SL_0-MSNet: Multi-scale Unfolding Network with Smooth l_0 Regularization for CS-MRI

YingYing Li
Department of Computer, Faculty of Computer Science and Technology, University of Qingdao, Qingdao 266071, China

* Corresponding Author: **YingYing Li**

Article Info

P-ISSN: 3051-3383
E-ISSN: 3051-3391
Volume: 07
Issue: 01
Received: 12-11-2025
Accepted: 14-12-2025
Published: 16-01-2026
Page No: 13-23

Abstract
Compressed Sensing Magnetic Resonance Imaging (CS-MRI) accelerates MRI acquisition by reconstructing high-quality images from sparsely sampled k-space data. Current deep unfolding reconstruction networks usually rely on l_1-norm relaxation, suffering from insufficient sparsity and loss of high-frequency details. And traditional single-scale feature extraction fails to fully capture the detailed features of the MRI data. In this paper, we propose a multi-scale reconstruction network (SL_0-MSNet) unfolded from the Alternating Direction Method of Multipliers (ADMM) framework to achieve accelerated MRI reconstruction. By incorporating l_0-norm regularization and designing an adaptive smooth thresholding function to address the non-differentiability challenge in l_0-norm optimization. Meanwhile, a Multi-Scale Feature Extraction and Fusion (MSFAF) module is designed, which captures local textures and global anatomical features synergistically through a cascaded structure of parallel convolutional layers with different receptive fields. Comparative experiments on standard Brain and Knee datasets show that our method significantly outperforms several mainstream CS-MRI algorithms in reconstruction quality, particularly in complex scenarios, demonstrating high clinical application potential.

DOI: https://doi.org/10.54660/IJAIET.2026.7.1.13-23

Keywords: MRI Reconstruction, Deep Learning, ADMM, Hard-Thresholding Function, Multi-Scale Feature Fusion

1. Introduction

Compressed Sensing Magnetic Resonance Imaging (CS-MRI) accelerates image reconstruction through sparse k-space sampling technology, addressing the core challenges of traditional MRI such as long scanning time, low imaging efficiency, and high hardware costs. Currently, deep unfolding networks (DUNs) have become the mainstream of research due to their advantages of integrating model-driven and data-driven approaches. Their core idea is to unfold the iterative steps of traditional optimization algorithms into neural network layers, optimizing hyperparameters through end-to-end training. It started with the work of Gregor and LeCun[Error! Reference source not found.] where the authors showed that one could unfold the ISTA to create a feed-forward network. Then, one can train ISTA in an end-to-end fashion to determine the parameters in ISTA so that they are best suitable for the training data. The work of ADMM-Net proposed by Yang *et al.*[Error! Reference source not found.] was the first to suggest the potential benefit of designing deep neural networks for inverse problems by unfolding optimization ADMM algorithms[Error! Reference source not found.]. Zhang *et al.*[Error! Reference source not found.] first proposed unfolding the iterative steps of ISTA[Error! Reference source not found.] into neural network layers, resulting in ISTA-Net, each layer of the network performs operations similar to those of the ISTA algorithm.

The mathematical foundation of Compressive Sensing (CS) is formalized as:

$$y=Ax \tag{1}$$

where measurement matrix $A = PF$ (P is the under-sampling operator, and F is the Fourier transform matrix), the core of CS-MRI is to infer the original signal $x \in R^N$ from the sparsely sampled k-space measurement values $y \in R^M$ ($M \ll N$). The

unconstrained optimization model for standard CS-MRI is:

x^= 〖arg (min)┬x〗 ⁡{1/2 〖||Ax-y||〗 _2^2+λ 〖||Φx||〗 _0 } (2)

where $\frac{1}{2}||Ax-y||_2^2$ is the data fidelity term, $||\Phi x||_0$ is the l_0 regularization term (Φ is sparse transform operator[Error! Reference source not found.]), and λ

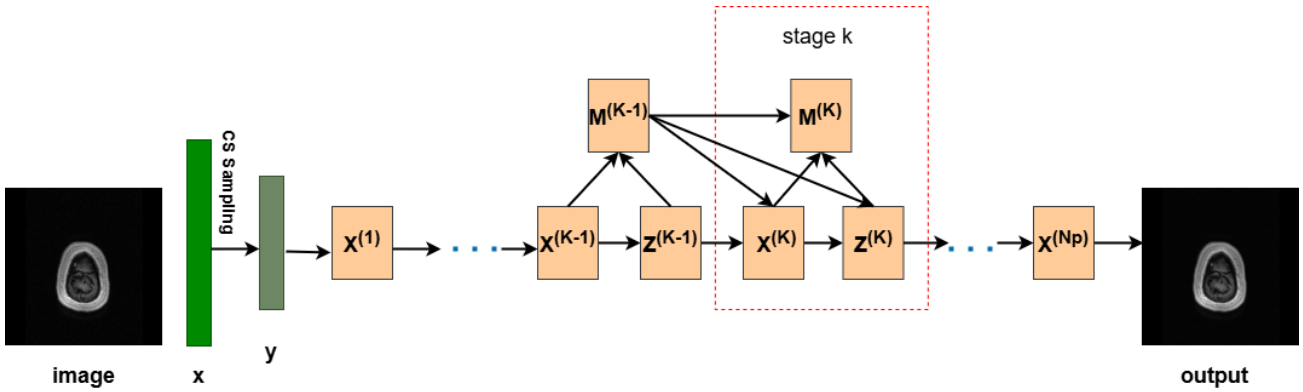


Fig 1: The overall data flow diagram of our network. There are three types of graph nodes at the k-th stage, which respectively correspond to the reconstruction layer (X), the auxiliary variable update layer (Z), and the multiplier update layer (M)

Balances the weights of the two terms. However, due to the NP-hard nature[Error! Reference source not found.] of l_0 regularization. it is difficult to optimize directly. Early studies used the l_1-norm as a substitute, leading to the derivation of several classic deep unfolding networks. ISTA-Net[Error! Reference source not found.], based on ISTA, approximated the sparsity constraint through the soft-thresholding function of the l_1-norm to achieve efficient reconstruction of k-space data. ADMM-Net[Error! Reference source not found.] employed the proximal operator of l_1 regularization to handle sparsity in the image domain, balancing reconstruction speed and stability. FISTA-Net[Error! Reference source not found.] introduced Nesterov momentum acceleration on the basis of ISTA-Net, achieving 8 × acceleration sampling through l_1 soft-thresholding. PD-Net[Error! Reference source not found.] relied on the primal-dual algorithm to jointly optimize TV-l_1 constraints in the image domain and data consistency in the k-space. Although the l_1-norm is widely used, it struggles to achieve the strict sparsity of the l_0-norm. Studies have shown that the l_0-norm can be solved using hard-thresholding functions[Error! Reference source not found.], but the non-differentiability of these functions at threshold points prevents their direct application in gradient-based deep learning frameworks. Recent research has approximated hard-thresholding functions with continuous functions to optimize l_0 regularization. For example, Yang *et al.*[Error! Reference source not found.] used trigonometric functions (e.g., sine) to approximate the l_0-norm in extreme learning machines to improve network compactness. Zhang *et al.*[Error! Reference source not found.] approximated the l_0-norm and corresponding Group l_0 regularization terms with a series of smooth functions, developing gradient training methods for smooth Group l_0 regularization. Nguyen *et al.*[Error! Reference source not found.] used continuous functions of vector variables to approximate l_0 regularization. However, these methods suffer from insufficient matching between the gradient characteristics of smooth functions and the abrupt sparsity-inducing nature of the l_0-norm, leading to significant errors in sparse representation. On the other hand, the receptive field characteristics of convolutional operations are critical for comprehensive feature extraction. In real medical images, anatomical structures and lesion features exist at multiple scales. Single-scale feature extraction alone is insufficient to model long-range dependencies in images, failing to fully capture the multi-scale information and global structural features in MRI data.

Based on the above discussions, this paper proposes a novel smooth l_0 regularized multi-scale reconstruction network to improve reconstruction quality in complex scenarios. The main contributions of this paper are as follows.

1. We develop a deep network architecture based on ADMM algorithm unfolding, which introduces l_0 regularization and a hard-thresholding function to align with the sparse-inducing mechanism of the l_0-norm, we further design a sigmoid-based smooth approximation method to enable differentiable approximation of the hard-thresholding logic, preserving the l_0 sparse-inducing capability while enhancing algorithm feasibility.
2. We introduce a customized MSFAF module, which employs convolution kernels of different sizes to extract multi-scale features and introduces skip connections and attention modules to fuse features, thereby enhancing the capability to capture high-frequency details.

2. Proposed Method

2.1. Traditional ADMM for CS

ADMM efficiently solves high-dimensional linear inverse problems by decomposing complex optimization problems into subproblems. By introducing an auxiliary variable $z = x$, Eq. **Error! Reference source not found.**) is equivalent to:

$$\hat{x}, z = \underset{x,z}{argmin}\left\{\frac{1}{2}||Ax-y||_2^2 + \lambda||\Phi z||_0\right\} \quad (3)$$

now, for $\rho > 0$ (penalty parameter), iteration number k and initial points $(x^{(0)}, z^{(0)}, \beta^{(0)}) = (0,0,0)$, the optimization problem in Eq. **Error! Reference source not found.**) can be solved by the iterative scheme of ADMM:

$$x^{(k)} = F^H(P^TP + \rho I)^{-1}\left[P^Ty + \rho F\left(z^{(k-1)} - \beta^{(k-1)}\right)\right] \quad (4)$$

$$z^{(k)} = \underset{z}{argmin}\left\{\lambda||\Phi z||_0 + \frac{\rho}{2}||x^{(k)} + \beta^{(k-1)} - z||_2^2\right\} \quad (5)$$

$$\beta^{(k)} = \beta^{(k-1)} + \eta\left(x^{(k)} - z^{(k)}\right) \quad (6)$$

where ŋ denotes the update rate, I denotes the identity matrix.

2.2. ADMM-Net Framework

As shown in

Fig 1, we map the ADMM iterative process in Eq. **Error! Reference source not found.**),**Error! Reference source not found.**),**Error! Reference source not found.**) into this data flow graph[Error! Reference source not found.], the k-th iteration of the ADMM algorithm corresponds to the k-th stage in the graph, and each network stage is composed of a reconstruction block (X), an auxiliary variable update block (Z) and a multiplier update layer (M). Next, we will introduce the framework structure of the network in detail.

Reconstruction Layer $X^{(k)}$

At the k-th stage, given the outputs $z^{(k-1)}$, $\beta^{(k-1)}$ from the previous layer, according to the calculation of Eq. **Error! Reference source not found.**), we can obtain the reconstructed value $x^{(k)}$,which is then used as the input for the subsequent auxiliary variable update block and the multiplier update layer. In the first stage (that is, $k = 1$), $x^{(1)} = F^H(P^TP + \rho I)^{-1}(P^Ty)$.

Auxiliary Variable Update Block $Z^{(k)}$

Eq. **Error! Reference source not found.**) is actually a special case of the proximal mapping, that is:

$$prox_{\frac{\lambda}{\rho}R}(x + \beta) = \underset{z}{argmin}\left\{\frac{1}{2}||(x + \beta) - z||_2^2 + \lambda||\Phi z||_0\right\} \quad (7)$$

The solution to this problem faces two key challenges: the non-differentiable nature of the l_0-norm makes gradient based optimization algorithms inapplicable directly, and additionally, the proximal mapping has a closed-form solution only when the transform matrix Φ satisfies orthogonality. To address this, we carry out the following work.

Smooth Hard-Threshold Function Based on the Sigmoid Function

The mathematical expression of the hard-thresholding function and its corresponding derivative are as follows:

$$HT_\theta(x) = \begin{cases} x & x > \theta \\ 0 & |x| \le \theta \\ x & x < -\theta \end{cases} \quad \frac{dHT_\theta(x)}{dx} = \begin{cases} 1 & x > \theta \\ 0 & |x| \le \theta \\ 1 & x < -\theta \end{cases} \quad (8)$$

where θ is the threshold. To construct a continuously differentiable smooth approximation function, we first introduce the Heaviside step function, defined as:

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \ge 0 \end{cases} \quad (9)$$

here, we select the sigmoid function[Error! Reference source not found.] as the smoothing tool and define its smooth approximation function $H_\varepsilon(x)$ as:

$$H_\varepsilon(x) = \frac{e^{\frac{x}{\varepsilon}}}{1 + e^{\frac{x}{\varepsilon}}} \quad (10)$$

where $\varepsilon = 0.2$ is a smoothness parameter determined through extensive experimental tuning, controlling the approximation degree of $H_\varepsilon(x)$ to the step function (as shown in Fig 2 and Fig 3). Based on the piecewise interval characteristics of the hard-thresholding function, we design three segmental characteristic functions:

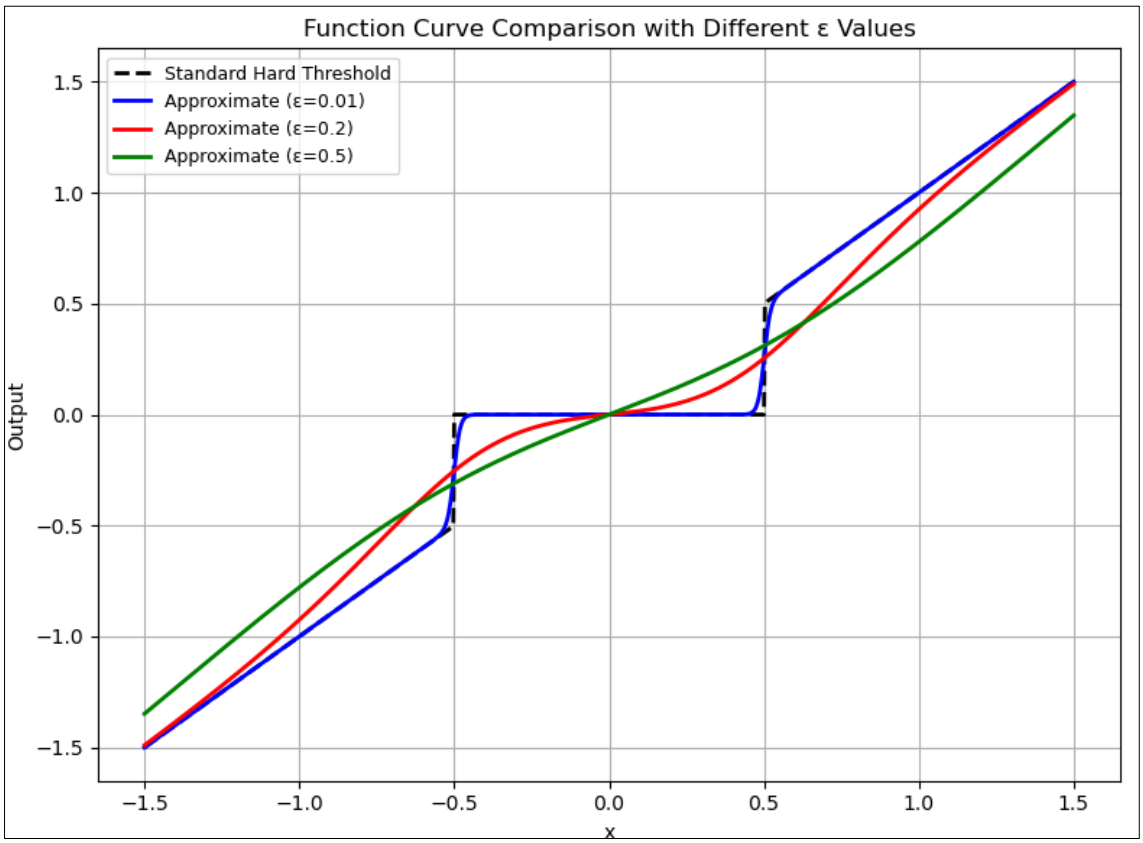


Fig 2: The curve shapes of approximate functions and the standard hard-thresholding function

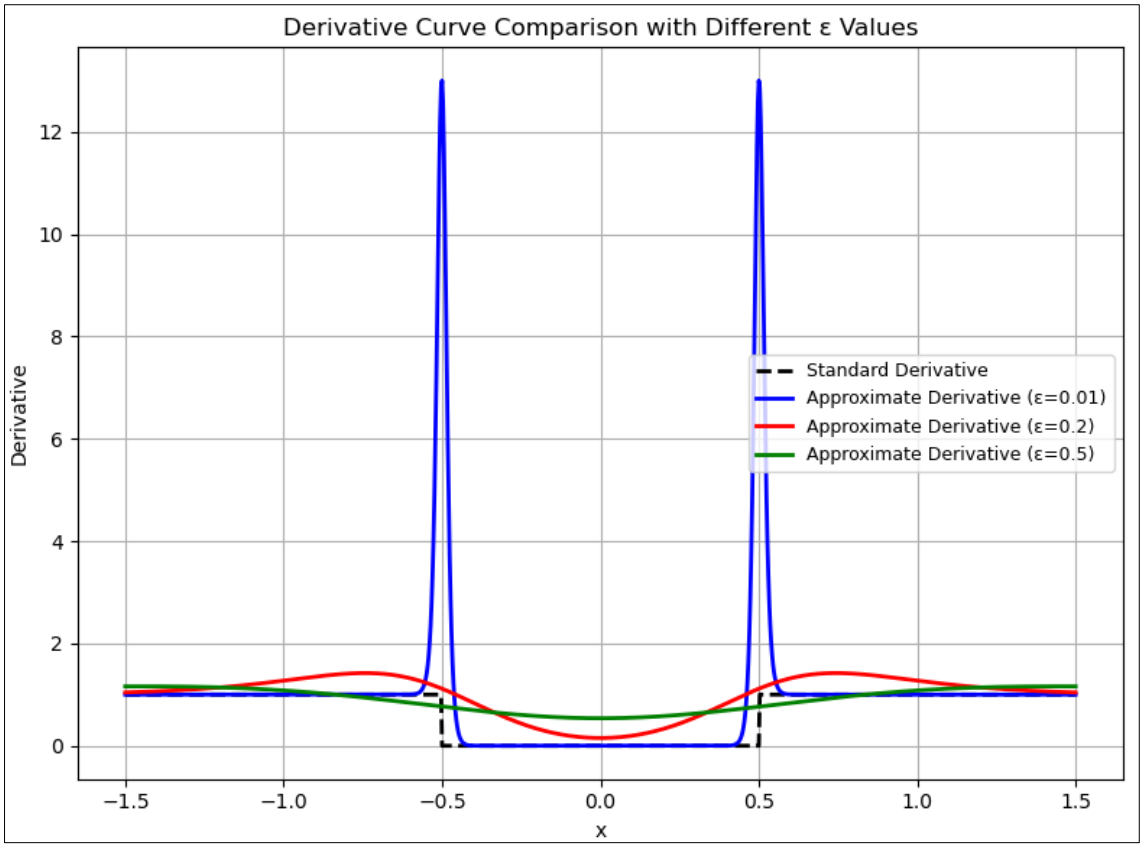


Fig 3: The derivative curves of approximate functions and the standard hard-thresholding function

Right Zone($x > \theta$): The characteristic function is designed as $\chi_1(x) = H(x-\theta)$. When $x > \theta$, and $H(x-\theta)$ approaches 1. As x increases, $H(x-\theta)$ remains constant at 1, consistent with the characteristic that the hard-thresholding function is constantly x in the right zone.

Middle Zone($|x| \le \theta$): The characteristic function is $\chi_2(x) = H(x+\theta) - H(x-\theta)$. Since $x \ge -\theta$, and $H(x+\theta)$ approaches 1, since $x \le \theta$, and $H(x-\theta)$ approaches 0, so $\chi_2(x)$ approaches 0, matching the characteristic that the hard-thresholding function is constantly 0 in the middle zone.

Left Zone($x < -\theta$): The characteristic function is $\chi_3(x) = 1 - H(x+\theta)$. When $x < -\theta$, and $H(x+\theta)$ approaches 0. As x decreases, $H(x+\theta)$ remains 0, so $\chi_3(x)$ approaches 1, consistent with the characteristic that the hard-thresholding function is constantly x in the left zone.

Based on the above three segmental characteristic functions, the approximate function $f_\theta(x)$ is obtained:

$$f_\theta(x) = x \cdot \chi_1(x) + 0 \cdot \chi_2(x) + x \cdot \chi_3(x) = x + x\big(H(x-\theta) - H(x+\theta)\big) \quad (11)$$

and the corresponding smooth approximation function $f_{\theta,\epsilon}(x)$ (as shown in Fig 2) is given by:

$$f_{\theta,\epsilon}(x) = x + x\big(H_\varepsilon(x-\theta) - H_\varepsilon(x+\theta)\big) \quad (12)$$

Design of Φ

In order to more comprehensively explore the complex features in the data, we use a non-linear transformation function F(·) to replace Φ (as shown in Fig 4), whose parameters are trainable. Then Eq. **Error! Reference source not found.**) becomes:

$$z^{(k)} = \underset{z}{argmin}\left\{\lambda||F(z)||_0 + \frac{\rho}{2}||x^{(k)} + \beta^{(k-1)} - z||_2^2\right\} \quad (13)$$

here, inspired by the powerful representation power of CNN[Error! Reference source not found.] and its universal approximation property[Error! Reference source not found.], we design F(·) as a structure composed of "convolutional layer (conv) + Rectified Linear Unit (ReLU) + convolutional layer (conv) + Multi-Head Attention Block (MAB)" (The default convolution kernel is 3×3, and the number of filters is $N_f = 33$). F(·) can be equivalently formulated in matrix form as

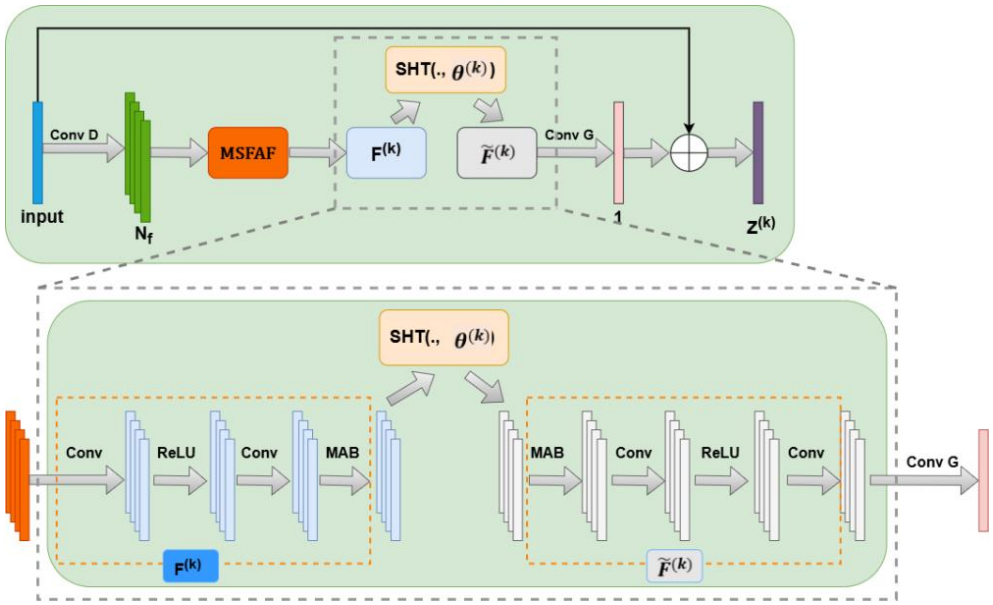


Fig 4: A schematic illustration of the framework of the auxiliary variable update block ($Z^{(k)}$) in each iterative stage of our network.

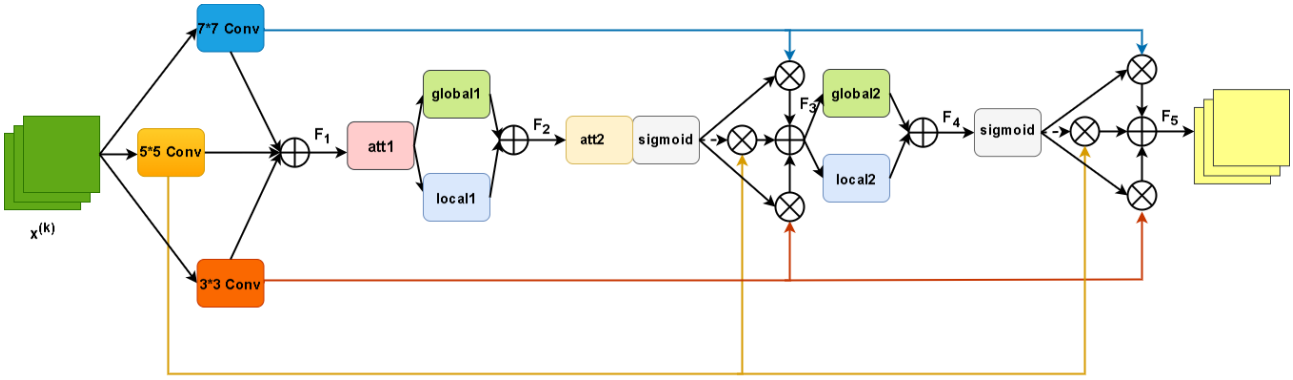


Fig 5: The Multi-Scale Feature Extraction and Fusion (MSFAF) module in our auxiliary variable update block is used to extract richer features

$F(\cdot) = MAB(BReLU(Ax))$, where A and B correspond to the front and rear convolution operators, respectively, and the

MAB[Error! Reference source not found.] is composed of Multi-Scale Large Kernel Attention (MLKA) and Gated Spatial Attention Unit (GSAU). By combining the multi-scale mechanism and large kernel attention, it can effectively capture the long-range dependencies at different granularities. Specifically, F(·) first
extracts local features layer by layer through two convolutional layers and ReLU activation functions to enhance non-linearity, providing highly discriminative inputs for subsequent attention modules. Then, by cascading MAB modules, it achieves collaborative optimization of local features and global dependencies. Drawing inspiration from[Error! Reference source not found.], we also introduce the left inverse of F(·), denoted as $\tilde{F}(\cdot)$, such that F˜ ◦ F = I, where I is the identity operator, $\tilde{F}(\cdot)$ has a symmetrical structure with F(·). Joint learning is achieved by introducing the symmetry constraint $\tilde{F}\circ F = I$ into the loss function during network training. This yields the following property:

$$||F\left(x^{(k)} + \beta^{(k-1)}\right) - F(z)||_2^2 \approx \alpha||x^{(k)} + \beta^{(k-1)} - z||_2^2 \quad (14)$$

where α is a scalar and is only related to the parameters of F(·). By incorporating this linear relationship into Eq. **Error! Reference source not found.**), the following optimization is obtained:

$$z^{(k)} = \underset{z}{argmin}\left\{\frac{1}{2}\left|\left|F\left(x^{(k)} + \beta^{(k-1)}\right) - F(z)\right|\right|_2^2 + \frac{\lambda\alpha}{\rho}||F(z)||_0\right\} \quad (15)$$

The closed-form solution is finally obtained via proximal mapping:

$$z^{(k)} = \tilde{F}\left(SHT\left(F\left(x^{(k)} + \beta^{(k-1)}\right), \theta^{(k)}\right)\right) \quad (16)$$

here, SHT(·) is the smooth hard-thresholding function $f_{\theta,\varepsilon}(x)$ we constructed previously, with $\theta = \frac{\lambda\alpha}{\rho}$. Since we initialize $\beta^{(0)} = 0$, at the first stage:

$$z^{(1)} = \tilde{F}\left(SHT\left(F\left(x^{(1)}\right), \theta^{(1)}\right)\right) \quad (17)$$

Multi-Scale Feature Extraction and Fusion Module
In compressive sensing reconstruction tasks, information loss caused by under-sampling requires the model to accurately recover multi-scale features from limited data, where large-scale features provide global structures and small-scale features capture local details. To enhance the feature extraction capability of subsequent convolution operations and enable subsequent convolution operations to extract richer information, as shown in Fig 4, we introduce a MSFAF module (as shown in Fig 5) before the MRI data enters the non-linear function F(·). Its core architecture achieves deep coupling of multi-scale features and attention mechanisms through five levels of processing, with specific designs as follows.

Multi-Scale Feature Initial Fusion
Parallel 3×3, 5×5, and 7×7 convolutional kernels are employed to extract multi-scale features. These features are initially fused through element-wise summation and subsequently enhanced by an attention module. This design effectively captures both local details and global structural information of the input image.

Local-Global Feature Co-optimization
The discriminative power of local features is strengthened via 1×1convolutional operations, while a global attention mechanism is constructed through downsampling-upsampling paths to expand the receptive field. The synergistic interaction between these components enables multi-level feature selection.

Adaptive Feature Weighted Fusion
A learnable sigmoid gating mechanism is introduced to dynamically adjust fusion weights for multi-scale features based on their importance, achieving adaptive feature selection.

Iterative Feature Refinement
Secondary local-global attention mechanisms combined with dynamic weighting strategies further optimize feature representation, enhancing the model's feature composition capability.

2.3. Loss Function Design

Given the training data pairs $\left\{x_i, x_i^{(N_p)}\right\}_{i=1}^{N_b}$, our network first takes x_i as the input. It first generates the measurement value y_i through the measurement matrix $A = PF$, and then enters our ADMM unfolded network with multiple stages to generate the final reconstruction result $x_i^{(N_p)}$ as the output. While satisfying the symmetric constraint condition $\tilde{F}^k\circ F^k = I\ \forall k = 1, \ldots, N_p$, we select the Normalized Root Mean Square Error (NRMSE)[Error! Reference source not found.], which is an average-based metric, as the loss function to reduce the difference between x_i and $x_i^{(N_p)}$. Therefore, we design the following end-to-end loss function:

$$\pounds_{total}(\Theta) = \pounds_{nrmse} + \Upsilon\pounds_{constraint} \quad (18)$$

$$\pounds_{nrmse} = \frac{1}{N_b}\sum_{i=1}^{N_b}\frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(x_{ij} - x_{ij}^{(N_p)}\right)^2}}{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(x_{ij} - \bar{x}_i\right)^2}} \quad (19)$$

$$\pounds_{constraint} = \frac{1}{N_b N}\sum_{i=1}^{N_b}\sum_{k=1}^{N_p}||\tilde{F}^{(k)}\left(F^k(x_i)\right) - x_i||_2^2 \quad (20)$$

where N_b, N_p, N and Υ are the total number of training data, the number of stages of the network, the dimension (the number of elements) of the training data, and the regularization parameter, respectively. In our experiment, the value of Υ is set to 0.01, and we learn the parameters $\Theta = \{\rho^{(k)}, \theta^{(k)}, F^{(k)}, \tilde{F}^{(k)}\}_{k=1}^{N_p}$ by minimizing the loss.

3. Experimental Results

3.1. Experimental Details and Datasets

All experiments are implemented on an NVIDIA TITAN RTX using PyTorch 2.0.0. We employ the Adam optimizer[Error! Reference source not found.] with a learning rate set to 0.001, which is gradually reduced during training based on an adaptive adjustment strategy. Our model is trained for 250 epochs at each CS ratio. Two benchmark datasets are used: Brain[Error! Reference source not found.] and Knee[Error! Reference source not found.]. Among them, the Brain dataset contains 100 brain images for training and 50 brain images for testing. The Knee

dataset we used is the knee DICOM data from the FastMRI dataset collected by the NYU School of Medicine for data acquisition through a 15-channel knee coil array, we select 200 pieces of data with a size of 256×256 from multiple DCM files as training images and 50 for testing. Additionally, our model is compared with six representative CS-MRI methods, including the baseline Zero-filled method, the classic data-driven network RDN[Error! Reference source not found.], and deep unfolding methods ISTA-Net[Error! Reference source not found.], ADMM-Net[Error! Reference source not found.], PUERT[Error! Reference source not found.], and ACIU-Net[Error! Reference source not found.]. We use the pseudo-radial sampling mask as the under-sampling matrix P and set four sampling rates (10%, 20%, 30%, 40%), the reconstruction accuracy of the network is evaluated by the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM), supplemented by visual analysis.

3.2. Comparison with State-of-The-Art Methods

First, considering the Brain dataset, Table 1 shows that our network outperforms other methods in terms of PSNR and SSIM across all sampling rates, with more significant advantages at low sampling rates (10%). Specifically, compared with the second-best method, ACIU-Net, our PSNR is increased by 0.17dB, and the SSIM is increased by approximately 0.0046. As shown in
Fig **6**, it is found that the edges and contours of the images reconstructed with our network are more precise, and the detailed textures are clearer. Additionally, as shown in
Fig **7**, our method is more effective in preserving major anatomical structures in regions with low signal-to-noise ratios. This study further validates the synergistic effect of the smoothing function and dynamic multi-scale modeling.
Similarly,
Table **2** shows that our network achieves the highest reconstruction quality on the Knee dataset. At a 40% sampling rate, it outperforms ACIU-Net by 0.18dB in PSNR and approximately 0.016 in SSIM.
Fig **8** illustrates that our reconstructed images possess clearer textures and richer details, benefiting from the l_0-norm constraint and the multi-scale feature modeling's capability to capture weak signals.

3.3. Ablation Study

To verify the effectiveness of the l_0-norm regularization term and MSFAF module, we design two sets of ablation experiments on the Brain and Knee datasets for comparison with the full network.

Influence of the l_0-Norm Regularization Term

Since most DUNs use the l_1-norm regularization term, we conduct comparative ablation experiments between the l_0-norm and l_1-norm regularization terms. Table 3 shows that the network with l_0-norm consistently achieves higher PSNR and SSIM values across all sampling rates.
Fig **9** reveals that images reconstructed with the l_1-norm exhibit jagged artifacts and blurriness in edge and texture regions, whereas
the l_0-norm, through its smooth approximation function, enables more natural transitions and precise capture of sparse structures. Both quantitative metrics and qualitative analysis confirm that l_0-norm improves reconstruction efficiency and detail fidelity in sparse modeling.

Influence of the MSFAF Module

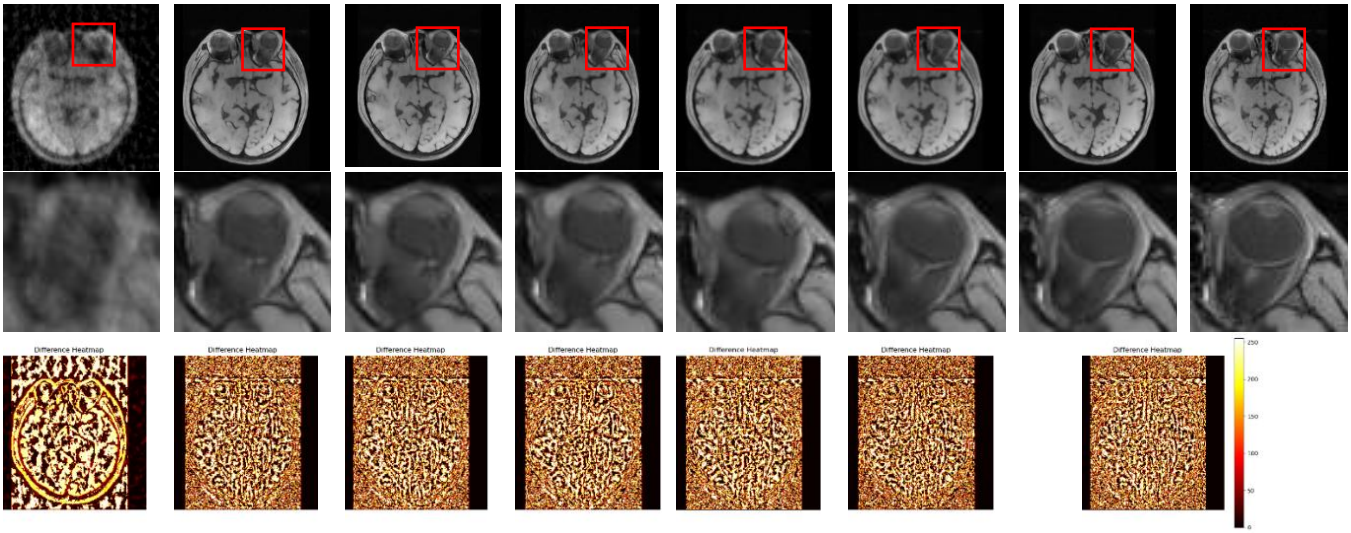
We compare the reconstruction performance of networks with and without the MSFAF module (denoted as MSFAF and no-MSFAF, respectively). Table 4 shows that the MSFAF variant achieves higher average PSNR and SSIM values across different sampling rates on both datasets. At a 20% sampling rate,
Fig **10** illustrates that the no-MSFAF network produces blurred and distorted images, while the MSFAF module, through multi-scale feature fusion, achieves more accurate boundary localization and detail reconstruction in texture-rich regions, enhancing the network's adaptability to complex structures and improving image clarity and realism.

3.4. Data-Efficient Reconstruction

To validate the model's advantage in data utilization efficiency, we randomly select 500 samples from 800 augmented Brain datasets as new training samples and compare them with ACIU-Net. As shown in Table 5, our model achieves reconstruction performance metrics comparable to ACIU-Net (trained on 800 samples) using only 550 training samples, while ACIU-Net trained on 500 samples shows significantly reduced reconstruction accuracy. As shown in
Fig **11**, the images reconstructed by ACIU-Net (550) exhibit significantly blurrier textures and severe detail loss. Therefore, our model achieves a dual

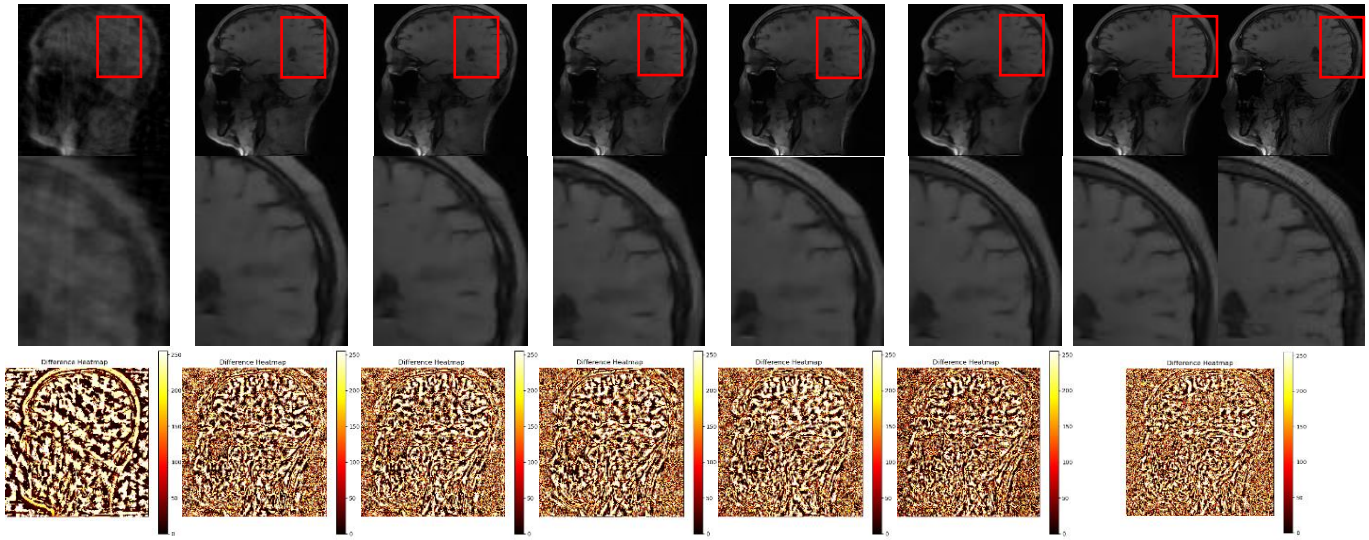
Table 1: Comparison of the evaluation metrics of Sℓ0-MSNet with Zero-filled, RDN, ISTA-Net, ADMM-Net, PUERT, and ACIU-Net on Brain dataset

Methods	Brain Dataset			
	10%	20%	30%	40%
	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Zero-filled	26.81 0.6030	30.42 0.7230	33.01 0.8022	35.15 0.8568
RDN(AAAI 2018)	34.38 0.8998	38.47 0.9474	40.80 0.9630	42.51 0.9719
ISTA-Net+(CVPR 2018)	34.62 0.9035	38.57 0.9478	40.89 0.9630	42.63 0.9725
ADMM-Net(TPAMI 2020)	34.52 0.8999	38.54 0.9473	40.83 0.9631	42.73 0.9731
PUERT(J-STSP 2022)	34.84 0.9048	38.78 0.9495	41.02 0.9642	42.73 0.9732
ACIU-Net(TETCI 2024)	36.02 0.9193	39.42 0.9540	41.42 0.9662	43.01 0.9743
Ours	36.19 0.9239	39.47 0.9545	41.52 0.9667	43.10 0.9747



Zero-filled PSNR:22.66 | RDN PSNR:30.26 | ISTA-Net+ PSNR: 30.55 | ADMM-Net PSNR:30.03 | PUERT PSNR:30.74 | ACIU-Net PSNR:32.00 | Ours Ground truth PSNR:32.12

Fig 6: Reconstruction examples in the comparison of SL_0-MSNet with Zero-filled, RDN, ISTA-Net, ADMM-Net, PUERT, and ACIU-Net on Brain dataset

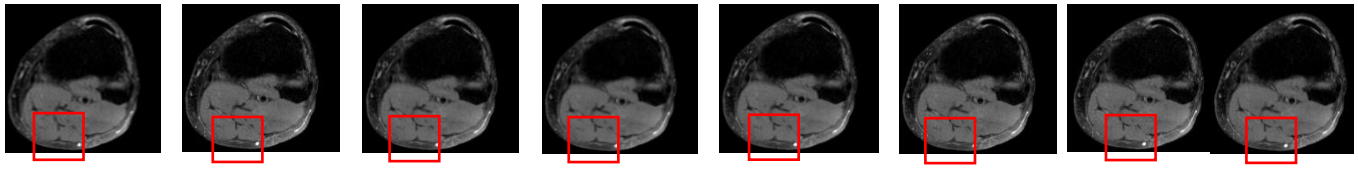


Zero-filled PSNR:28.13 | RDN PSNR:34.59 | ISTA-Net+ PSNR: 34.77 | ADMM-Net PSNR:34.23 | PUERT PSNR:34.65 | ACIU-Net PSNR:35.89 | Ours Ground truth PSNR:36.43

Fig 7: Reconstruction examples under low signal-to-noise ratio in the comparison of SL_0-MSNet with Zero-filled, RDN, ISTA-Net, ADMM-Net, PUERT, and ACIU-Net on Brain dataset

Table 2: Comparison of the evaluation metrics of SL_0-MSNet with Zero-filled, RDN, ISTA-Net, ADMM-Net, PUERT, and ACIU-Net on Knee dataset

Methods	Knee Dataset			
	10%	20%	30%	40%
	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Zero-filled	23.36 0.5425	27.81 0.6984	30.78 0.7886	33.63 0.8483
RDN(AAAI 2018)	31.01 0.8324	35.89 0.9184	38.62 0.9462	40.73 0.9524
ISTA-Net+(CVPR 2018)	31.42 0.8391	36.03 0.9204	38.76 0.9497	40.90 0.9561
ADMM-Net(TPAMI 2020)	31.56 0.8413	35.99 0.9193	38.69 0.9479	40.97 0.9573
PUERT(J-STSP 2022)	31.54 0.8407	36.08 0.9215	38.82 0.9486	41.48 0.9674
ACIU-Net(TETCI 2024)	32.18 0.8617	36.42 0.9240	39.16 0.9519	41.44 0.9670
Ours	32.36 0.8627	36.55 0.9261	39.26 0.9529	41.62 0.9686



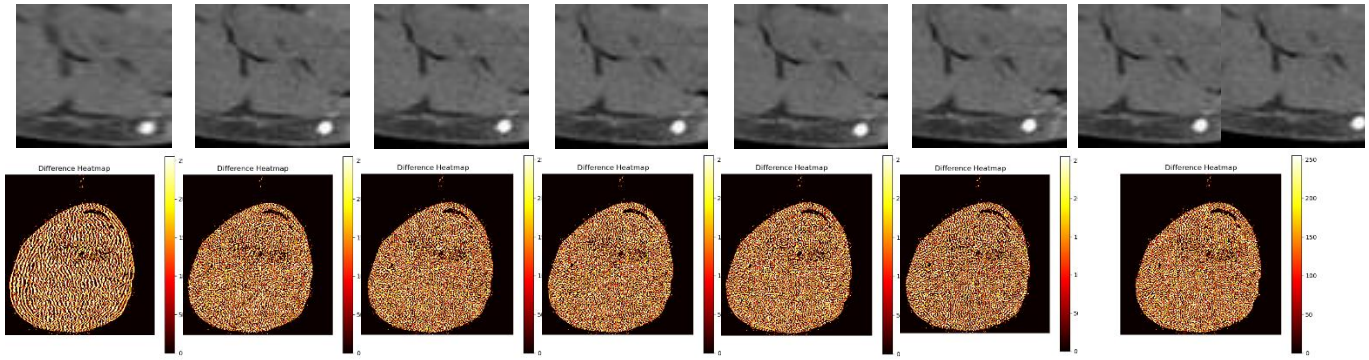


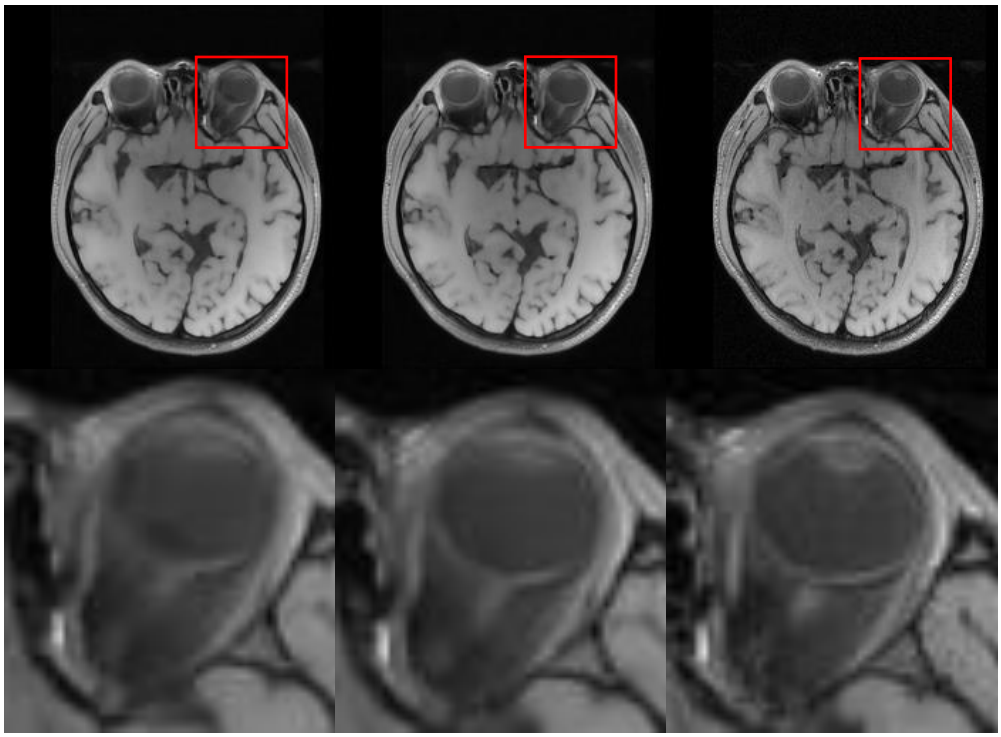
Fig 8: Reconstruction examples in the comparison of SL_0-MSNet with Zero-filled, RDN, ISTA-Net, ADMM-Net, PUERT, and ACIU-Net on Knee dataset

Table 3: Comparison of the evaluation metrics of network using l_0-norm with using l_1-norm on Brain and Knee datasets

Datasets	**Ratios**	**L_0-norm**	**L_1-norm**
		PSNR SSIM	**PSNR SSIM**
Brain	10%	36.19 0.9239	36.02 0.9218
	20%	39.47 0.9545	39.40 0.9538
	30%	41.52 0.9667	41.42 0.9663
	40%	43.10 0.9747	42.99 0.9743
Knee	10%	32.36 0.8627	32.13 0.8610
	20%	36.55 0.9261	36.42 0.9255
	30%	39.26 0.9526	39.10 0.9518
	40%	41.62 0.9686	41.49 0.9680

Table 4: Comparison of the evaluation metrics of network using MSFAF with no-MSFAF on Brain and Knee datasets

Datasets	**Ratios**	**MSFAF**	**no-MSFAF**
		PSNR SSIM	**PSNR SSIM**
Brain	10%	36.19 0.9239	35.86 0.9193
	20%	39.47 0.9545	39.27 0.9528
	30%	41.52 0.9667	41.37 0.9658
	40%	43.10 0.9747	43.01 0.9742
Knee	10%	32.36 0.8627	32.24 0.8603
	20%	36.55 0.9261	36.35 0.9235
	30%	39.26 0.9526	39.10 0.9518
	40%	41.62 0.9686	41.49 0.9680



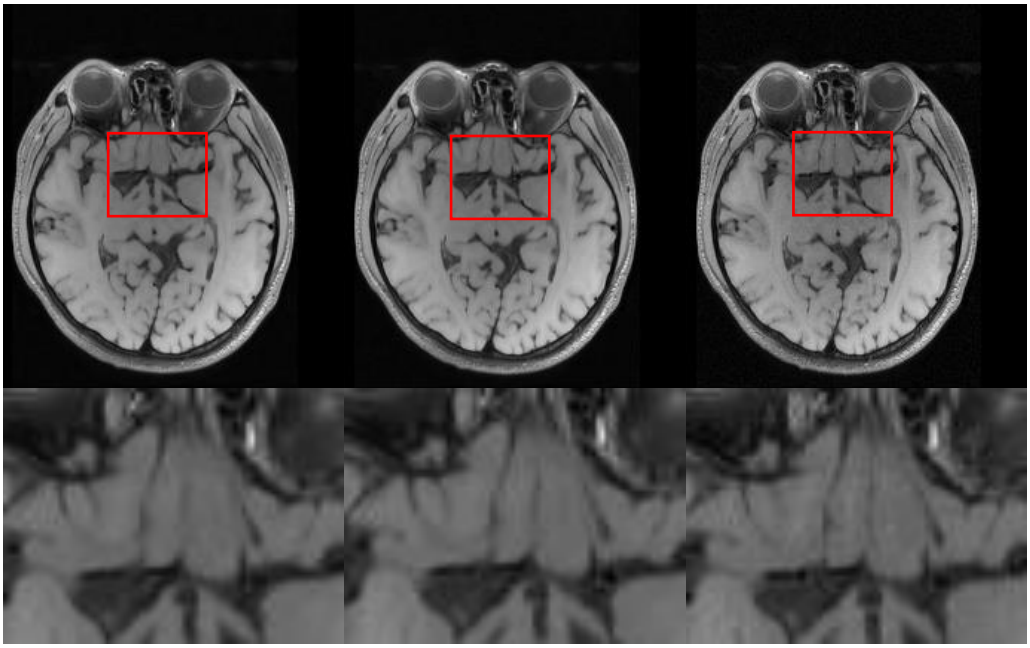


Fig 9: Reconstruction examples in the comparison of network using l_0-norm with using l_1-norm on Brain and Knee datasets

Fig 10: Reconstruction examples in the comparison of network using MSFAF with no-MSFAF on Brain and Knee datasets

Table 5: Comparison of the evaluation metrics of SL_0-MSNet with ACIU-Net on 550 Brain dataset samples

Methods	**Brain Dataset**			
	10%	**20%**	**30%**	**40%**
	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
ACIU-Net（550）	35.81 0.9192	39.25 0.9527	41.28 0.9654	42.93 0.9740
ACIU-Net（800）	36.02 0.9193	39.42 0.9540	41.42 0.9662	43.01 0.9743
Ours（550）	36.07 0.9224	39.39 0.9540	41.42 0.9663	43.02 0.9744

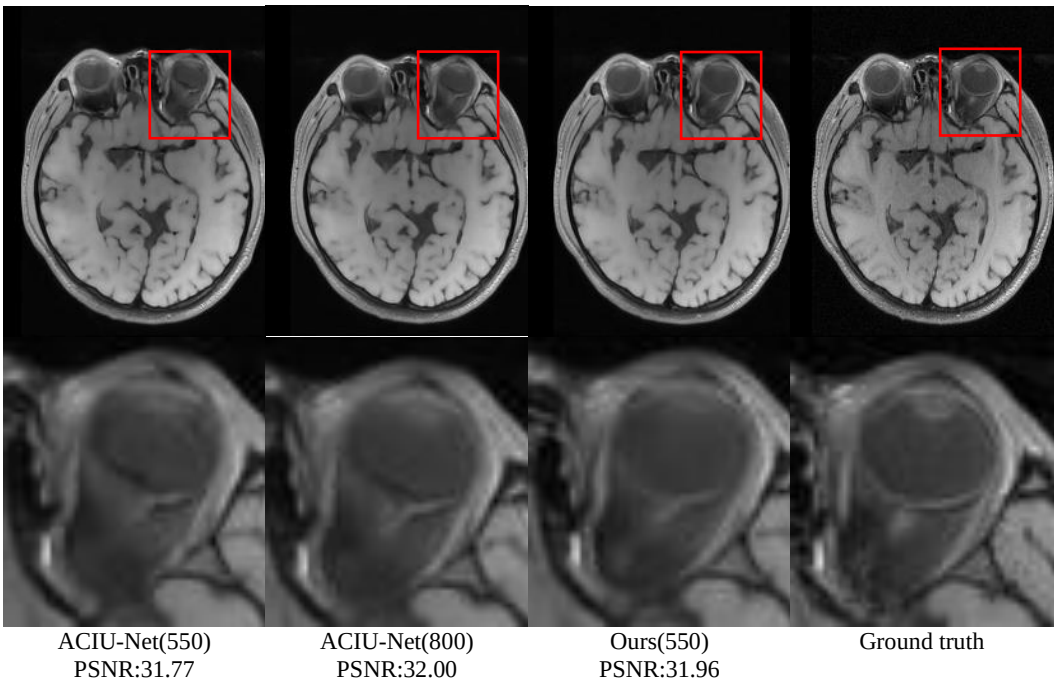


Fig 11: Reconstruction examples in the comparison of SL_0-MSNet with ACIU-Net on 550 Brain dataset samples

breakthrough in data efficiency and reconstruction performance through architectural and algorithmic

optimizations.

4. Conclusions and Future Work

This paper proposes a Multi-scale Reconstruction Network (Sℓ0-MSNet) based on the ADMM framework. By introducing ℓ0-norm sparse constraint and designing a sigmoid-based smooth function, we address the issues of insufficient sparsity and detail loss caused by ℓ1-norm relaxation. In addition, we develop the MSFAF module to fuse multi-scale features using convolution kernels of different sizes and a gating mechanism, further enhancing reconstruction capabilities in complex anatomical scenarios. Experimental results show that our method outperforms state-of-the-art algorithms in PSNR and SSIM across various datasets and sampling rates, particularly excelling in preserving details and edges at low sampling rates. Future work can focus on improving the reconstruction efficiency and detail fidelity of the model at extremely low sampling rates.

5. References

1. Xiang JX, Dong YG, Yang YJ. FISTA-Net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging. IEEE Trans Med Imaging. 2021;40(5):1329–1339.
2. Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci. 2009;2(1):183–202.
3. Boyd S, Parikh N, Chu E, *et al*. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn. 2011;3(1):1–122.
4. Wang ZY, Wang XP, Sohel F, *et al*. PD-Net: Point Dropping Network for flexible adversarial example generation with l₀ regularization. In: Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN); 2021. p. 1–7.
5. Kulkarni K, Lohit S, Turaga P, *et al*. Recon-net: Non-iterative reconstruction of images from compressively sensed measurements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 449–458.
6. Zhang J, Ghanem B. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 1828–1837.
7. Sun J, Yang Y, Li HB, *et al*. Deep ADMM-Net for compressive sensing MRI. In: Advances in Neural Information Processing Systems (NeurIPS); 2016. p. 10–18.
8. Ma M, Gan H. Attention-enhanced convolutional iterative unrolling network for under-sampled MRI reconstruction. IEEE Trans Emerg Top Comput Intell. 2024;9:1177–1188.
9. Mun S, Fowler JE. Block compressed sensing of images using directional transforms. In: Proceedings of the 16th IEEE International Conference on Image Processing (ICIP); 2009. p. 3021–3024.
10. Chen XJ, Ge DD, Wang ZZ, *et al*. Complexity of unconstrained l_2–l_p minimization. Math Program. 2014;143:371–383.
11. Yang C, Nie K, Qiao J, *et al*. Design of extreme learning machine with smoothed l₀ regularization. Mob Netw Appl. 2020;25(6):2434–2446.
12. Zhang H, Tang Y, Liu X. Batch gradient training method with smoothing l₀ regularization for feedforward neural networks. Neural Comput Appl. 2015;26(2):383–390.
13. Nguyen TT, Thang VD, Nguyen VT, *et al*. SGD method for entropy error function with smoothing l₀ regularization for neural networks. Appl Intell. 2024;54(13):7213–7228.
14. Panda G, Kundu S, Bhattacharya S, *et al*. l₀-regularized sparse coding-based interpretable network for multi-modal image fusion. arXiv. 2024.
15. Mount J. The equivalence of logistic regression and maximum entropy models. 2011.
16. Dong C, Loy CC, He KM, *et al*. Learning a deep convolutional network for image super-resolution. In: Computer Vision – ECCV 2014: 13th European Conference; 2014. p. 184–199.
17. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw. 1989;2(5):359–366.
18. Wang Y, Li YS, Wang G, *et al*. Multi-scale attention network for single image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 5950–5960.
19. Mentaschi L, Besio G, Cassola F, *et al*. Why NRMSE is not completely reliable for forecast/hindcast model test performances. In: Geophys Res Abstr; 2013. p. 1–2.
20. Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR); 2015. p. 1–2.
21. Zbontar J, Knoll F, Sriram A, *et al*. fastMRI: An open dataset and benchmarks for accelerated MRI. arXiv. 2018;1811.08839.
22. Yang Y, Sun J, Li H, *et al*. ADMM-CSNet: A deep learning approach for image compressive sensing. IEEE Trans Pattern Anal Mach Intell. 2020;42(3):521–538.
23. Xie J, Zhang J, Zhang Y, *et al*. PUERT: Probabilistic under-sampling and explicable reconstruction network for CS-MRI. IEEE J Sel Top Signal Process. 2022;16(4):737–749.
24. Gregor K, LeCun Y. Learning fast approximations of sparse coding. In: Proceedings of the 27th International Conference on Machine Learning (ICML); 2010. p. 399–406.

How to Cite This Article

Li YY. SL₀-MSNet: Multi-scale unfolding network with smooth l₀ regularization for CS-MRI. Int J Artif Intell Eng Transform. 2026;7(1):13–23. doi:10.54660/IJAIET.2026.7.1.13-23.