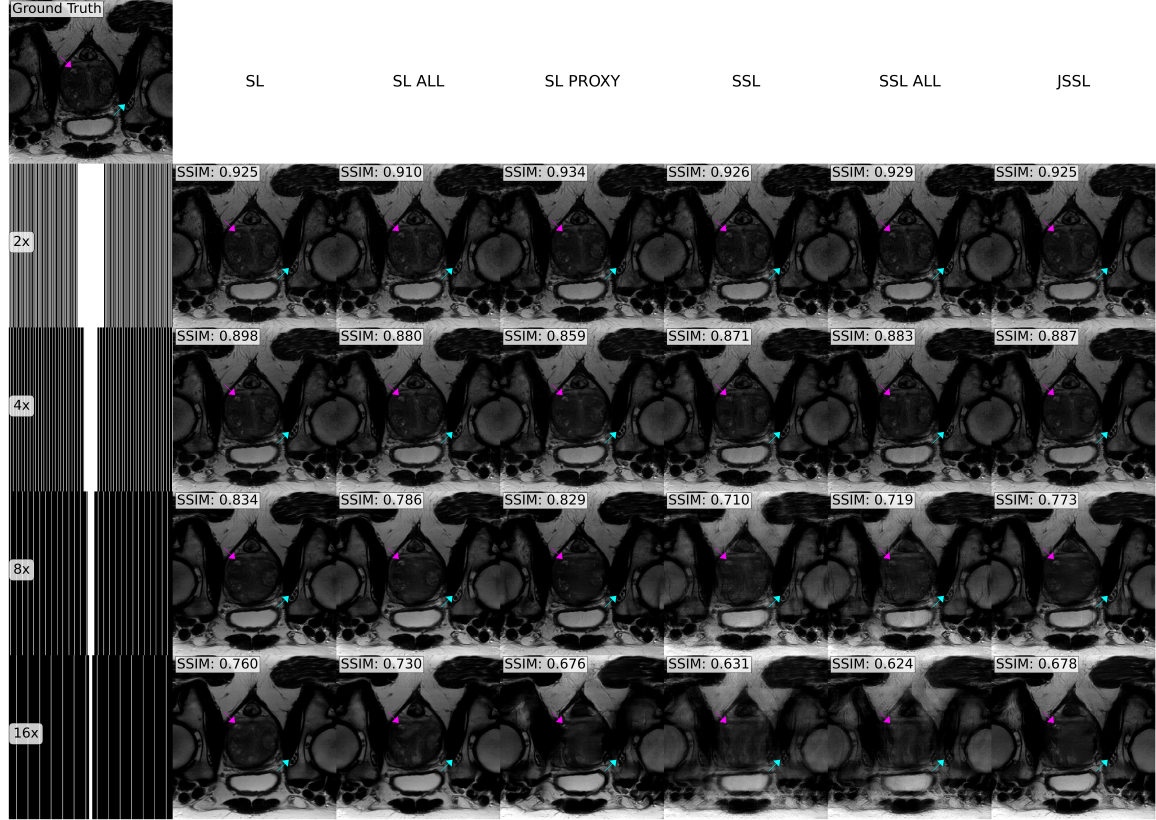


Figure S6: Example reconstructions of a prostate slice subsampled at different acceleration factors (left-most column) from the test set (experiment set **A**) from each training setup in the comparison studies (Section 3.6 of the main paper) visualized against the ground truth. Arrows point to regions of interest.

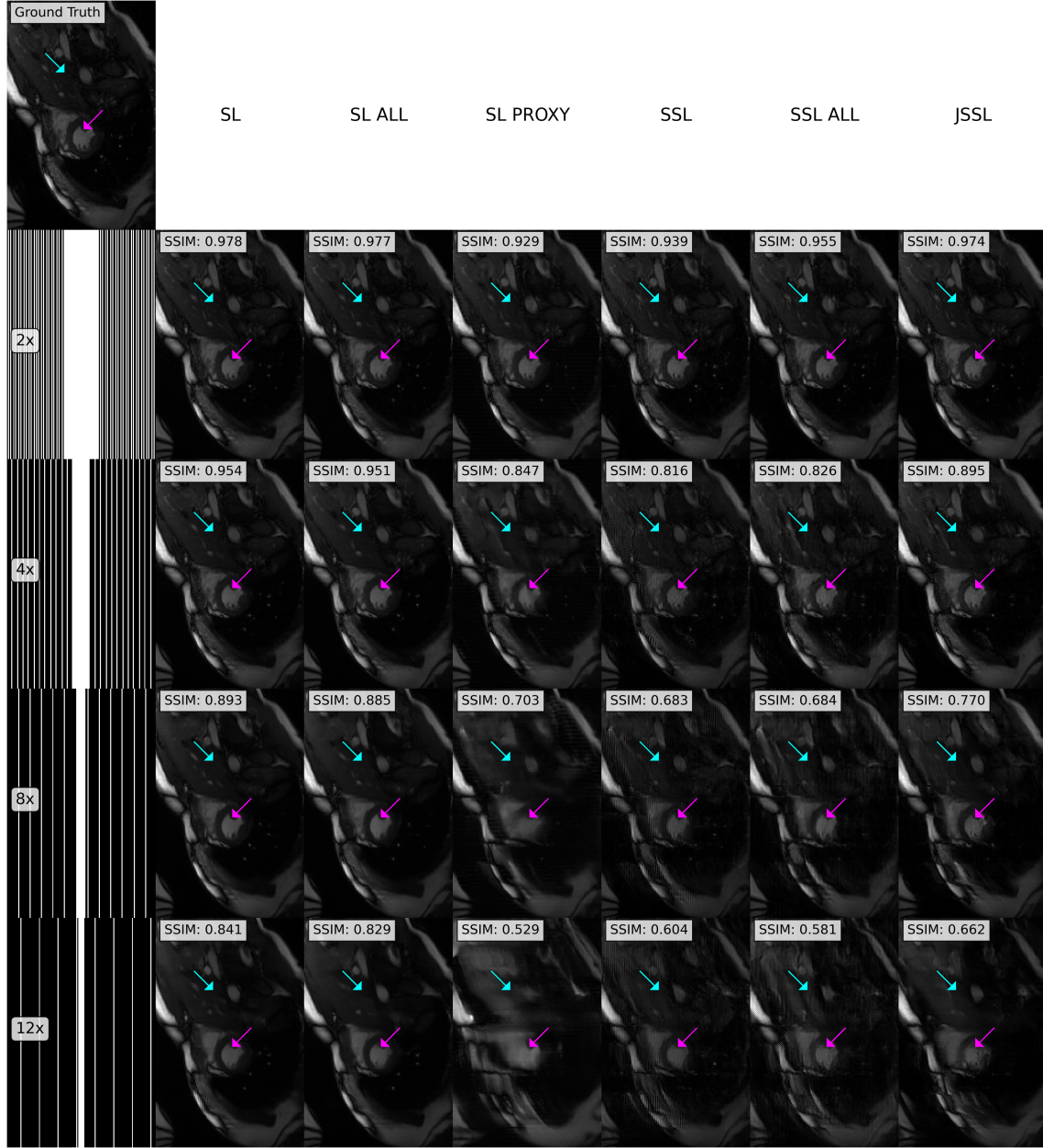


Figure S7: Example reconstructions of a cardiac slice subsampled at different acceleration factors (left-most column) from the test set (experiment set **B**) from each training setup in the comparison studies (Section 3.6 of the main paper) visualized against the ground truth. Arrows point to regions of interest.

## F.2. Ablation Studies

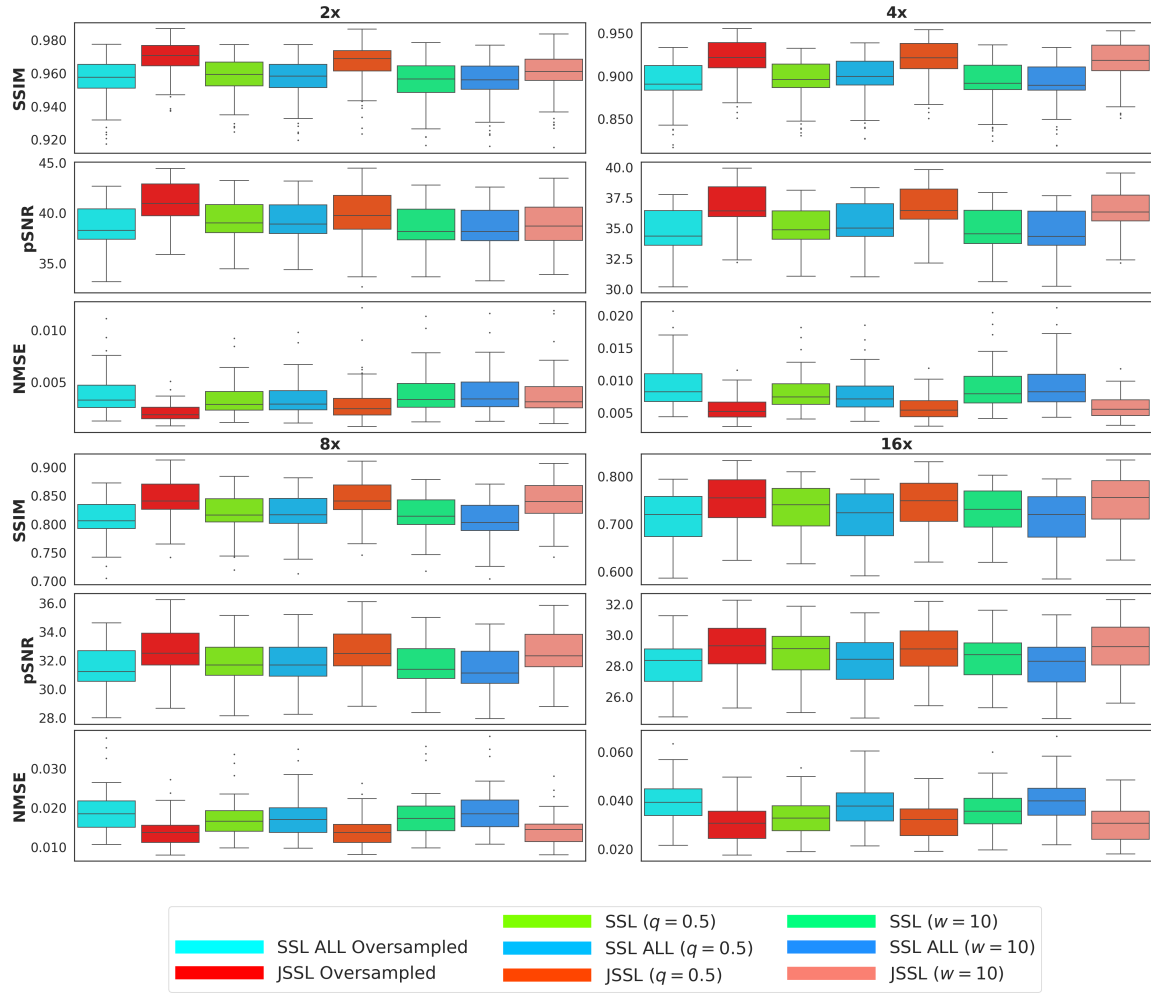


Figure S8: Evaluation results for varying JSSL and SSL setups for the alternative configurations studies of Appendix E.1.

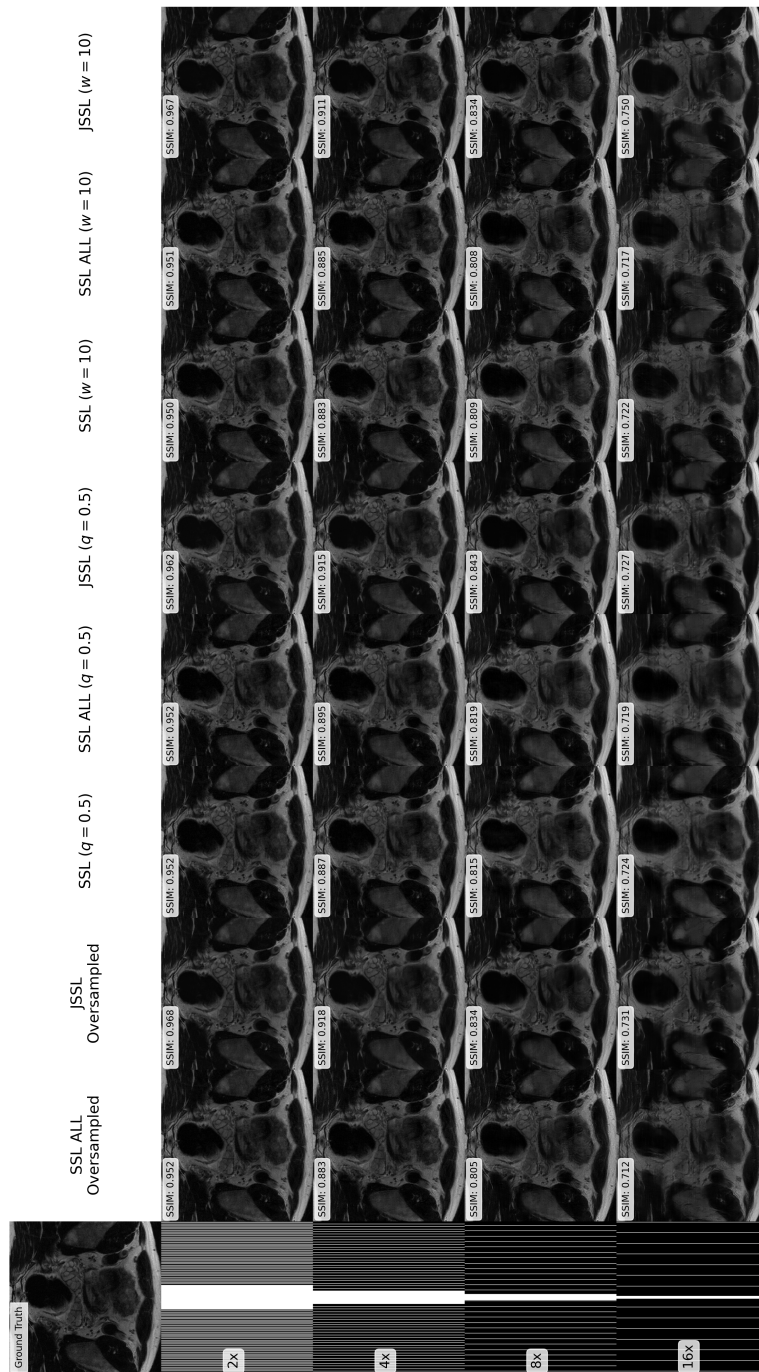


Figure S9: Example 1 of reconstructions of a prostate slice subsampled at different acceleration factors (left-most column) from the test set from each training setup in the alternative configurations studies (Appendix E.1) visualized against the ground truth. Arrows point to regions of interest.



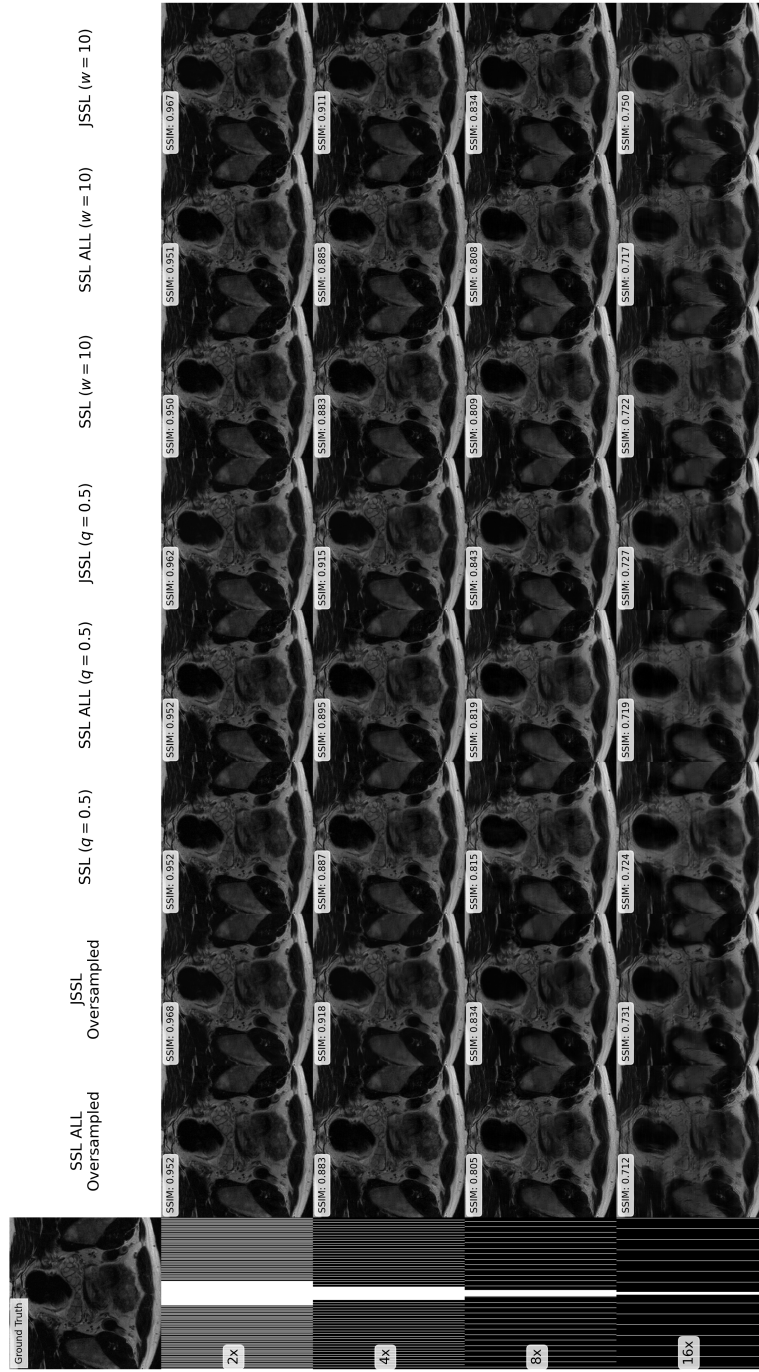


Figure S10: Example 2 of reconstructions of a prostate slice subsampled at different acceleration factors (left-most column) from the test set from each training setup in the alternative configurations studies (Appendix E.1) visualized against the ground truth. Arrows point to regions of interest.

## Appendix G. Extended Discussion

### G.1. Limitations

While our experiments indicate that JSSL demonstrates improvements over conventional SSL methods, several limitations warrant discussion. Firstly, the efficacy of JSSL is highly dependent on the availability and quality of proxy datasets. Although datasets such as the fastMRI datasets contain fully-sampled data and are readily available, there might be instances where such datasets cannot be used. This could occur in cases where the anatomical regions of interest in the proxy datasets are not sufficiently similar to those in the target dataset, or where differences in imaging protocols and acquisition parameters introduce significant discrepancies.

For instance, in experiment set **A**, where the fastMRI prostate data served as the target domain and brain and knee fastMRI datasets were used as proxies, the SL PROXY setup showed relatively good performance, indicating that training with similar proxy domains can still be beneficial for out-of-distribution inference. However, in experiment set **B**, where the CMRxRecon cardiac data was the target and brain, knee, and prostate fastMRI datasets served as proxies, the performance of SL PROXY was significantly lower than all methods, highlighting that when proxies are dissimilar to the target, SL PROXY struggles to generalize effectively. In both scenarios, JSSL consistently surpassed SL PROXY, indicating that the combined supervised and self-supervised approach is more robust, regardless of the proxy dataset’s similarity.

Additionally, the inclusion of proxy datasets in training can introduce biases, particularly if there are substantial differences between the proxy and target domains. This bias could potentially degrade the model’s performance on the target dataset, as observed in some of our supervised learning experiments.

Moreover, similar to any DL-based method, JSSL’s performance is influenced by the choice of loss functions for each component of the JSSL loss and their weighting in the loss  $\mathcal{L}_{\psi}^{\text{JSSL}}$ . In our experiments, we employed identical dual-domain loss functions for each component and equal weighting for the SL and SSL components (see Appendix Section D). However, different loss and weighting choices might affect JSSL’s performance.

JSSL performance also depends on the partitioning strategy used for subsampled data in self-supervised learning. While we adopted a Gaussian partitioning scheme, alternative strategies might yield different results and require further exploration. The optimal partitioning scheme may vary depending on the specific characteristics of the target and proxy datasets, as well as the desired reconstruction quality.

Lastly, our experiments are limited to comparing only one SSL method (SSDU) and does not consider other proposed self-supervised methodologies. However, the reason for comparing to SSDU only is that we consider it representative, as most SSL-based methods are derivatives of SSDU and still employ SSL-based losses to train their models (refer to Appendix B). In addition, comparing to methods that train more than one model as their SSL task is outside the scope of this research, as this can introduce additional computational difficulties and are derivatives of the SSDU method. Our purpose is to compare JSSL and SSL training methods in their general forms.