
Time-Embedded Algorithm Unrolling for Computational MRI

Junno Yun

University of Minnesota
yun00049@umn.edu

Yaşar Utku Alçalar

University of Minnesota
alca1029@umn.edu

Mehmet Akçakaya*

University of Minnesota
akcakaya@umn.edu

Abstract

Algorithm unrolling methods have proven powerful for solving the regularized least squares problem in computational magnetic resonance imaging (MRI). These approaches unfold an iterative algorithm with a fixed number of iterations, typically alternating between a neural network-based proximal operator for regularization, a data fidelity operation and auxiliary updates with learnable parameters. While the connection to optimization methods dictate that the proximal operator network should be shared across unrolls, this can introduce artifacts or blurring. Heuristically, practitioners have shown that using distinct networks may be beneficial, but this significantly increases the number of learnable parameters, making it challenging to prevent overfitting. To address these shortcomings, by taking inspirations from proximal operators with varying thresholds in approximate message passing (AMP) and the success of time-embedding in diffusion models, we propose a time-embedded algorithm unrolling scheme for inverse problems. Specifically, we introduce a novel perspective on the iteration-dependent proximal operation in vector AMP (VAMP) and the subsequent Onsager correction in the context of algorithm unrolling, framing them as a time-embedded neural network. Similarly, the scalar weights in the data fidelity operation and its associated Onsager correction are cast as time-dependent learnable parameters. Our extensive experiments on the fastMRI dataset, spanning various acceleration rates and datasets, demonstrate that our method effectively reduces aliasing artifacts and mitigates noise amplification, achieving state-of-the-art performance. Furthermore, we show that our time-embedding strategy extends to existing algorithm unrolling approaches, enhancing reconstruction quality without increasing the computational complexity significantly. Code available at <https://github.com/JN-Yun/TE-Unrolling-MRI>.

1 Introduction

Algorithm unrolling/unfolding has emerged as an effective method for addressing inverse problems in computational MRI [40, 28, 3, 29, 37, 50, 43, 54, 64]. In this framework, traditional iterative optimization problems are unrolled for a fixed number of steps, with the network alternating between enforcing data fidelity based on the known physics-based forward operator and applying implicit regularization via a neural network based proximal operator. This unrolled network is trained end-to-end to jointly optimize the weight(s) for data fidelity and the neural network parameters for the proximal operator. Several different optimization methods have been explored for algorithm unrolling in MRI [40, 37, 25, 34, 63, 33], including gradient descent (GD) [28], proximal gradient descent (PGD) [54, 69, 33, 43], variable splitting with quadratic penalty (VSQP) [3, 24, 64] and alternating direction method of multipliers (ADMM) [58, 66], among others [50, 1].

*Corresponding Author

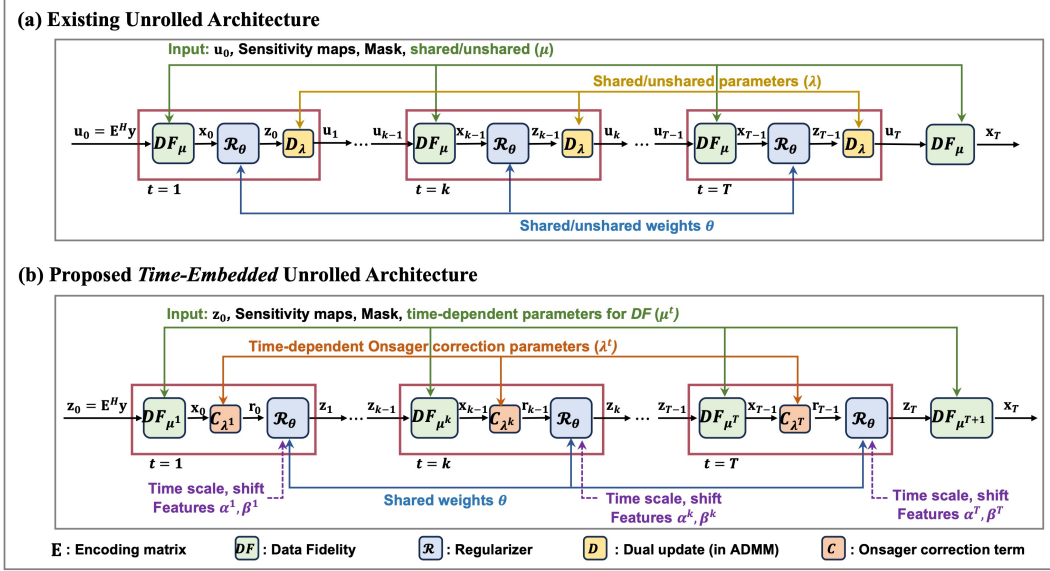


Figure 1: Descriptions of (a) the existing unrolled architecture and (b) the proposed time-embedded unrolled architecture.

Beyond the choice of optimization framework, another critical design decision in algorithm unrolling is whether to share the proximal operator for the learned regularizer across unrolls. While theoretical connections to optimization theory suggest that the proximal operator should remain fixed to maintain consistency with traditional methods [3, 46], this may lead to unwanted artifacts. To address this, many practitioners instead allow the proximal operator to vary across iterations, effectively using distinct networks to learn iteration-specific regularization [28, 38, 47]. This empirical strategy often enhances reconstruction quality but comes with practical trade-offs, such as larger number of trainable parameters which can heighten the risk of over-fitting, especially for applications with *limited training data* [6, 63, 17]. Such limited data settings, which are the focus of this study, are especially important in many translational applications, where new sequences are being implemented or higher resolutions are being pursued, as it is often not feasible to curate databases with thousands of slices.

A related perspective on iterative reconstruction emerges from approximate message passing (AMP) methods, which have been developed as an iterative Bayesian estimator for recovering a sparse signal for certain classes of measurement matrices in the context of compressed sensing [23, 22]. AMP adapts the proximal operator at each iteration based on the prior distribution and includes an *Onsager correction* term to stabilize the process and accelerate signal recovery. A notable extension, vector AMP (VAMP) [52], improves this approach by introducing vector-valued variable nodes, and estimating each node using Minimum Mean Square Error (MMSE) and Linear MMSE (LMMSE) estimators while preserving the properties of AMP. This enables VAMP to remain effective for a broader class of measurement matrices, extending the applicability of AMP to a broader class of measurement matrices.

Similar to the iteration-dependent proximal operator in AMP methods, a time-dependent denoiser has shown to be highly effective in the context of diffusion models [31, 57, 21, 32, 35]. In diffusion-based approaches, the denoiser adapts dynamically at each time step to better preserve structure and enhance signal recovery [56, 31]. This time-dependent adjustment has been shown to outperform static denoisers, particularly in tasks requiring high fidelity and sharpness [31, 21], as it allows the model to better handle varying noise levels throughout the diffusion process.

Building on these principles from AMP methods and diffusion models, we propose a novel time-embedded unrolling of optimization algorithms, theoretically motivated from VAMP. Our main contributions are:

- We introduce time (or iteration)-dependent unrolling of optimization algorithms by incorporating time-information into the proximal operator, theoretically motivated by VAMP formalism. To the best of our knowledge, our approach is the first attempt to bring in the time information into

the algorithm unrolling to further improve performance with minimal increase in computational complexity.

- Our method also learns the guidance scale (*i.e.*, the data fidelity weight in our case) in a time-dependent manner during training, which is a major deviation from commonly used guidance methods in diffusion models [21, 32].
- We demonstrate that our time-embedding strategy can be extended to various optimization algorithms, such as VSQP and ADMM, and applied to different neural network architectures for the proximal operator.
- We showcase the efficacy of incorporating the time information to the unrolling process through both quantitative and qualitative assessments on fastMRI dataset [39, 68]. Our approach performs on par with methods that use distinct proximal operator weights, which has substantially more learnable parameters and may face performance decrease in small training database such as ours. Furthermore, our method consistently outperforms the baseline shared-regularizer approach across various acceleration rates, producing artifact-free reconstructions with minimal processing overhead.

2 Background and Related Work

2.1 Inverse Problems in Computational MRI

A canonical problem in computational MRI is to recover an image $\mathbf{x} \in \mathbb{C}^N$ from noisy sub-sampled measurements $\mathbf{y}_\Omega \in \mathbb{C}^M$. The forward model in this case is given as

$$\mathbf{y}_\Omega = \mathbf{E}_\Omega \mathbf{x} + \mathbf{n}, \quad (1)$$

where $\mathbf{E}_\Omega \in \mathbb{C}^{M \times N}$ is a known encoding matrix that samples the Fourier domain (*i.e.* k-space) locations specified by Ω , and includes coil sensitivities, and $\mathbf{n} \in \mathbb{C}^M$ is i.i.d. Gaussian measurement noise. The inverse problem corresponding to Eq. (1) is typically ill-conditioned [8, 41, 5, 4, 7], necessitating additional regularization to be incorporated into the objective function [29]:

$$\arg \min_{\mathbf{x}} \|\mathbf{y}_\Omega - \mathbf{E}_\Omega \mathbf{x}\|_2^2 + \mathcal{R}(\mathbf{x}), \quad (2)$$

where the first term ensures data fidelity with the acquired measurements and $\mathcal{R}(\cdot)$ is a regularizer.

2.2 Algorithm Unrolling

The optimization problem in Eq. (2) can be solved using various methods [25], including VSQP [2] and ADMM [14], all of which have been explored in algorithm unrolling. The unrolled network iterates between data fidelity and regularization, with the latter implicitly enforced via a neural network, as illustrated in Fig. 1(a). VSQP unrolling [3, 19, 24, 65, 64, 62] solves Eq. (2) via:

$$\mathbf{x}^t = (\mathbf{E}_\Omega^H \mathbf{E}_\Omega + \mu \mathbf{I})^{-1} (\mathbf{E}_\Omega^H \mathbf{y}_\Omega + \mu \mathbf{z}^t), \quad (3)$$

$$\mathbf{z}^{t+1} = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x}^t - \mathbf{z}\|_2^2 + \mathcal{R}(\mathbf{z}) \triangleq \text{Prox}_{\mathcal{R}}(\mathbf{x}^t), \quad (4)$$

where the data fidelity parameter μ is learnable, and $\text{Prox}_{\mathcal{R}}(\cdot)$ is learned implicitly via a neural network. While Eq. (3) has a closed-form solution, it is numerically solved using the CG method [3]. In contrast, ADMM is a commonly used optimization approach with better convergence than VSQP [14], owing to an additional Lagrangian update, and has been popular in algorithm unrolling [66, 58, 26, 20]:

$$\mathbf{x}^{t+1} = (\mathbf{E}_\Omega^H \mathbf{E}_\Omega + \mu \mathbf{I})^{-1} (\mathbf{E}_\Omega^H \mathbf{y}_\Omega + \mu (\mathbf{z}^t - \mathbf{u}^t)), \quad (5)$$

$$\mathbf{z}^{t+1} = \text{Prox}_{\mathcal{R}}(\mathbf{x}^{t+1} + \mathbf{u}^t), \quad (6)$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \lambda (\mathbf{x}^{t+1} - \mathbf{z}^{t+1}), \quad (7)$$

where $\mathcal{R}(\cdot)$, μ and λ are learnable [66, 58].

Although in all of the cases, optimization theory [25] dictates that $\mathcal{R}(\cdot)$ in Eq. (2) should be fixed across unrolls, researchers heuristically realized enabling $\mathcal{R}(\cdot)$ to change across iterations yields better reconstructions [38, 47]. However, this increases the number of trainable parameters, and the risk of overfitting, particularly in data-limited settings.

2.3 Approximate Message Passing

AMP [23, 22] provides an alternative approach to solving Eq. (1) when $\mathbf{E}$ is a large i.i.d. (sub-Gaussian) matrix. The AMP algorithm uses an *iteration-dependent* proximal operator and an *Onsager correction* term, which together enable faster convergence compared to PGD [18]. At iteration t , AMP applies the proximal operator with threshold proportional to σ^t that represents an estimate of the mean squared error of the current estimate. However, when the measurement matrix deviates from the i.i.d. sub-Gaussian regime, AMP methods often fail to converge [52].

Vector AMP (VAMP) algorithm [52] is an alternative, offering convergence in the large N limit for a broader class of matrices $\mathbf{E}$. It extends the AMP framework to vector-valued nodes [51, 48, 55], and has connections to the ADMM algorithm [42, 48], while preserving the desirable properties of AMP. These vector-valued operations lead to a data fidelity operation based on linear MMSE estimation, and its associated Onsager correction as:

$$\mathbf{x}^t = (\mathbf{E}^H \mathbf{E} + \mu_x^t \mathbf{I})^{-1} (\mathbf{E}^H \mathbf{y} + \mu_x^t \mathbf{r}^t), \quad (8)$$

$$v_x^t = \frac{1}{N} \text{Tr} [(\mathbf{E}^H \mathbf{E} + \mu_x^t \mathbf{I})^{-1}]; \mu_z^t = \frac{1}{v_x^t} - \mu_x^t; \mathbf{u}^t = \left(\frac{\mathbf{x}^t}{v_x^t} - \mu_x^t \mathbf{r}^t \right) / \mu_z^t \quad (9)$$

followed by the proximal operator/denoising step with its Onsager correction:

$$\mathbf{z}^t = \text{Prox}_{\mathcal{R}_{\mu_z^t}}(\mathbf{u}^t), \quad (10)$$

$$v_z^t = \frac{1}{\mu_z^t} \left\langle \nabla \text{Prox}_{\mathcal{R}_{\mu_z^t}}(\mathbf{u}^t) \right\rangle; \mu_x^{t+1} = \frac{1}{v_z^t} - \mu_z^t; \mathbf{r}^{t+1} = \left(\frac{\mathbf{z}^t}{v_z^t} - \mu_z^t \mathbf{u}^t \right) / \mu_x^{t+1}. \quad (11)$$

Notably, both data fidelity and denoising steps have parameters, μ_x^t, μ_z^t , which are functions of the iteration number.

We note that AMP and its variants have also been explored in the context of algorithm unrolling. In [12] and [13], $\mathbf{E}$ and $\mathbf{E}^H$ are reparameterized with tunable parameters as $\beta^t \mathbf{E}$ and $\mathbf{E}^H \mathbf{C}^t$, where β^t and $\mathbf{C}^t$ are trainable across unrollings via neural networks. This reparameterization influences the Onsager correction term and the denoising threshold, improving the robustness of $\mathbf{E}$. Subsequent studies have explored training distinct, *i.e.* unshared in our previous terminology, proximal operators [45, 36] over iterations using neural networks, rather than reparameterizing the matrix $\mathbf{E}$ and $\mathbf{E}^H$. Other studies have also explored training both the system matrix and proximal operators using neural networks [70, 36] across iterations.

2.4 Time-Embedding in Neural Networks

Time-dependent processing plays a crucial role in diffusion models as well [31, 57, 21, 32]. In this context, information about the current diffusion step is encoded to guide the CNN model to capture sequential relationships effectively and to reverse the noise process efficiently.

Feature-wise linear modulation (FiLM) [49] is widely utilized for transforming inputs with time-embedded features, as illustrated in Fig. 2 (a) and (b). The time information features are obtained through a sinusoidal encoder [60], followed by a learned function $f(\cdot)$. Subsequently, the functions g_i and h_i are adaptively learned to generate α_i^t and β_i^t , respectively as:

$$\alpha_i^t = g_i(f(t)); \quad \beta_i^t = h_i(f(t)), \quad (12)$$

where α_i^t and β_i^t modulate the i^{th} features $\mathcal{F}_i^t$ of CNNs at the t^{th} iteration using FiLM, which applies scaling and shifting transformations using α_i^t and β_i^t respectively. Moreover, [21] demonstrates that combining group normalization [61] with the FiLM approach enhances the efficacy of time-embedded features, leading to improved model performance in diffusion model as follows:

$$\mathcal{H}_i^t = \alpha_i^t \odot \text{GroupNorm}(\mathcal{F}_i^t) \oplus \beta_i^t, \quad (13)$$

where $\mathcal{H}_i^t$ are features conditioned by time-embedded layers, $\odot$ is feature-wise multiplication, $\oplus$ is feature-wise addition. Each feature map in the network is modulated independently by α_i^t and β_i^t . For example, $\mathcal{H}_i^t$ is passed onto the next block as input $\mathcal{F}_{i+1}^t$ and modulated by the next time-embedded scaling and shifting factors α_{i+1}^t and β_{i+1}^t .

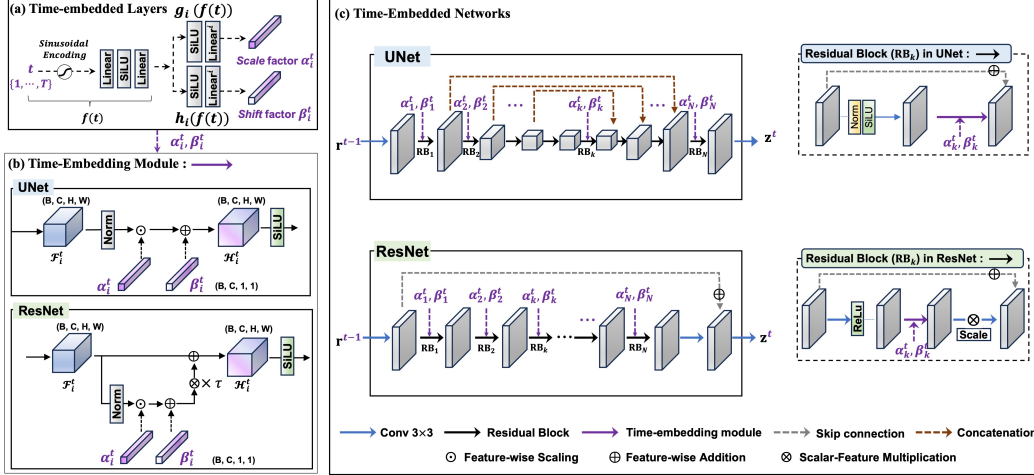


Figure 2: Illustrations of (a) Positional Encoder generating time-dependent scaling (α_t^t) and shift (β_t^t) features, (b) Time-embedding module for ResNet and U-Net, and (c) Architectures of ResNet and U-Net showing how time-embedded features are applied.

3 Methodology

3.1 Proposed Time-Embedding Strategies for Algorithm Unrolling

Building on Section 2.3 and Section 2.4, we propose a time-embedding framework for algorithm unrolling. Our time-dependent proximal operator is inspired by VAMP, and is implemented as a CNN with the time-embedding techniques from Section 2.4. This enables the proximal operator to exploit temporal dependencies across iterations, adapting its behavior dynamically, similar to the denoising in diffusion models. We note that, while the time step t explicitly models the noise level in the given stage in diffusion models, in our case with algorithm unrolling, it implicitly modulates the proximal operator’s behavior across *iterations* by capturing the evolving distribution of intermediate features, similar to the effect of the Onsager correction in VAMP.

Time-embedding in proximal operators We first consider the proximal operator step of VAMP in Eq. (10) and its corresponding Onsager corrections in Eq. (11). In the context of algorithm unrolling, these suggest that the learned proximal operator should be time-dependent, but with a well-defined time schedule. Furthermore, the Onsager corrections in Eq. (11) are also functions of time in the original VAMP setting, and $\mathbf{r}^{t+1}$ in Eq. (11) is effectively a function of $\mathbf{u}^t$ in Eq. (10) and t . To this end, in an unrolled network setting, where the intermediate parameters can also be learned, we propose to model this relationship directly with a time-embedded neural network. In other words, a time-embedded neural network is used to model all steps in (10)-(11), effectively capturing both the time-dependent denoising and the associated Onsager corrections implicitly to map $\mathbf{r}^{t+1}$ from $\mathbf{u}^t$. We represent this relationship as:

$$\mathbf{r}^{t+1} = \text{prox}_{\mathcal{R}}(\mathbf{u}^{t+1}, \alpha^t, \beta^t, t), \quad (14)$$

where α^t and β^t capture the time-embedding information as described in Section 2.4.

Time-embedding for data fidelity The data fidelity term in Eq. (8) is of the same form as the data fidelity term in Eq. (3) in VSQP, with the notable distinction that the quadratic penalty μ^t evolves in a time-dependent manner in the former. Thus, we implement μ^t as a time-dependent learnable parameter. Furthermore, the Onsager correction in (9) can be written as:

$$\mathbf{u}^t = \mathbf{x}^t + \rho^t(\mathbf{x}^t - \mathbf{r}^t), \quad (15)$$

Algorithm 1 Time-embedded Unrolling Algorithms

Require: $T, \mathbf{E}_{\Omega}, \mathbf{y}_{\Omega}$

- 1: Initialize $\mathbf{r}^0$ and $\mu^0, \rho^0 \geq 0$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\mathbf{x}^{t+1} = (\mathbf{E}_{\Omega}^H \mathbf{E}_{\Omega} + \mu^t \mathbf{I})^{-1} (\mathbf{E}_{\Omega}^H \mathbf{y}_{\Omega} + \mu^t \mathbf{r}^t)$,
- 4: $\mathbf{u}^{t+1} = \mathbf{x}^{t+1} + \rho^t(\mathbf{x}^{t+1} - \mathbf{r}^t)$,
- 5: $\mathbf{r}^{t+1} = \text{prox}_{\mathcal{R}}(\mathbf{u}^{t+1}, \alpha^t, \beta^t, t)$
- 6: **end for**
- 7: **return** $\mathbf{x}^{(T)}$

Table 1: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Quantitative results are reported using *limited data* on the coronal PD, coronal PD-FS, and axial T2 datasets, with equispaced undersampling patterns at acceleration rates $R = 4, 6$, and 8 . The **best** and **second-best** results for each architecture are highlighted.

	R	U-Net						ResNet					
		VSQP (♠)		ADMM (♠)		Ours		VSQP (♠)		ADMM (♠)		Ours	
		VSQP (♣)	ADMM (♣)	ADMM (♣)	ADMM (♣)	(5 unrolls)	(10 unrolls)	VSQP (♣)	ADMM (♣)	ADMM (♣)	ADMM (♣)	(5 unrolls)	(10 unrolls)
Coronal PD	$\times 4$	PSNR $\uparrow$	40.50	40.31	40.76	40.51	40.94	40.99	41.11	40.99	41.27	41.11	41.41
		SSIM $\uparrow$	0.962	0.960	0.964	0.963	0.964	0.964	0.965	0.963	0.964	0.964	0.966
	$\times 6$	PSNR $\uparrow$	38.12	38.02	38.85	38.52	39.08	38.93	39.54	39.18	39.61	39.60	39.65
		SSIM $\uparrow$	0.945	0.942	0.950	0.949	0.952	0.950	0.954	0.950	0.953	0.953	0.954
	$\times 8$	PSNR $\uparrow$	35.98	35.61	36.31	35.71	36.45	36.34	36.46	36.04	36.72	36.41	36.87
		SSIM $\uparrow$	0.920	0.914	0.924	0.917	0.925	0.923	0.924	0.919	0.926	0.921	0.925
Coronal PD-FS	$\times 4$	PSNR $\uparrow$	35.09	35.10	35.31	35.23	35.23	35.38	35.31	35.23	35.37	35.23	35.42
		SSIM $\uparrow$	0.849	0.847	0.851	0.848	0.847	0.851	0.851	0.847	0.848	0.849	0.847
	$\times 6$	PSNR $\uparrow$	34.17	34.05	34.26	34.27	34.29	34.44	34.48	34.25	34.53	34.33	34.54
		SSIM $\uparrow$	0.821	0.817	0.821	0.824	0.822	0.825	0.823	0.820	0.822	0.823	0.822
	$\times 8$	PSNR $\uparrow$	33.11	32.86	33.21	33.06	33.27	33.36	33.09	32.71	33.35	33.09	33.48
		SSIM $\uparrow$	0.794	0.791	0.795	0.796	0.797	0.797	0.796	0.785	0.796	0.789	0.794
Axial T2	$\times 4$	PSNR $\uparrow$	36.37	36.42	36.60	36.54	36.59	36.60	36.63	36.53	36.81	36.75	36.81
		SSIM $\uparrow$	0.927	0.926	0.928	0.928	0.925	0.928	0.926	0.923	0.925	0.926	0.927
	$\times 6$	PSNR $\uparrow$	34.53	34.69	35.05	34.91	35.03	35.09	35.07	34.94	35.35	35.10	35.37
		SSIM $\uparrow$	0.903	0.910	0.910	0.910	0.906	0.909	0.913	0.906	0.910	0.909	0.910
	$\times 8$	PSNR $\uparrow$	32.90	32.70	33.41	32.98	33.26	33.41	33.15	32.99	33.43	33.14	33.67
		SSIM $\uparrow$	0.889	0.889	0.893	0.890	0.890	0.892	0.894	0.885	0.890	0.889	0.891

where $\rho^t = \frac{\mu_x^t}{1/v_x^t - \mu_x^t}$. Thus, in the algorithm unrolling framework, the scalars in Eq. (9) can be replaced with a time-dependent learnable parameter ρ^t for a learned Onsager correction term. Thus, the full time-embedded unrolled network is summarized in Alg. 1.

3.2 Neural Network Architectures for Time-Embedded Proximal Operators

The U-Net architecture [53] has been widely used, especially in the context of diffusion models, as a time-embedded network by integrating time-embedding features into different layers [31, 57, 21, 32], as shown in Fig. 2 (c). We follow the time-embedded U-Net design based on ADM from [21], which employs group normalization and the FiLM method, as formulated in Eq. (12) and Eq. (13), with modifications in the number of channels and up/down sampling. We also note that U-Net has connections with message passing through belief propagation [44].

We additionally propose a novel time-embedding module for ResNet [30], which is commonly used as a proximal operator in unrolling algorithms for MR reconstruction [64, 63], as shown in Fig. 2 (b) and (c). The time-embedding module in ResNet is designed as:

$$\mathcal{H}_i^t = \mathcal{F}_i^t + \tau \times (\alpha_i^t \odot \text{GroupNorm}(\mathcal{F}_i^t) \oplus \beta_i^t), \quad (16)$$

where α_i^t and β_i^t are as in Eq. (12), and τ is a scaling factor. Instead of directly applying the transformed features from α_i^t and β_i^t , this module utilizes τ as a scaling factor to indirectly influence the features. This approach ensures a stable integration of time information into the ResNet architecture.

4 Experiments and Results

4.1 Experimental Setup

We carried out an in-depth assessment of our approach, analyzing its effectiveness quantitatively and visually through multiple acceleration rates and datasets. The data included fully sampled coronal proton density (PD) and PD with fat-suppression (PD-FS) knee MRI scans, as well as axial T2-weighted brain MRI scans. These scans were obtained from the New York University (NYU) fastMRI database [39, 68], and were acquired with appropriate institutional review board approvals.

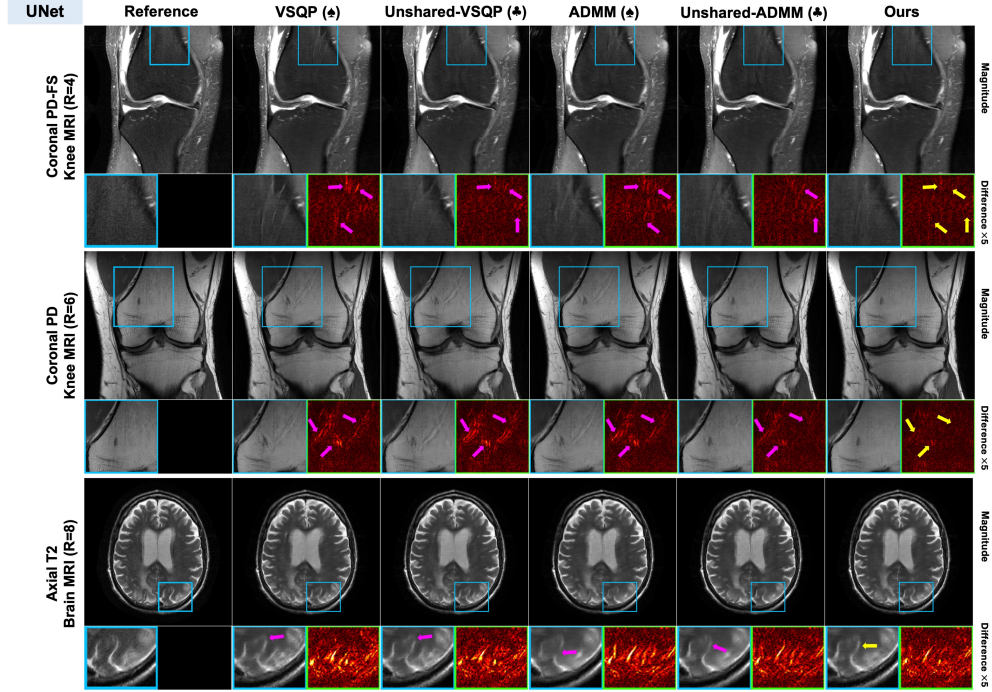


Figure 3: Qualitative comparisons between the standard shared (♠) and unshared (♣) $\mathcal{R}(\cdot)$ optimization methods (VS, ADMM) and the proposed time-embedded unrolled algorithms with $T = 10$ unrolls in U-Net. **Top:** Results for $R = 4$ using PD-FS data. **Middle:** Results for $R = 6$ using PD data. **Bottom:** Results for $R = 8$ using Axial T2-W data. The proposed methods reduce artifacts (yellow arrow) that the shared and unshared methods fail to eliminate (pink arrow).

All datasets were retrospectively undersampled using uniform/equidistant undersampling at acceleration factors of $R = 4, 6$, and 8 with 24 central kspace lines kept. For this study, we focused on uniform/equidistant undersampling patterns, as they are more commonly used in clinical practice and produce coherent artifacts that are more challenging to remove [29]. We additionally evaluate the generalization performance of our method on random undersampling patterns in Section 4.6. For knee datasets, model training was conducted using 300 slices from 10 subjects, while testing was carried out on 380 slices from a separate set of 10 subjects [28]. For the brain dataset, training and testing were performed using 300 slices each.

4.2 Implementation Details

We compared our method with conventional algorithm unrolling based on VSQP and ADMM, which were implemented with both shared [3, 64] and unshared [28, 47] weights across iterations. We note that multiple variants of ADMM and VSQP unrolling [64, 66, 67, 24] have been proposed with different names, primarily differing in their choice of network architectures for the proximal step and their training strategies. In this work, our focus is not on comparing these variations, but rather on analyzing the effect of the outer algorithm unrolling itself with matching proximal operator network structures and training processes. For the least squares problem in the data fidelity of these approaches, conjugate gradient (CG) with 15 iterations was utilized [3]. For the proximal operators, we chose two distinct network architectures: 1) a ResNet model with 15 residual blocks, where each block consists of 3×3 convolutional layers with 64 channels [64], and 2) a U-Net model, adapted from the ADM diffusion model [21] with slight modifications to number of channels and up/down sampling layers. Details about model architectures and hyperparameters are provided in Appendix A. The comparisons were first divided by the proximal network architecture, *i.e.* ResNet vs U-Net based. For each of these two proximal network architectures, we trained five unrolled networks from scratch: the proposed

Table 2: The number of parameters for the shared (♠), unshared (♣), and our proposed methods using different networks with $T = 10$ unrolls.

Networks	VSQP (♠)	VSQP (♣)	ADMM (♠)	ADMM (♣)	Ours
U-Net	1,724,035	17,240,341	1,724,036	17,240,342	1,963,479
ResNet	592,129	5,921,281	592,130	5,921,282	866,581

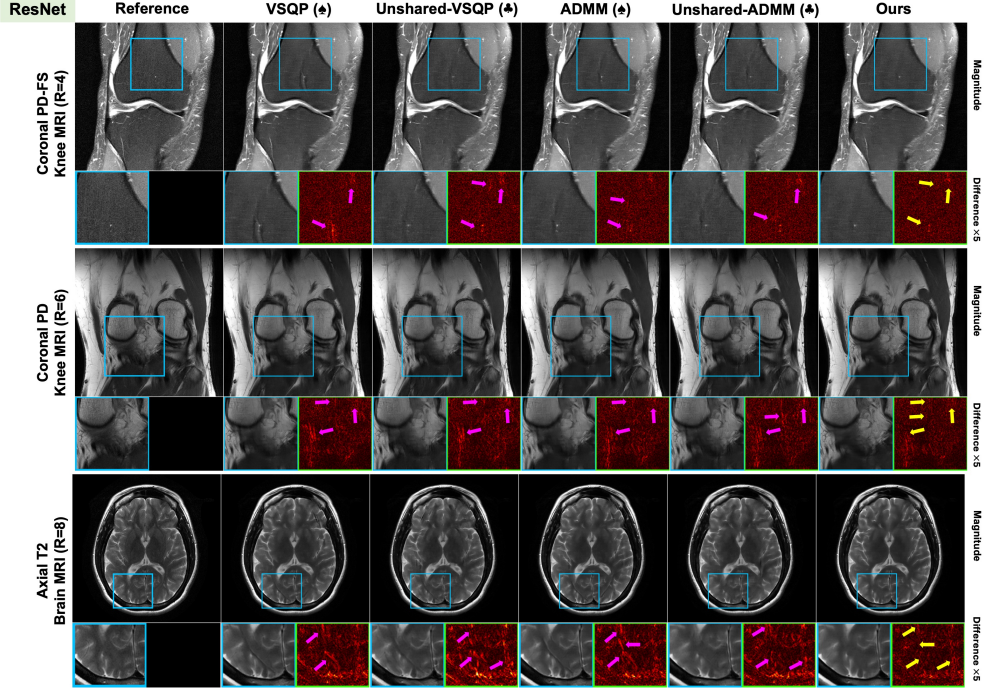


Figure 4: Qualitative comparisons using ResNet (instead of U-Net in Fig. 3). The proposed methods reduce artifacts (yellow arrow) that the shared and unshared methods fail to eliminate (pink arrow).

time-embedded unrolling, VSQP with shared parameters and unshared parameters, ADMM with shared and unshared parameters. Note that incorporating time embedding into the shared proximal networks leads to a modest increase in parameters, as shown in Tab. 2, which is considerably smaller than the increase caused by using a distinct unshared regularizer at each unroll. This shows the efficacy of our proposed approach, which adapts the regularizer over time without significantly increasing network size.

4.3 Performance of Time-Embedded Unrolling versus Existing Methods

Quantitative Results Tab. 1 depicts the performance of different approaches on the Coronal PD, PD-FS, and Axial T2 datasets across different acceleration rates. The shared and unshared baselines are trained with $T = 10$ unrolls, while our proposed method is trained with both $T = 5$ and 10 unrolls. In almost all cases, the unshared baselines perform worse than their shared counterparts for both U-Net and ResNet in this *limited data* setting. Though unshared methods are known to generalize well in large data regimes [47], in our limited data setting, they exhibit performance degradation due to the high number of trainable parameters, as further detailed in Appendix B with experiments on fine-tuning from pretrained shared baselines. Our proposed method with $T = 10$ unrolls outperforms both shared and unshared methods, achieving the best or second-best performance across all acceleration rates and datasets. The only exceptions are SSIM on Coronal PD at $R = 8$ and Axial T2 at $R = 6$ when using the U-Net proximal operator, and SSIM on Coronal PD-FS at $R = 6$ and $R = 8$ when using the ResNet proximal operator. These results demonstrate that our proposed method performs best in the limited training data regime for a fixed number of unrolls.

Remarkably, even with $T = 5$ unrolls, our proposed method achieves performance comparable to the shared baselines with $T = 10$ unrolls in most cases. The only notable exception is the Axial T2 dataset, where a small performance gap remains compared to the top-performing methods. This shows that our method can deliver strong performance while halving the number of network computations and reducing inference time by $\sim 50\%$, offering a substantial advantage in clinical applications.

Overall, ResNet-based unrolling networks demonstrate stronger quantitative performance than U-Net-based ones. Additionally, there are a few cases where our performance does not fall within the second-best range. Nevertheless, we note that PSNR and SSIM do not necessarily capture finer

details, as noted in earlier studies [11, 47, 38, 15, 9]. Therefore, in the next section, we provide qualitative results to further demonstrate the effectiveness of our approach.

Qualitative Results Fig. 3 and Fig. 4 depict representative reconstructions from different unrolling approaches with $T = 10$ unrolls using U-Net and ResNet based proximal operators, respectively. The shared VSQP and ADMM exhibit artifacts for both proximal operators in various cases. The unshared versions of these methods, which consist of 10 independent regularizers, cannot properly mitigate these artifacts (arrows). In contrast, our proposed method with the time-embedded proximal operators effectively addresses these artifacts across all acceleration rates and datasets. In addition to artifact reduction, our proposed method also enhances image sharpness, most clearly visible in the Axial T2 example in Fig. 3. More visual examples are provided in Appendix G.

To further investigate the subtle and diagnostically important improvements afforded by our method, representative reconstructions were reviewed by an expert musculoskeletal radiologist, who was blinded to the reconstruction method. The reviewed images included the cases shown in Fig. 3 and Fig. 4, as well as sample annotated pathology cases from the fastMRI+ [71] dataset. The radiologist noted that our method was able to remove subtle artifacts that were observed with the other methods in all the reviewed cases. A detailed description of the readings and the artifacts, as well as the pathological region assessments on fastMRI+ are provided in Appendix G. These evaluations further confirm the strong performance of our approach beyond standard quantitative metrics.

4.4 Extension to Other Unrolling Algorithms

Steps 3 and 5 of Alg. 1 are analogous to the VSQP updates in Eq. (3)-(4), except with a time-dependent quadratic penalty parameter in the former and a time-dependent proximal operator in the latter. Thus, we set out to explore the effect of the Onsager correction in Step 4 of Alg. 1, specifically to empirically characterize whether the correction to the data fidelity output is minimal, *i.e.*, $\mathbf{x}^{t+1} \approx \mathbf{u}^{t+1}$. Indeed, we observed that the network made only minor differences between $\mathbf{x}^{t+1}$ and $\mathbf{u}^{t+1}$ over iterations.

Details are provided in Appendix C. Omitting the Onsager correction in Step 4 turns Alg. 1 to a time-embedded version of VSQP. Similarly, we can also unroll other algorithms, such as ADMM in the proposed time-embedded manner. As shown in Tab. 3, time-embedded proximal units and data fidelity parameters improve the performance of VSQP and ADMM across all acceleration rates. Like our method in Section 4.3, other time-embedded unrolled networks also exhibit superior performance in effectively reducing artifacts. Further details are presented in Appendix D.

Table 3: Comparison of shared baseline methods and their time-embedded versions (baseline-TE) using the U-Net proximal operator on coronal PD.

R		VSQP	VSQP-TE	ADMM	ADMM-TE
$\times 4$	PSNR $\uparrow$	40.50	40.92	40.76	40.87
	SSIM $\uparrow$	0.962	0.964	0.964	0.964
$\times 6$	PSNR $\uparrow$	38.12	38.50	38.85	38.87
	SSIM $\uparrow$	0.945	0.946	0.950	0.951
$\times 8$	PSNR $\uparrow$	35.98	36.15	36.31	36.48
	SSIM $\uparrow$	0.920	0.921	0.924	0.925

4.5 Ablation Studies

We conducted three ablation studies to evaluate the effects of varying hyperparameters of time-embedded unrolled networks.

Effect of Varying the Numbers of Unrolls We trained all the methods in Tab. 1 for $T \in \{5, 15\}$. Our proposed method maintained stable performance regardless of the number of unrolls, consistently reducing artifacts, while yielding sharp images, whereas other baselines exhibited varying performance depending on the number of iterations. Further details are given in Appendix E.

Efficiency Analysis with Respect to the Number of Parameters We investigated whether increasing the number of trainable parameters improves performance, in the shared and unshared baselines. Increasing parameters do not improve results in ResNet, whereas a slight quantitative gain is observed for U-Net, though visual residual artifacts persist. These larger models also incur higher computational costs, whereas our method achieves better performance with only a marginal increase in parameters. Detailed experimental results are provided in Appendix E Tab. 7.

Time-Embedding Module with Different Hyperparameters We conducted experiments with a time-embedded U-Net to evaluate key hyperparameters of the time-embedding module, including the sinusoidal encoding frequency, embedding dimension, and the number of hidden channels in the MLP layers. Performance was influenced by these hyperparameters, with optimal results achieved using a period of 10,000, an embedding dimension of 32, and 128 hidden channels. These settings were applied consistently across all experiments. Additional information is in Appendix E Tab. 8.

4.6 Extended Experiments

Artifact Evolution Across Unrolls While the time step t in diffusion models explicitly determines the noise level at each stage of the forward process, the time step t in our setting plays a different role. t implicitly governs the evolution of the proximal operator across iterations by accounting for the changing distribution of intermediate features, analogous to the Onsager correction term in VAMP, which stabilizes updates by compensating for iterative correlations. This enables time-embedded proximal operators with temporal information to adaptively apply varying levels of denoising at different stages, which is further illustrated in Appendix D Fig. 6.

Validation on Non-Uniform Sampling Masks To evaluate the effectiveness of our proposed method under non-uniform (random) sampling patterns, we conducted experiments at various acceleration rates, using both baseline methods and our approach with a U-Net architecture and 10 unrolls. These results confirm that the effectiveness of our method extends to non-uniform undersampling patterns. Results are provided in Appendix F.

Comparison with Diffusion-Based Models Since diffusion-based reconstruction provides a promising approach for solving MR inverse problems [35, 16, 10, 27], we compared our results with Decomposed Diffusion Sampling (DDS) [16]. Our method outperforms DDS in terms of both PSNR and SSIM. Details of the implementation and the corresponding results are provided in Appendix F.

5 Limitations and Discussion

Limitations. As discussed in Section 1 and Section 4, our experiments were conducted in a limited data regime, using 300 slices per dataset. This setting is particularly relevant for translational applications, where new imaging sequences are being developed or when higher resolutions are targeted. While our method demonstrated strong performance and generalization in this regime, the performance gap between our approach and unshared baselines may narrow when training with more data samples, as the risk of overfitting will be lower for the latter.

Discussion. Through empirical evaluation, we examined whether second-moment matching holds in our time-embedded unrolling algorithms inspired by the Vector AMP framework, as well as the Lipschitz constants and stability of the time-embedded FiLM layers. Detailed discussions are provided in Appendix I.

6 Conclusion

In this study, we introduced a time-embedded algorithm unrolling framework inspired by AMP theory and time-embedding in diffusion models. Our unrolled networks used time-embedding in proximal operators, which performed both denoising and Onsager correction, as well as in data fidelity weights. We extended these ideas to VSQP and ADMM-based unrolling, demonstrating the framework’s versatility. Our method outperformed both shared and unshared unrolling approaches under matched settings, producing sharper images with fewer artifacts, especially in limited data regime. Unlike unshared models, which showed signs of overfitting, our method generalized better and remained robust across different unroll depths.

7 Acknowledgements

The authors gratefully acknowledge Dr. Jutta Ellermann for providing expert radiologist assessments during the rebuttal stage on short notice. This work was partially supported by NIH R01HL153146, NIH R01EB032830, NIH P41EB027061.

References

- [1] J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE Trans. Med. Imag.*, 37(6):1322–1332, 2018.
- [2] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.*, 19(9):2345–2356, 2010.
- [3] H. K. Aggarwal, M. P. Mani, and M. Jacob. MoDL: Model-based deep learning architecture for inverse problems. *IEEE Trans. Med. Imag.*, 38(2):394–405, 2019.
- [4] M. Akçakaya, T. A. Basha, B. Goddu, L. A. Goepfert, K. V. Kissinger, V. Tarokh, W. J. Manning, and R. Nezafat. Low-dimensional-structure self-learning and thresholding: regularization beyond compressed sensing for MRI reconstruction. *Magn. Reson. Med.*, 66(3):756–767, 2011.
- [5] M. Akçakaya, P. Hu, M. L. Chuang, T. H. Hauser, L. H. Ngo, W. J. Manning, V. Tarokh, and R. Nezafat. Accelerated noncontrast-enhanced pulmonary vein MRA with distributed compressed sensing. *J. Magn. Reson. Imaging*, 33(5):1248–1255, May 2011.
- [6] M. Akçakaya, S. Moeller, S. Weingärtner, and K. Uğurbil. Scan-specific robust artificial-neural-networks for k-space interpolation (RAKI) reconstruction: Database-free deep learning for fast imaging. *Magn. Reson. Med.*, 81(1):439–453, Jan. 2019.
- [7] M. Akçakaya, S. Nam, P. Hu, M. H. Moghari, L. H. Ngo, V. Tarokh, W. J. Manning, and R. Nezafat. Compressed sensing with wavelet domain dependencies for coronary MRI: a retrospective study. *IEEE Trans. Med. Imag.*, 30(5):1090–1099, 2010.
- [8] M. Akçakaya, M. Doneva, and C. Prieto (eds.). *Magnetic Resonance Image Reconstruction: Theory, Methods, and Applications*, volume 7 of *Advances in Magnetic Resonance Technology and Applications*. Academic Press, 2022.
- [9] Y. U. Alçalar and M. Akçakaya. Zero-shot adaptation for approximate posterior sampling of diffusion models in inverse problems. In *Proc. Eur. Conf. Comput. Vis.*, pages 444–460, 2024.
- [10] Y. U. Alçalar, J. Yun, and M. Akçakaya. Automated tuning for diffusion inverse problem solvers without generative prior retraining. In *Proc. IEEE Int. Workshop CAMSAP*, 2025.
- [11] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pages 6228–6237, 2018.
- [12] M. Borgerding and P. Schniter. Onsager-corrected deep learning for sparse linear inverse problems. In *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, pages 227–231, 2016.
- [13] M. Borgerding, P. Schniter, and S. Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Trans. Signal Process.*, 65(16):4293–4308, 2017.
- [14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.
- [15] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. In *Proc. Int. Conf. Learn. Represent.*, 2023.
- [16] H. Chung, S. Lee, and J. C. Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *Proc. Int. Conf. Learn. Represent.*, 2024.
- [17] M. Z. Darestani and R. Heckel. Accelerated MRI with un-trained neural networks. *IEEE Trans. Comput. Imag.*, 7:724–733, 2021.
- [18] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.*, 57(11):1413–1457, Nov. 2004.
- [19] Ö. B. Demirel, B. Yaman, L. Dowdle, S. Moeller, L. Vizioli, E. Yacoub, J. Strupp, C. A. Olman, K. Uğurbil, and M. Akçakaya. 20-fold accelerated 7T fMRI using referenceless self-supervised deep learning reconstruction. In *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, pages 3765–3769, 2021.
- [20] O. B. Demirel, B. Yaman, C. Shenoy, S. Moeller, S. Weingärtner, and M. Akçakaya. Signal intensity informed multi-coil encoding operator for physics-guided deep learning reconstruction of highly accelerated myocardial perfusion CMR. *Magn. Reson. Med.*, 89(1):308–321, Jan. 2023.

- [21] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 8780–8794, 2021.
- [22] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proc. Natl. Acad. Sci.*, 106(45):18914–18919, 2009.
- [23] D. L. Donoho, A. Maleki, and A. Montanari. Message passing algorithms for compressed sensing: I. motivation and construction. In *Proc. Inf. Theory Workshop*, pages 1–5, 2010.
- [24] J. Duan, J. Schlemper, C. Qin, C. Ouyang, W. Bai, C. Biffi, G. Bello, B. Statton, D. P. O’regan, and D. Rueckert. VS-Net: Variable splitting network for accelerated parallel MRI reconstruction. In *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, pages 713–722, 2019.
- [25] J. A. Fessler. Optimization methods for magnetic resonance image reconstruction. *IEEE Signal Process. Mag.*, 37(1):33–40, 2020.
- [26] H. Gu, B. Yaman, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya. Revisiting ℓ_1 -wavelet compressed-sensing MRI in the era of deep learning. *Proc. Natl. Acad. Sci.*, 119(33), 2022. Art. no. e2201062119.
- [27] M. Gülle, J. Yun, Y. U. Alçalar, and M. Akçakaya. Consistency models as plug-and-play priors for inverse problems, 2025. arXiv:2509.22736.
- [28] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magn. Reson. Med.*, 79(6):3055–3071, 2018.
- [29] K. Hammernik, T. Küstner, B. Yaman, Z. Huang, D. Rueckert, F. Knoll, and M. Akçakaya. Physics-driven deep learning for computational magnetic resonance imaging: Combining physics and machine learning for improved medical imaging. *IEEE Signal Process. Mag.*, 40(1):98–114, 2023.
- [30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.
- [31] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 6840–6851, 2020.
- [32] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *Proc. NeurIPS Workshop DGMs Appl.*, 2021.
- [33] S. A. H. Hosseini, B. Yaman, S. Moeller, M. Hong, and M. Akçakaya. Dense recurrent neural networks for accelerated MRI: history-cognizant unrolling of optimization algorithms. *IEEE J. Sel. Topics Signal Process.*, 14(6):1280–1291, Oct. 2020.
- [34] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo. Deep learning for undersampled MRI reconstruction. *Phys. Med. Biol.*, 63(13), 2018.
- [35] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir. Robust compressed sensing MRI with deep generative priors. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 14938–14954, 2021.
- [36] A. Karan, K. Shah, S. Chen, and Y. C. Eldar. Unrolled denoising networks provably learn to perform optimal Bayesian inference. In *Proc. Adv. Neural Inf. Process. Syst.*, 2024.
- [37] F. Knoll, K. Hammernik, C. Zhang, S. Moeller, T. Pock, D. K. Sodickson, and M. Akçakaya. Deep-learning methods for parallel magnetic resonance imaging reconstruction: A survey of the current approaches, trends, and issues. *IEEE Signal Process. Mag.*, 37(1):128–140, 2020.
- [38] F. Knoll, T. Murrell, A. Sriram, N. Yakubova, J. Zbontar, M. Rabbat, A. Defazio, M. J. Muckley, D. K. Sodickson, C. L. Zitnick, et al. Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge. *Magn. Reson. Med.*, 84(6):3054–3070, 2020.
- [39] F. Knoll, J. Zbontar, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, et al. fastMRI: a publicly available raw k-space and DICOM dataset of knee images for accelerated MR image reconstruction using machine learning. *Radiol. Artif. Intell.*, 2(1), Jan. 2020. Art. no. e190007.
- [40] D. Liang, J. Cheng, Z. Ke, and L. Ying. Deep magnetic resonance image reconstruction: Inverse problems meet neural networks. *IEEE Signal Process. Mag.*, 37(1):141–151, 2020.
- [41] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.*, 58(6):1182–1195, Dec. 2007.

- [42] A. Manoel, F. Krzakala, G. Varoquaux, B. Thirion, and L. Zdeborová. Approximate message-passing for convex optimization with non-separable penalties, 2018. arXiv:1809.06304.
- [43] M. Mardani, Q. Sun, D. Donoho, V. Pappas, H. Monajemi, S. Vasanawala, and J. Pauly. Neural proximal gradient descent for compressive imaging. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 9573–9583, 2018.
- [44] S. Mei. U-Nets as belief propagation: Efficient classification, denoising, and diffusion in generative hierarchical models. In *Proc. Int. Conf. Learn. Represent.*, 2025.
- [45] C. Metzler, A. Mousavi, and R. Baraniuk. Learned D-AMP: Principled neural network based compressive image recovery. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1770–1781, 2017.
- [46] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.*, 38(2):18–44, 2021.
- [47] M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, et al. Results of the 2020 fastMRI challenge for machine learning MR image reconstruction. *IEEE Trans. Med. Imag.*, 40(9):2306–2317, 2021.
- [48] M. Oppor, O. Winther, and M. J. Jordan. Expectation consistent approximate inference. *J. Mach. Learn. Res.*, 6(12), 2005.
- [49] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proc. AAAI Conf. Artif. Intell.*, pages 3942–3951, 2018.
- [50] Z. Ramzi, G. Chaithya, J.-L. Starck, and P. Ciuciu. NC-PDNet: A density-compensated unrolled network for 2D and 3D non-cartesian MRI reconstruction. *IEEE Trans. Med. Imag.*, 41(7):1625–1638, Jul. 2022.
- [51] S. Rangan. Generalized approximate message passing for estimation with random linear mixing. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2168–2172, 2011.
- [52] S. Rangan, P. Schniter, and A. K. Fletcher. Vector approximate message passing. *IEEE Trans. Inf. Theory*, 65(10):6664–6684, 2019.
- [53] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, pages 234–241, 2015.
- [54] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Trans. Med. Imag.*, 37(2):491–503, 2018.
- [55] M. Seeger. Expectation propagation for exponential families. Technical report, Univ of California at Berkeley, 2005.
- [56] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 11918–11930, 2019.
- [57] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Proc. Int. Conf. Learn. Represent.*, 2021.
- [58] J. Sun, H. Li, Z. Xu, et al. Deep ADMM-Net for compressive sensing MRI. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 10–18, 2016.
- [59] R. Timofte, E. Agustsson, L. V. Gool, M.-H. Yang, and L. Zhang. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. Workshop*, pages 114–125, 2017.
- [60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 6000–6010, 2017.
- [61] Y. Wu and K. He. Group normalization. In *Proc. Eur. Conf. Comput. Vis.*, pages 3–19, 2018.
- [62] B. Yaman, H. Gu, S. A. H. Hosseini, Ö. B. Demirel, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya. Multi-mask self-supervised learning for physics-guided neural networks in highly accelerated magnetic resonance imaging. *NMR Biomed.*, 35(12), 2022. Art. no. e4798.
- [63] B. Yaman, S. A. H. Hosseini, and M. Akçakaya. Zero-shot self-supervised learning for MRI reconstruction. In *Proc. Int. Conf. Learn. Represent.*, 2022.

- [64] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. *Magn. Reson. Med.*, 84(6):3172–3191, Dec. 2020.
- [65] B. Yaman, C. Shenoy, Z. Deng, S. Moeller, H. El-Rewaify, R. Nezafat, and M. Akçakaya. Self-supervised physics-guided deep learning reconstruction for high-resolution 3D LGE CMR. In *Proc. IEEE Int. Symp. Biomed. Imag.*, pages 100–104, 2021.
- [66] Y. Yang, J. Sun, H. Li, and Z. Xu. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(3):521–538, Mar. 2020.
- [67] G. Yiasemis, N. Moriakov, J.-J. Sonke, and J. Teuwen. vsharp: Variable Splitting Half-quadratic ADMM algorithm for reconstruction of inverse-problems. *Magn. Reson. Med.*, 115:110266, 2025.
- [68] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno, et al. fastMRI: An open dataset and benchmarks for accelerated MRI, 2019. arXiv:1811.08839v2.
- [69] J. Zhang and B. Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pages 1828–1837, 2018.
- [70] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Trans. Image Process.*, 30:1487–1500, 2020.
- [71] R. Zhao, B. Yaman, Y. Zhang, R. Stewart, A. Dixon, F. Knoll, Z. Huang, Y. W. Lui, M. S. Hansen, and M. P. Lungren. fastMRI+, clinical pathology annotations for knee and brain fully sampled magnetic resonance imaging data. *Scientific Data*, 9(1):152, 2022.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The claims we made in abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We have provided a detailed discussion of our study's limitations in Appendix Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: All assumptions and proofs can be found in Section 3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided a full description of the algorithms used in the paper, with detailed explanations included in Section 4 and the Appendix Section A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Data for our retrospective studies are openly available and details to reproduce the main experimental results are provided in Section 4.1, Section 4.2 and Appendix Section A. The code is available publicly at <https://github.com/JN-Yun/TE-Unrolling-MRI>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental details are provided in the Appendix Section A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars/standard deviations for the quantitative metrics are reported in the Appendix Section H.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All computational resources are reported in Appendix Section A.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research presented in this paper fully complies with the NeurIPS Code of Ethics in all respects.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed the potential positive societal impacts of this work in Section 1 and Section 4. We believe our work does not pose any negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All codes and datasets used in this paper have been properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: This paper does not involve crowdsourcing nor research with human subject.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: Knee and brain imaging data was obtained from the publicly available NYU fastMRI dataset [39, 68], which was collected with IRB approval and subject consent as detailed in the original publication. No additional data involving human subjects were collected in this study.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This work does not involve the use of LLMs in any core method or experimental component.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A Model Architectures and Relevant Hyperparameters

Neural Network Architectures. A ResNet and a U-Net architecture was used for proximal operators in the unrolled networks, as illustrated in Fig. 2 (c). The ResNet used for the proximal operator [59] consists of 15 residual blocks, each containing 3×3 convolutional layers with ReLU activation and a scaling term. The scaling term is set to 1×10^{-1} [64]. The U-Net used for the proximal operator, which is designed based on ADM from [21] with slight modifications, has 2 downsampling layers, 2 upsampling layers, and a bottleneck. It uses residual blocks with 3×3 convolutional layers, normalization, and SiLU activation. The initial channel size is 32, which doubles during downsampling and is recovered during upsampling. For time-embedded architectures, time-embedded features are injected and modulated through group normalization and the FiLM method in each residual block, as shown in Fig. 2 (b) and (c). All training processes are conducted using one NVIDIA A100-SXM4-40GB GPU.

Shared/Unshared Baseline. For the data fidelity term, we use a shared μ initialized to 5×10^{-2} in VSQP and 1.5×10^{-2} in ADMM. In ADMM, we also initialized the dual update parameter λ to 1×10^{-1} . For the shared baseline networks, both a ResNet and U-Net proximal operator as described above were used, without time-embedding features. In the unshared case, there is a separate proximal operator for each unroll with no weight sharing or time-embedding. We train ResNet for 100 epochs and U-Net for 50 epochs, using a learning rate of 5×10^{-4} for coronal PD/PD-FS knee data and 2×10^{-4} for Axial T2 brain data with the Adam optimizer.

Time Embedded Unrolled Networks. For our proposed time-embedded unrolled networks, as described in Section 3.1 and Section 3.2, we utilized time-dependent data fidelity, Onsager correction parameters, and time-embedded proximal operators. In particular, the data fidelity scalars μ^t were initialized to 1.5×10^{-2} , and the time-dependent Onsager correction parameters ρ^t to 1×10^{-1} . For time-embedded neural networks, we apply the same optimization strategies as in the baseline, except for ResNet in the coronal PD-FS case, where we use a learning rate of 1.8×10^{-2} . The scaling factor τ of FiLM in ResNet is set to 0.1. To extend our approach to other unrolling algorithms described in Section 4.4, we replace the shared data fidelity term μ with the unshared data fidelity term μ^t and utilize the same proximal operators as in our proposed methods.

Generalization to Diverse Datasets Our method incorporates a time-embedding module, which introduces additional hyperparameters. These include the frequency of the sinusoidal encoding, the embedding dimension, and the number of hidden channels in the MLP layers that process the time embeddings. We use the same configuration across different datasets (Coronal PD, Coronal PD-FS, and Axial T2 brain), demonstrating the

Table 4: Comparison of the results from the fine-tuned **Unshared** (ADMM) method with those of shared and unshared baselines trained from scratch in *limited data* settings. **FS**: From Scratch; **FT**: Fine-Tuning. Quantitative results are reported across three datasets with varying undersampling patterns. The best values are highlighted in **bold**.

		U-Net								ResNet						
		FS (Shared)		FS (Unshared)	FT (Unshared)				FS (Shared)		FS (Unshared)	FT (Unshared)				
		Epoch	100	100	10	20	30	40	50	100	100	10	20	30	40	50
Coronal PD	$\times 4$	PSNR $\uparrow$	40.76	40.51	40.96	40.89	40.82	40.78	40.69	41.27	41.11	41.45	41.37	41.28	41.25	41.18
		SSIM $\uparrow$	0.964	0.963	0.964	0.964	0.964	0.963	0.963	0.965	0.964	0.965	0.965	0.965	0.964	0.964
	$\times 6$	PSNR $\uparrow$	38.85	38.52	39.13	39.13	39.09	38.99	38.94	39.61	39.60	39.88	39.78	39.72	39.72	39.64
		SSIM $\uparrow$	0.950	0.949	0.952	0.952	0.952	0.951	0.951	0.953	0.953	0.955	0.955	0.955	0.955	0.953
	$\times 8$	PSNR $\uparrow$	36.31	35.71	36.51	36.29	36.29	36.25	36.11	36.72	36.41	36.97	36.93	36.76	36.74	36.63
		SSIM $\uparrow$	0.924	0.917	0.925	0.923	0.923	0.922	0.920	0.926	0.921	0.927	0.927	0.925	0.925	0.923
Coronal PD-FS	$\times 4$	PSNR $\uparrow$	35.31	35.23	35.44	35.32	35.22	35.12	35.01	35.37	35.23	35.59	35.57	35.54	35.51	35.50
		SSIM $\uparrow$	0.851	0.848	0.850	0.849	0.847	0.844	0.841	0.848	0.849	0.850	0.851	0.852	0.852	0.852
	$\times 6$	PSNR $\uparrow$	34.26	34.27	34.45	34.33	34.26	34.17	34.11	34.53	34.33	34.69	34.67	34.64	34.61	34.58
		SSIM $\uparrow$	0.821	0.824	0.822	0.822	0.820	0.819	0.819	0.822	0.823	0.823	0.823	0.825	0.825	0.825
	$\times 8$	PSNR $\uparrow$	33.21	33.06	33.34	33.18	33.05	32.96	32.92	33.35	33.09	33.61	33.57	33.53	33.44	33.38
		SSIM $\uparrow$	0.795	0.796	0.795	0.792	0.790	0.788	0.786	0.796	0.789	0.797	0.767	0.797	0.795	0.795
Axial T2	$\times 4$	PSNR $\uparrow$	36.60	36.54	36.67	36.67	36.60	36.67	36.62	36.81	36.75	36.87	36.83	36.86	36.83	36.85
		SSIM $\uparrow$	0.928	0.928	0.927	0.927	0.927	0.927	0.927	0.925	0.926	0.924	0.925	0.925	0.925	0.925
	$\times 6$	PSNR $\uparrow$	35.05	34.91	35.16	35.16	35.15	35.06	35.10	35.35	35.10	35.41	35.41	35.47	35.44	35.33
		SSIM $\uparrow$	0.910	0.910	0.910	0.911	0.910	0.911	0.910	0.910	0.909	0.907	0.908	0.909	0.908	0.907
	$\times 8$	PSNR $\uparrow$	33.41	32.98	33.52	33.40	33.36	33.39	33.42	33.43	33.14	33.91	33.93	33.90	33.83	33.78
		SSIM $\uparrow$	0.893	0.890	0.893	0.892	0.892	0.892	0.892	0.890	0.889	0.890	0.890	0.890	0.890	0.890

robustness of our approach to diverse data. Similarly, we apply identical hyperparameters to the neural networks used as proximal operators across all datasets, on which the networks consistently perform well.

B Fine-tuned Unshared Methods in Limited Data Settings

To mitigate the overfitting tendency of unshared networks in limited data settings, we explore fine-tuning instead of training from scratch. Specifically, We initialize the unshared baseline unrolled network from the pre-trained shared baseline unrolled network. This unshared unrolled network is then fine-tuned for several epochs with a learning rate of 1×10^{-4} for both ResNet and U-Net, and for knee and brain data comprising 300 slices each. As shown in Tab. 4, the fine-tuned unshared methods generally outperform both shared and unshared methods trained from scratch. However, overfitting remains evident in the fine-tuned models as the number of training epochs increases under limited data conditions. Furthermore, although the fine-tuned unshared methods improve PSNR and SSIM scores, they still struggle to suppress artifacts over iterations (see Fig. 8 for visual examples).

C Analysis of the Onsager Correction Term in Algorithm 1

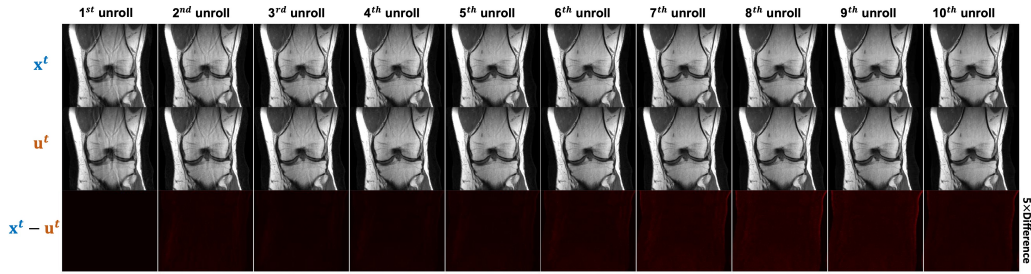


Figure 5: The intermediate visual results of the proposed method with ResNet-TE proximal operator at each iteration in a coronal PD slice.

Table 5: Average normalized mean squared error between $\mathbf{x}^t$ and $\mathbf{u}^t$ at each iteration of the unrolled network with ResNet-TE on the coronal PD test set ($R = 4$).

Iteration	1	2	3	4	5	6	7	8	9	10
Normalized MSE	1.01×10^{-9}	5.46×10^{-3}	9.02×10^{-3}	3.77×10^{-3}	6.75×10^{-3}	1.26×10^{-2}	2.55×10^{-2}	3.35×10^{-2}	2.68×10^{-2}	9.96×10^{-3}

To explore the effect of the Onsager correction in Step 4 of Algorithm 1, we evaluated whether the intermediate updates in the network satisfy $\mathbf{x}^t \approx \mathbf{u}^t$. We compared the outputs of the data fidelity, ($\mathbf{x}^t$) and its Onsager correction term output, ($\mathbf{u}^t$) across unrolls, which is shown in Fig. 5. The bottom row shows the scaled ($\times 5$) difference between them, which is minimal upon visual inspection. We further quantified this difference by calculating the normalized mean squared error between $\mathbf{x}^t$ and $\mathbf{u}^t$ at each iteration. Tab. 5 shows that the difference ranges from 1.01×10^{-9} to 3.35×10^{-2} , indicating no substantial variation.

D Qualitative Comparison of the Baseline and Time-Embedded Algorithm Unrolling

As discussed in Section 4.4 and in view of Section C, our time-embedding approach can be extended to other unrolled algorithms (VSQP and ADMM) by incorporating a time-embedding module into the proximal operators of these unrolled networks. The time-embedded versions of the unrolled algorithms are given below. For time-embedded VSQP:

$$\mathbf{x}^t = \left(\mathbf{E}_\Omega^H \mathbf{E}_\Omega + \mu^t \mathbf{I} \right)^{-1} \left(\mathbf{E}_\Omega^H \mathbf{y}_\Omega + \mu^t \mathbf{z}^t \right), \quad (17)$$

$$\mathbf{z}^{t+1} = \text{prox}_{\mathcal{R}}(\mathbf{x}^t, \alpha^t, \beta^t, t), \quad (18)$$

For time-embedded ADMM:

$$\mathbf{x}^{t+1} = \left(\mathbf{E}_\Omega^H \mathbf{E}_\Omega + \mu^t \mathbf{I} \right)^{-1} \left(\mathbf{E}_\Omega^H \mathbf{y}_\Omega + \mu^t (\mathbf{z}^t - \mathbf{u}^t) \right), \quad (19)$$

$$\mathbf{z}^{t+1} = \text{prox}_{\mathcal{R}}(\mathbf{x}^{t+1} + \mathbf{u}^t, \alpha^t, \beta^t, t), \quad (20)$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \lambda(\mathbf{x}^{t+1} - \mathbf{z}^{t+1}), \quad (21)$$

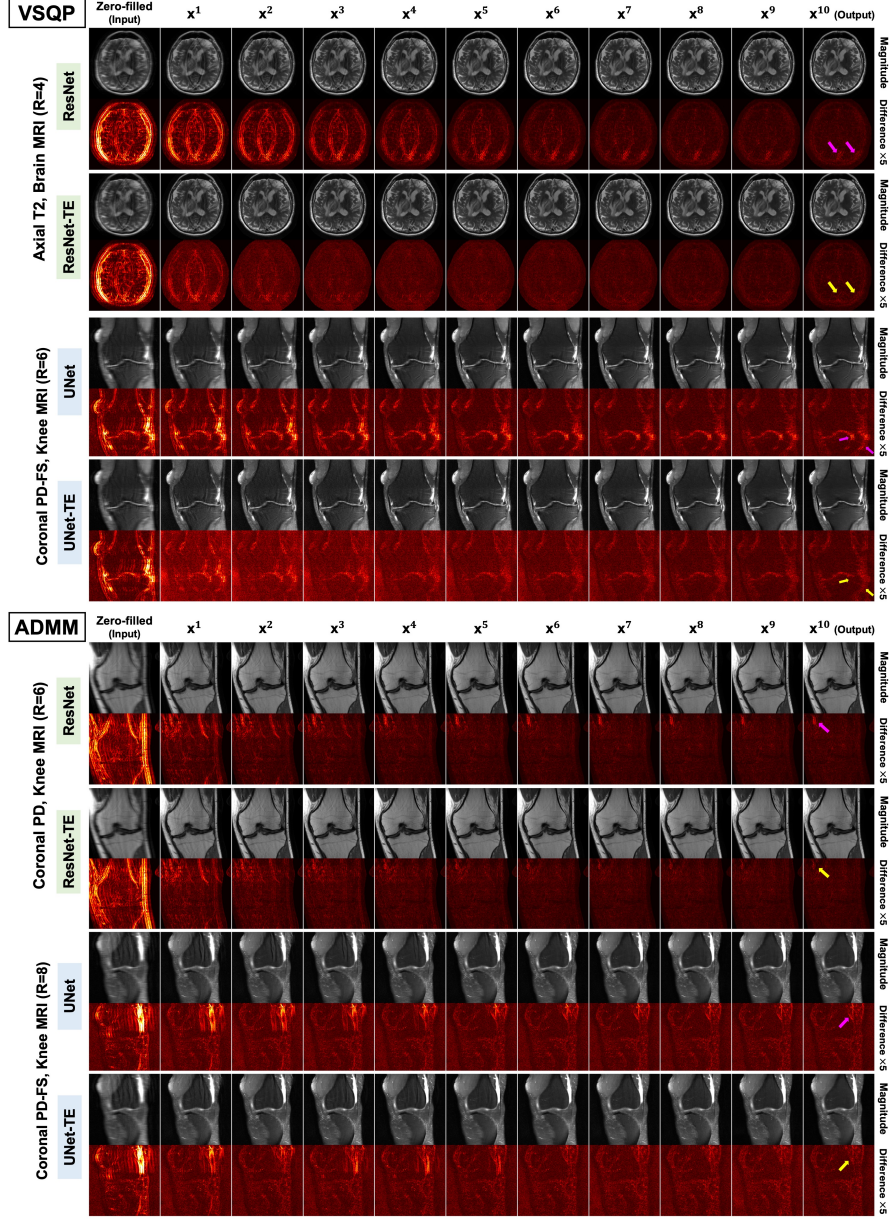


Figure 6: The estimates through different stages of the unrolled network with shared baseline VSQP, ADMM, and the time-embedded VSQP, ADMM for different datasets with varying acceleration rates. The error maps illustrate the differences between the intermediate estimates, $\{\mathbf{x}^t\}_{t=1}^T$, and the reference. The shared baselines, utilizing both U-Net and ResNet proximal operators, exhibit persistent errors across unrolls, which are highlighted with pink arrows in the last unroll, and can be visualized at the same location in prior unrolls. In contrast, the time-embedded networks, with proximal operators U-Net-TE and ResNet-TE, effectively reduce noise (yellow arrows).

where the data fidelity parameter μ and the proximity operator $\text{prox}_{\mathcal{R}}(\cdot)$ are replaced with time-dependent parameters μ^t and networks $\text{prox}_{\mathcal{R}}(\cdot, \alpha^t, \beta^t, t)$, respectively.

Fig. 6 presents examples of how the reconstruction evolves when comparing shared VSQP and ADMM with its time-embedded counterparts. The shared methods exhibit persistent errors over iterations that the proximal operator fails to eliminate. However, integrating our proposed time-embedding methods into the baseline algorithms effectively addresses these issues, demonstrating gradual denoising, as intended.

Table 6: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Quantitative results on the coronal PD datasets using equispaced undersampling patterns at $R = 4, 6$, and 8 with **5 and 15 unrolls**. The **best** and **second-best** values for each architecture are highlighted along with their relative difference to **10 unrolls**.

		U-Net					ResNet				
	R	VS (♣)		ADMM (♣)		Ours	VS (♣)		ADMM (♣)		Ours
		VS (♠)	(Fine-tuned)	ADMM (♠)	(Fine-tuned)		VS (♠)	(Fine-tuned)	ADMM (♠)	(Fine-tuned)	
5 unrolls	$\times 4$	PSNR $\uparrow$	40.58	40.71	40.86	40.93	40.66	41.19	41.16	41.38	41.41
		SSIM $\uparrow$	0.963	0.963	0.964	0.964	0.963	0.963	0.964	0.965	0.966
	$\times 6$	PSNR $\uparrow$	38.01	38.57	38.49	38.92	39.25	39.57	39.44	39.76	39.65
		SSIM $\uparrow$	0.943	0.948	0.947	0.951	0.952	0.952	0.954	0.955	0.954
	$\times 8$	PSNR $\uparrow$	35.63	35.81	35.79	35.99	35.94	36.44	36.40	36.65	36.76
		SSIM $\uparrow$	0.916	0.918	0.917	0.920	0.919	0.922	0.924	0.925	0.925
15 unrolls	$\times 4$	PSNR $\uparrow$	40.36	40.84	40.59	41.01	41.16	41.36	41.44	41.48	41.52
		SSIM $\uparrow$	0.962	0.964	0.963	0.965	0.964	0.964	0.966	0.965	0.966
	$\times 6$	PSNR $\uparrow$	38.08	38.77	38.74	39.01	39.61	39.78	39.61	39.80	39.83
		SSIM $\uparrow$	0.944	0.950	0.948	0.952	0.953	0.952	0.952	0.954	0.954
	$\times 8$	PSNR $\uparrow$	35.44	35.97	36.25	36.50	36.41	36.87	36.95	37.09	37.09
		SSIM $\uparrow$	0.908	0.918	0.923	0.926	0.923	0.926	0.928	0.927	0.929

Table 7: The comparison results for different model sizes ($T = 10$, $R = 4$, and Coronal PD).

	Method	Channel	# Param.	PSNR $\uparrow$	SSIM $\uparrow$
ResNet-VSQP	Shared $\mathcal{R}$	64	592,129	41.11	0.965
	Unshared $\mathcal{R}$	64	5,921,281	40.99	0.963
	Shared $\mathcal{R}$	96	1,330,561	41.09	0.965
	Unshared $\mathcal{R}$	96	13,305,601	41.00	0.963
	Ours ($T = 5$)	64	866,571	41.41	0.966
	Ours ($T = 10$)	64	866,581	41.43	0.965
UNet-VSQP	Method	Channel	# Param.	PSNR $\uparrow$	SSIM $\uparrow$
	Shared $\mathcal{R}$	[32, 64, 128]	1,724,035	40.50	0.962
	Unshared $\mathcal{R}$	[32, 64, 128]	17,240,341	40.31	0.960
	Shared $\mathcal{R}$	[64, 128, 256]	6,878,467	40.77	0.964
	Unshared $\mathcal{R}$	[64, 128, 256]	68,784,661	40.55	0.962
	Ours ($T = 5$)	[32, 64, 128]	1,963,469	40.94	0.964
	Ours ($T = 10$)	[32, 64, 128]	1,963,479	40.99	0.964

Table 8: The comparison results for time-embedded unrolling networks with different time-embedding hyperparameters ($T = 10$, $R = 6$, and Coronal PDFS).

Freq.	Emb. dim.	Hidden layer dim.	PSNR $\uparrow$	SSIM $\uparrow$
1,000	32	128	34.34	0.824
5,000			34.35	0.822
10,000			34.44	0.825
10,000	32	128	34.44	0.825
	64		34.32	0.824
	96		34.34	0.824
10,000	32	64	34.38	0.824
		128	34.44	0.825
		196	34.37	0.824

E Additional Details on the Ablation Studies

This section presents further implementation details and results for the experiments described in Section 4.5.

Robust Time-Embedded Unrolling with Different Numbers of Unrolls Time-embedding denoisers can recognize temporal sequence information, allowing them to adaptively apply varying degrees of denoising at different stages, as shown in Fig. 6. Based on these observations, we hypothesized that our proposed method can achieve stable performance even with a reduced or increased number of unrolling iterations. We compared our approach with both shared and unshared unrolling methods, where each was trained with $T = 5$ and 15 unrolls. In this experiment, the unshared networks were fine-tuned to improve performance; for a detailed rationale, please refer to Appendix B. Tab. 6 presents quantitative reconstruction results for $T = 5$ and 15 unrolls. With fewer iterations ($T = 5$ unrolls), our approach exhibits greater flexibility and robustness compared to the shared baseline algorithms, which experience performance degradation as the number of unrolls decreases. Notably, for $R = 8$, both shared and unshared baselines for each architecture show significant PSNR degradation when using $T = 5$ unrolls. In contrast, our proposed method maintains performance even with fewer iterations, showing either a slight improvement or only minimal degradation compared to $T = 10$ unrolls, depending on the choice of the proximal operator architecture. Moreover, as the number of iterations are increased ($T = 15$ unrolls), our proposed method maintains its robustness and consistently improves performance against the baseline models with shared $\mathcal{R}(\cdot)$, while introducing only a minimal increase in computational complexity.

The qualitative results in Fig. 9 and Fig. 10 for $T = 5$ and 15 unroll iterations, respectively, support these quantitative observations. Our proposed method effectively reduces artifacts and enhances image sharpness, while the shared and unshared baseline models struggle to achieve similar improvements, both with fewer and increased iterations.

Table 9: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Quantitative results are reported on the Coronal PD, Coronal PD-FS, and axial T2 datasets, with non-uniform undersampling patterns at acceleration rates $R = 4, 6$, and 8 . The **best** result for each architecture are highlighted.

R		VSQP (♠)	VSQP (♣)	ADMM (♠)	ADMM (♣)	Ours	
Cor. PD	×4	PSNR↑	40.10	40.26	40.20	40.13	40.43
		SSIM↑	0.961	0.961	0.961	0.961	0.962
	×6	PSNR↑	38.41	38.59	38.64	38.72	38.73
		SSIM↑	0.947	0.947	0.948	0.949	0.949
	×8	PSNR↑	37.30	37.39	37.64	37.52	37.76
		SSIM↑	0.935	0.934	0.939	0.938	0.938
R		VSQP (♠)	VSQP (♣)	ADMM (♠)	ADMM (♣)	Ours	
Cor. PDFS	×4	PSNR↑	35.61	35.60	35.62	35.66	35.68
		SSIM↑	0.855	0.853	0.853	0.856	0.854
	×6	PSNR↑	34.59	34.55	34.70	34.61	34.74
		SSIM↑	0.828	0.824	0.827	0.827	0.830
	×8	PSNR↑	33.76	33.97	34.06	34.07	34.14
		SSIM↑	0.809	0.808	0.810	0.810	0.812
R		VSQP (♠)	VSQP (♣)	ADMM (♠)	ADMM (♣)	Ours	
Axial T2	×4	PSNR↑	35.90	36.36	36.50	36.42	36.52
		SSIM↑	0.932	0.931	0.930	0.931	0.930
	×6	PSNR↑	35.03	35.11	35.26	35.25	35.21
		SSIM↑	0.917	0.916	0.915	0.905	0.914
	×8	PSNR↑	34.09	34.31	34.37	34.39	34.51
		SSIM↑	0.908	0.908	0.903	0.906	0.904

Efficiency Relative to the Number of Parameters and Time-Embedding Hyperparameters To assess efficiency with respect to the number of parameters, we explored the effect of increasing the number of parameters on performance, which resulted in higher total parameter counts in both the shared and unshared baselines. As shown in Tab. 7, we use the following setups: (1) increasing the channels in ResNet residual blocks from 64 to 96, (2) increasing the channels in U-Net up/downsampling blocks from [32, 64, 128] to [64, 128, 256], and (3) using $T = 10$, $R = 4$, with the Coronal PD dataset.

For efficiency relative to the time-embedding hyperparameters, we evaluated (1) the frequency of the sinusoidal encoding, (2) the embedding dimension, and (3) the number of hidden channels in the MLP layers. The experiments were conducted using a U-Net architecture with $T = 10$ and $R = 6$ on the Coronal PDFS dataset. Implementation details and results are provided in Tab. 8.

F Details on the Extended Experiments

This section provides additional implementation details and results for the experiments described in Section 4.6.

Experiments on Non-Uniform Undersampling Masks As shown in Tab. 9, our method consistently outperforms the baselines in terms of PSNR in all cases except for the Axial T2 dataset at $R=6$. For SSIM, our method shows improvement in most cases for the PD and PD-FS datasets, although no improvement is observed for the Axial T2 dataset.

Table 10: The comparison results with diffusion-based model (DDS).

Method	Data	R	PSNR $\uparrow$	SSIM $\uparrow$
DDS (100)	PD	$\times 4$	37.41 ± 3.25	0.940 ± 0.029
Ours (U-Net)	PD	$\times 4$	40.09 ± 2.51	0.958 ± 0.017

Comparison with Diffusion Model-Based Reconstruction Since DDS requires 320×320 inputs due to its generative pre-trained prior, we additionally evaluated PSNR and SSIM using the central 320×320 region. Note that this differs from the results reported in Tab. 1, where evaluations were performed on images of size 320×368 , aligned with the original raw k-space data. We set the number of sampling steps to 100 for DDS. For our method, we used $T = 10$ unrolls. All experiments were conducted on the Coronal PD test dataset with an acceleration factor of $R = 4$. As shown in Tab. 10, our method outperforms DDS in both PSNR and SSIM. Furthermore, diffusion-based reconstruction requires tens to hundreds of neural function evaluations (NFEs) during inference [16], which remains far from the efficiency needed for large-scale or real-time applications. In contrast, our time-embedded unrolled networks achieve more promising results with substantially fewer NFEs (e.g., 5–10), even with a smaller network architecture compared to diffusion-based models.

G Additional Qualitative Results

Pathological Region Inspection Using fastMRI+ We leveraged the annotations of pathological regions provided by fastMRI+ [71] to further validate the strengths of our method. As shown in Fig. 7, our approach produces clearer contrast in the pathological regions compared to other methods, which is further corroborated with radiologist assessments, as detailed next.

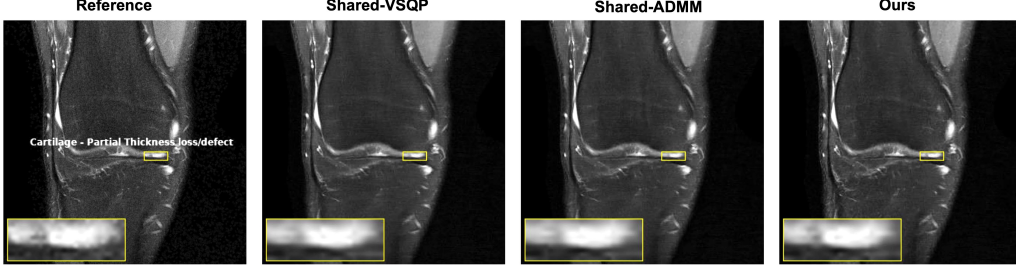


Figure 7: Reconstruction results with annotated pathological regions.

Radiologist Readings For a subset of all the data processed in this study, a musculoskeletal radiologist with over 30 years of experience blindly reviewed the reconstructed images from the different methods. The radiologist’s assessments highlight improvements achieved by our method that are critical for diagnostic purposes. Details are provided below.

- Fig. 3 (Middle) exhibits aliasing artifacts in the distal femoral metaphysis medially on PD-weighted images for VSQP, Unshared-VSQP, ADMM, Unshared-ADMMs. The artifacts are effectively removed in the proposed method (ours).
- Fig. 3 (Bottom) shows blurring in the right occipital lobe in VSQP, Unshared-VSQP, ADMM, Unshared-ADMM. This is notably improved in the proposed method, where the gyri and sulci appear sharper.
- Fig. 4 (Top) shows visible aliasing artifacts in the central aspect of the distal femoral condyle of the inset for VSQP, Unshared-VSQP, ADMM, and Unshared-ADMM. This artifact is removed in the proposed method. The prominent penetrating intraosseous vessel at the lateral aspect of the proximal tibia (lower left on the image) appears sharper in proposed method, though overall image sharpness is similar among methods.
- Fig. 4 (Middle): There is an oblong, hypointense appearing aliasing artifact on the non-fat saturated PD-weighted images of the knee joint, just distal to the posteromedial femoral condyle, seen on VSQP, Unshared-VSQP, ADMM, and Unshared-ADMM, when compared to the reference. This artifact is removed from the image for the proposed method. Thus, only the proposed method accurately resembles the reference image.
- Fig. 4 (Bottom) reveals aliasing artifacts in VSQP, Unshared-VSQP, ADMM, Unshared-ADMM, depicted as curvilinear, oblique hypointense signal in the occipital lobe. The artifact is nearly completely removed in Unshared-ADMM, and only vaguely seen. The artifact is completely removed in ours, most accurately resembling the reference image.
- In Fig. 7, on the reference image there is a focal area of T2-hyperintense signal, most consistent with a partial/full thickness cartilage defect in the anteromedial trochlea. Images reconstructed by VSQP, Unshared-VSQP, ADMM, and Unshared-ADMM reveal significant blurring in this area. Proposed method shows the least amount of blurring compared to other methods, and shows the hyperintense region in the trochlear articular cartilage with the most fidelity compared to the reference data.

Additional Qualitative Examples We provide additional representative reconstruction examples that demonstrate the visual superiority of our proposed method, since PSNR/SSIM do not necessarily align with perception, as discussed in Section 4.3. Fig. 11, Fig. 12, and Fig. 13 shows reconstruction results across all datasets and proximal operator architectures for $R = 4, 6$, and 8 , respectively, using the implementations described in the main text.

H Extended Quantitative Results with Standard Deviation

Tab. 11 summarizes the standard deviation of PSNR and SSIM for the same settings as in Tab. 1.

Table 11: ♠: Shared $\mathcal{R}(\cdot)$ weights; ♣: Unshared $\mathcal{R}(\cdot)$ weights. Standard deviations of PSNR and SSIM results for the same settings in Tab. 1.

	R	U-Net					ResNet				
		VSQP (♠)	VSQP (♣)	ADMM (♠)	ADMM (♣)	Ours	VSQP (♠)	VSQP (♣)	ADMM (♠)	ADMM (♣)	Ours
Coronal PD	$\times 4$	PSNR	2.56	2.59	2.42	2.41	2.40	2.95	3.04	2.97	3.03
		SSIM	0.02	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.02
	$\times 6$	PSNR	2.10	2.31	2.21	2.10	2.17	2.77	2.81	2.74	2.74
		SSIM	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	$\times 8$	PSNR	2.15	2.29	2.22	2.09	2.00	2.60	2.70	2.66	2.67
		SSIM	0.03	0.04	0.04	0.03	0.03	0.04	0.04	0.04	0.03
Coronal PD-FS	$\times 4$	PSNR	2.73	2.79	2.80	2.77	2.76	2.78	2.80	2.82	2.79
		SSIM	0.10	0.10	0.10	0.10	0.09	0.10	0.10	0.10	0.10
	$\times 6$	PSNR	2.56	2.61	2.59	2.53	2.58	2.69	2.70	2.70	2.68
		SSIM	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
	$\times 8$	PSNR	2.41	2.44	2.43	2.36	2.44	2.46	2.48	2.53	2.54
		SSIM	0.11	0.11	0.11	0.11	0.11	0.12	0.12	0.12	0.12
Axial T2-W	$\times 4$	PSNR	3.01	2.88	2.99	2.91	2.92	3.18	3.17	3.29	3.12
		SSIM	0.06	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.06
	$\times 6$	PSNR	3.05	2.68	2.88	2.79	2.98	3.04	2.99	3.22	3.08
		SSIM	0.08	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07
	$\times 8$	PSNR	2.40	2.35	2.55	2.57	2.61	2.60	2.53	2.77	2.55
		SSIM	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07

I Discussions

Second-Moment Matching. Since our methods are inspired by the VAMP framework, an assessment of second-moment matching for VAMP is desirable. Second-moment matching is typically evaluated by examining the estimated variances v_x^t and v_z^t across iterations, as in (9) and (11). However, even if consistent trends are empirically observed in these estimates, this would not constitute a formal proof. This is further complicated in our case, as we hypothesize that a time-embedded neural network models all update steps in (10)-(11), and the learnable scalar parameter ρ^t encapsulates the entire process described in (9). As a result, v_x^t and v_z^t are embedded within black-box neural modules and are not explicitly accessible.

Instead, to indirectly assess whether second-moment matching holds, we analyzed the relationships between intermediate estimates. Specifically, we examined the empirical differences between $\mathbf{x}^t$ and $\mathbf{u}^t$, and between $\mathbf{u}^t$ and $\mathbf{r}^t$ in Alg. 1, as these pairs are intrinsically related to v_x^t and v_z^t , respectively. If these differences remained consistently small across iterations, it provided evidence that the underlying variance estimates are stable. The empirical difference between $\mathbf{x}^t$ and the reconstructed $\mathbf{u}^t$ was reported in Appendix C. As shown in Tab. 5, the difference (normalized MSE) ranges from 1.01×10^{-9} to 3.35×10^{-2} , demonstrating stable behavior. These findings suggest that second-moment matching is empirically preserved, despite the use of learned components.

Lipschitz Constant or Gradient Explosion/Vanishing of Time-embedding (FiLM) Layers. Consider a network where at each time step $t \in \{1, \dots, T\}$, there are K consecutive layers within the proximal operator networks composed of intermediate transformations followed by FiLM modulation. For each layer $k \in \{1, \dots, K\}$,

$$x^{(t,k)} = \text{FiLM}(f_k(x^{(t,k-1)}), t), \quad (22)$$

where $f_k(\cdot)$ denotes the intermediate layers (e.g., convolution + activation) preceding the FiLM block at layer k . Suppose each intermediate layer is Lipschitz continuous with constant L_k , and satisfies

$$\|f_k(x)\| \leq L_k \|x\| + \delta_k, \quad (23)$$

for some small $\delta_k \geq 0$, allowing for nonzero bias or offset when $f_k(0) \neq 0$. Similarly, each FiLM block is Lipschitz continuous with constant $\Gamma_{t,k}$, satisfying

$$\|\text{FiLM}(z, t)\| \leq \Gamma_{t,k} \|z\| + B_{t,k}. \quad (24)$$

The composite function then satisfies:

$$\|\text{FiLM}(f_k(x), t)\| \leq \Gamma_{t,k} L_k \|x\| + (\Gamma_{t,k} \delta_k + B_{t,k}). \quad (25)$$

Thus, the overall Lipschitz constant of the composite layer at layer k and time t is $\Gamma_{t,k} L_k$. As T grows, if each $\Gamma_{t,k} L_k$ is strictly less than 1, their product decays exponentially, which may cause vanishing activations and hinder learning. If any are ≥ 1 , the product can grow exponentially, causing exploding activations and instability. Thus, controlling cumulative constant, $\prod_{t=1}^T \prod_{k=1}^K \Gamma_{t,k} L_k$ is crucial for stable training.

While a formal proof is not provided, as we do not analytically characterize the Lipschitz constants of individual components, our empirical results support the stability of the proposed approach. In particular, we adopt the

U-Net architecture and FiLM modules commonly used in diffusion denoising tasks. In the diffusion model literature, a large number of diffusion steps ($T \geq 1,000$) is typically employed, which has been shown to promote stable training. In our unrolled networks, where a substantially smaller number of steps is used (*e.g.* $T = 5-15$), we observe that training remains stable, suggesting that the reduced T does not compromise empirical stability. These observations underscore the empirical nature of our Lipschitz constant bounds and their relevance to practical performance.

Qualitative results of fine-tuned unshared networks with 10 unrolls

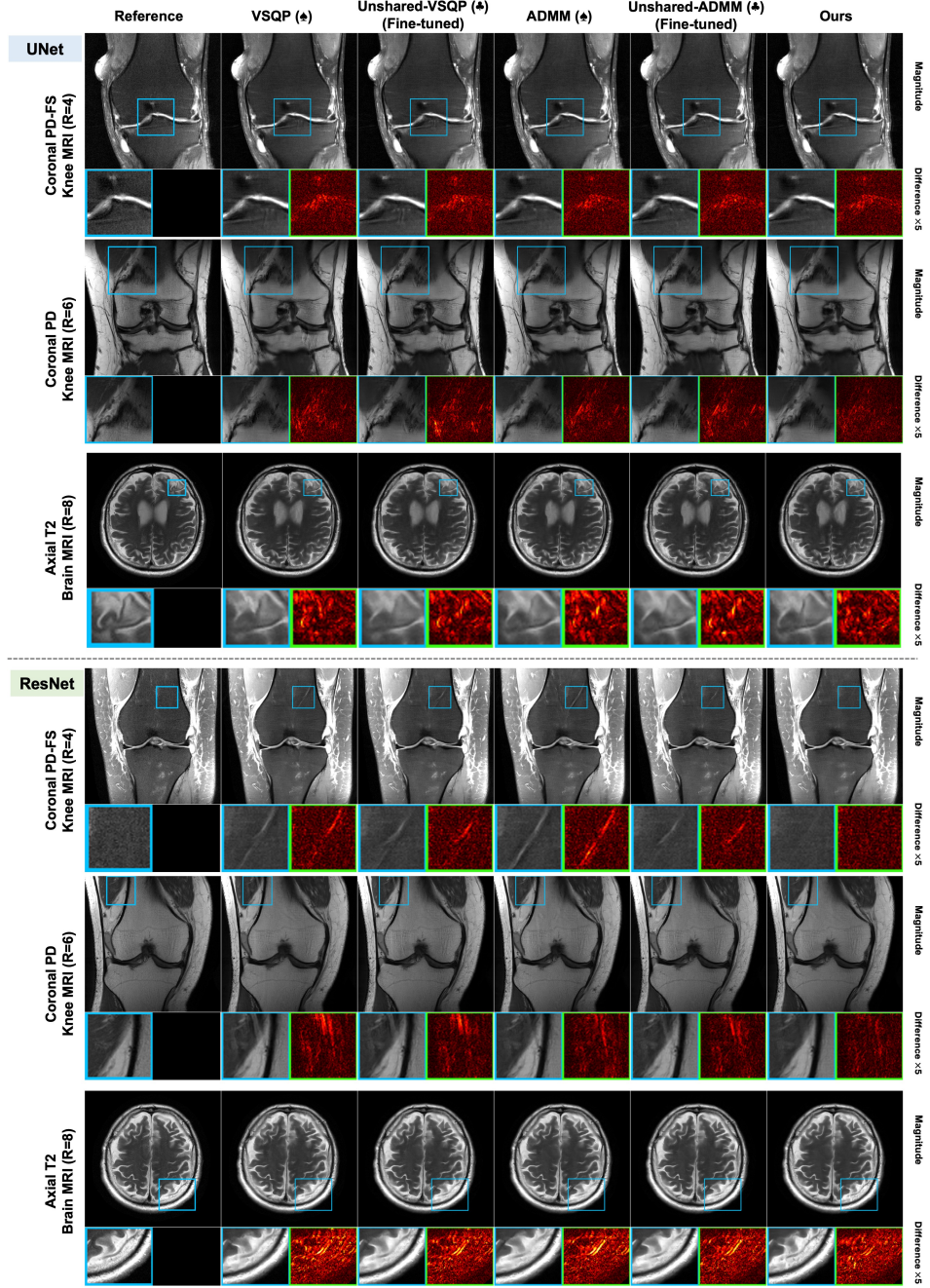


Figure 8: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: **Fine-tuned** Unshared $\mathcal{R}(\cdot)$ weights. Qualitative comparisons of each unrolled network with $T = 10$ unrolls for U-Net and ResNet proximal operators. In each proximal operator, **Top**: Results for $R = 4$ using PD data. **Middle**: Results for $R = 6$ using PD-FS data. **Bottom**: Results for $R = 8$ using Axial T2-W data. The fine-tuned unshared networks still struggle to suppress artifacts over iterations, whereas the proposed methods perform well, effectively reducing artifacts.

Qualitative results of each unrolled networks with **5 unrolls**

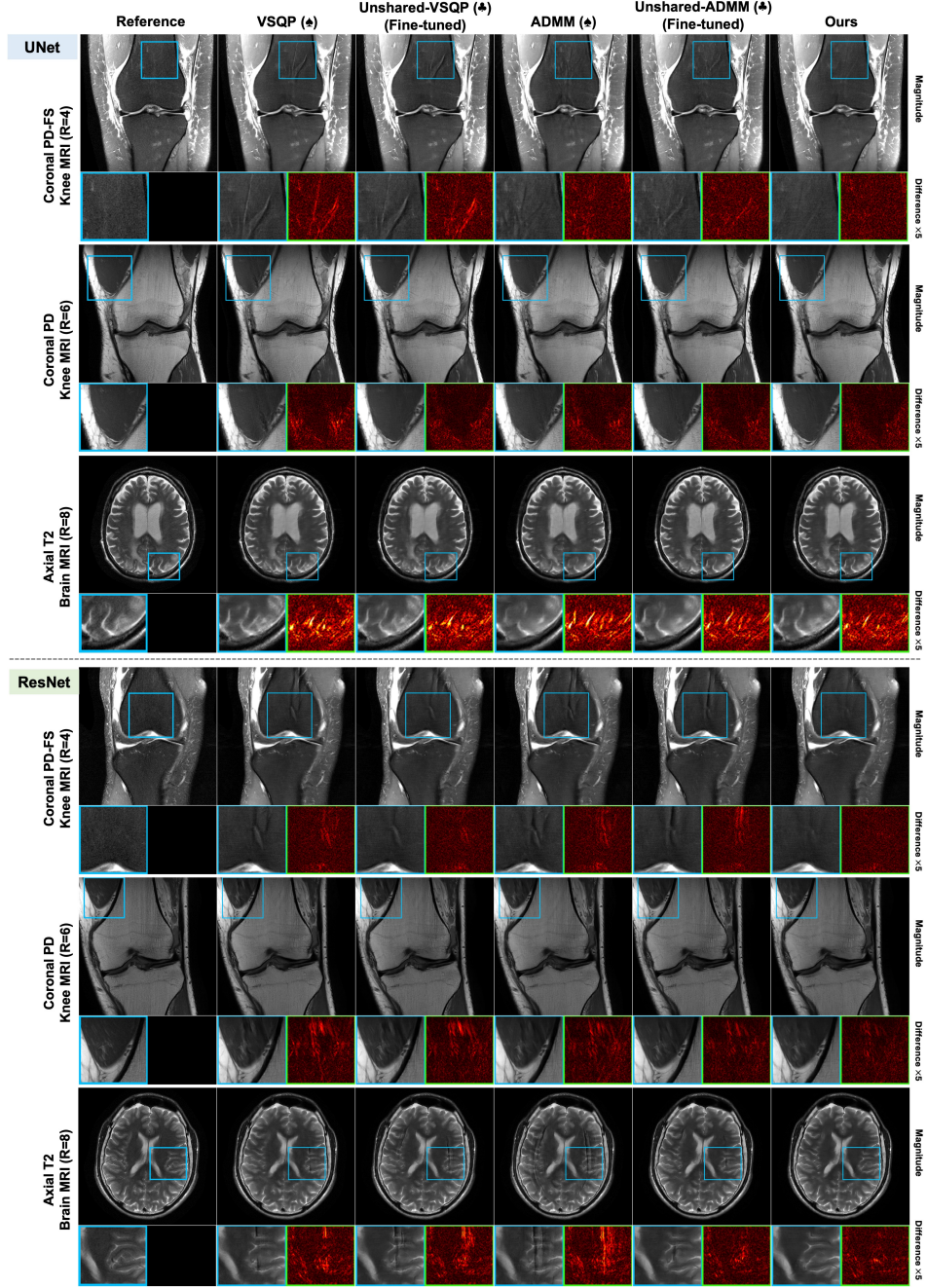


Figure 9: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Qualitative comparisons of each unrolled network with $T=5$ unrolls for U-Net and ResNet proximal operators. In each proximal operator, **Top:** Results for $R=4$ using PD data. **Middle:** Results for $R=6$ using PD-FS data. **Bottom:** Results for $R=8$ using Axial T2-W data. The proposed methods still perform well with fewer iterations, effectively reducing artifacts.

Qualitative results of each unrolled networks with **15 unrolls**

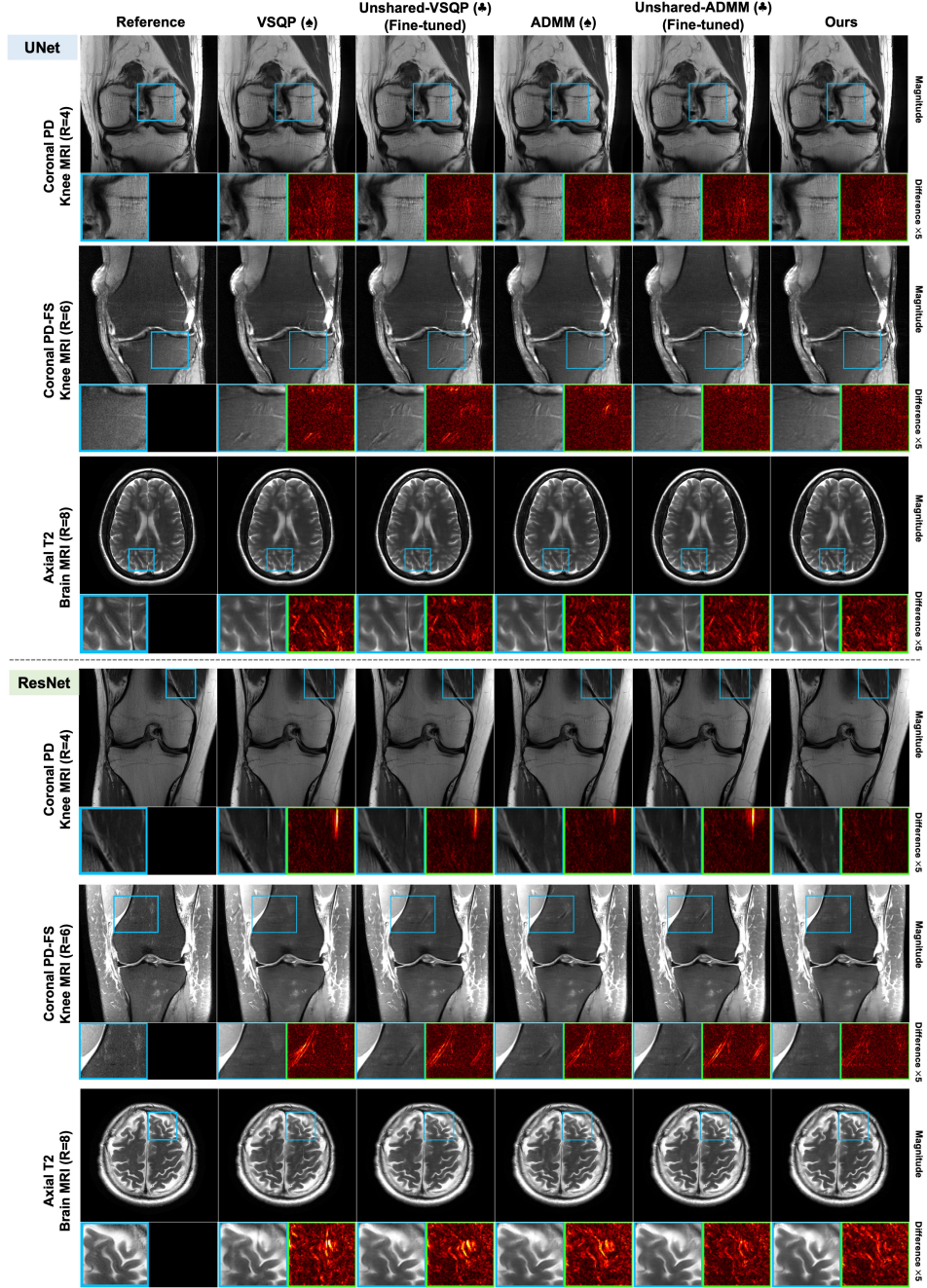


Figure 10: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Qualitative comparisons of each unrolled network with $T=15$ unrolls for **U-Net** and **ResNet** proximal operators. In each proximal operator, **Top**: Results for $R = 4$ using PD data. **Middle**: Results for $R = 6$ using PD-FS data. **Bottom**: Results for $R = 8$ using Axial T2-W data. The proposed methods effectively reduce artifacts and sharpen images, whereas the baseline methods fail to achieve this, even with 15 unrolls.

Additional qualitative results for $R=4$

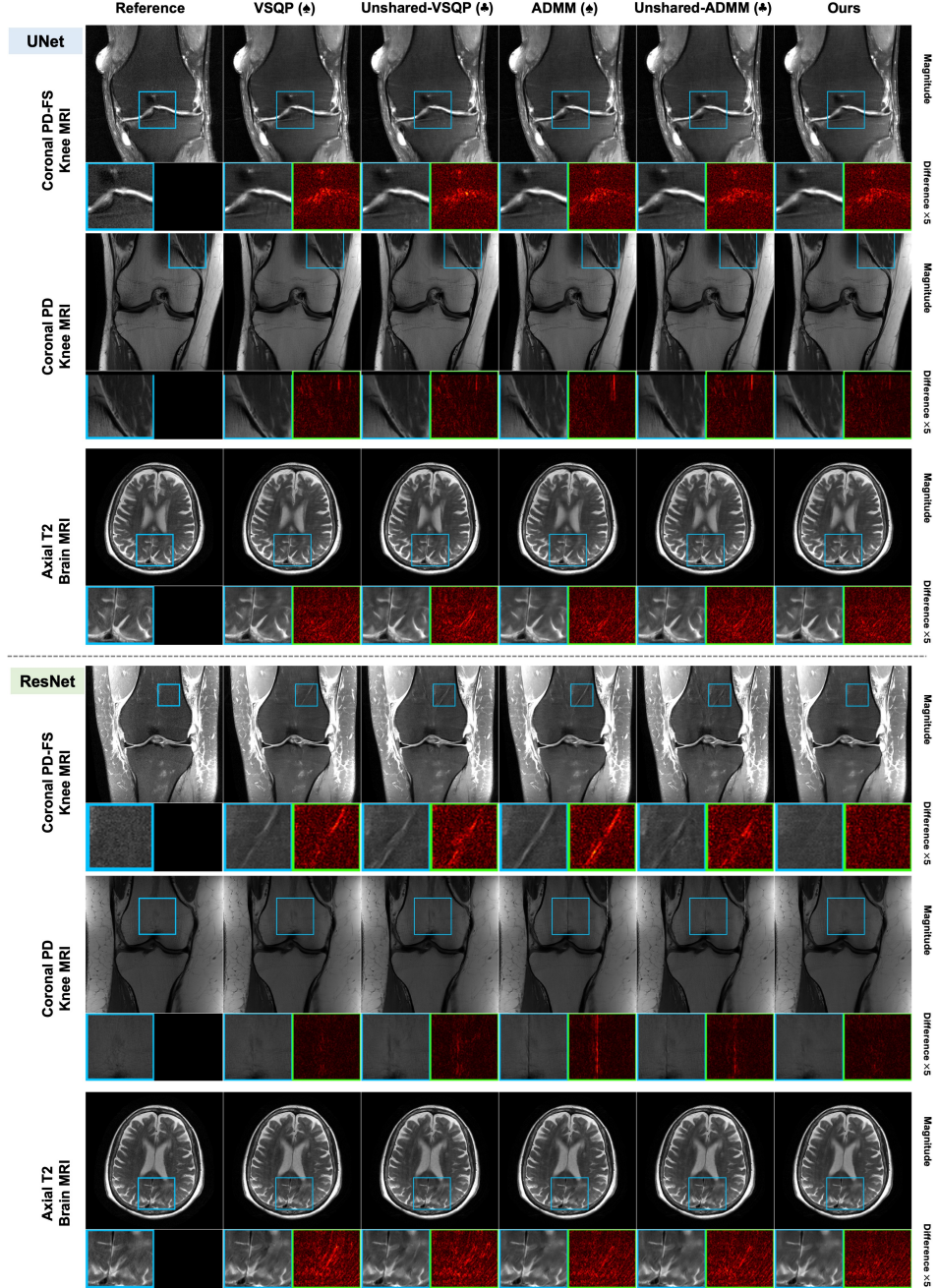


Figure 11: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Qualitative comparisons for $R = 4$ across datasets for each proximal operator ($T = 10$ unrolls). Our proposed method consistently demonstrates superior performance by reducing artifacts.

Additional qualitative results for $R=6$

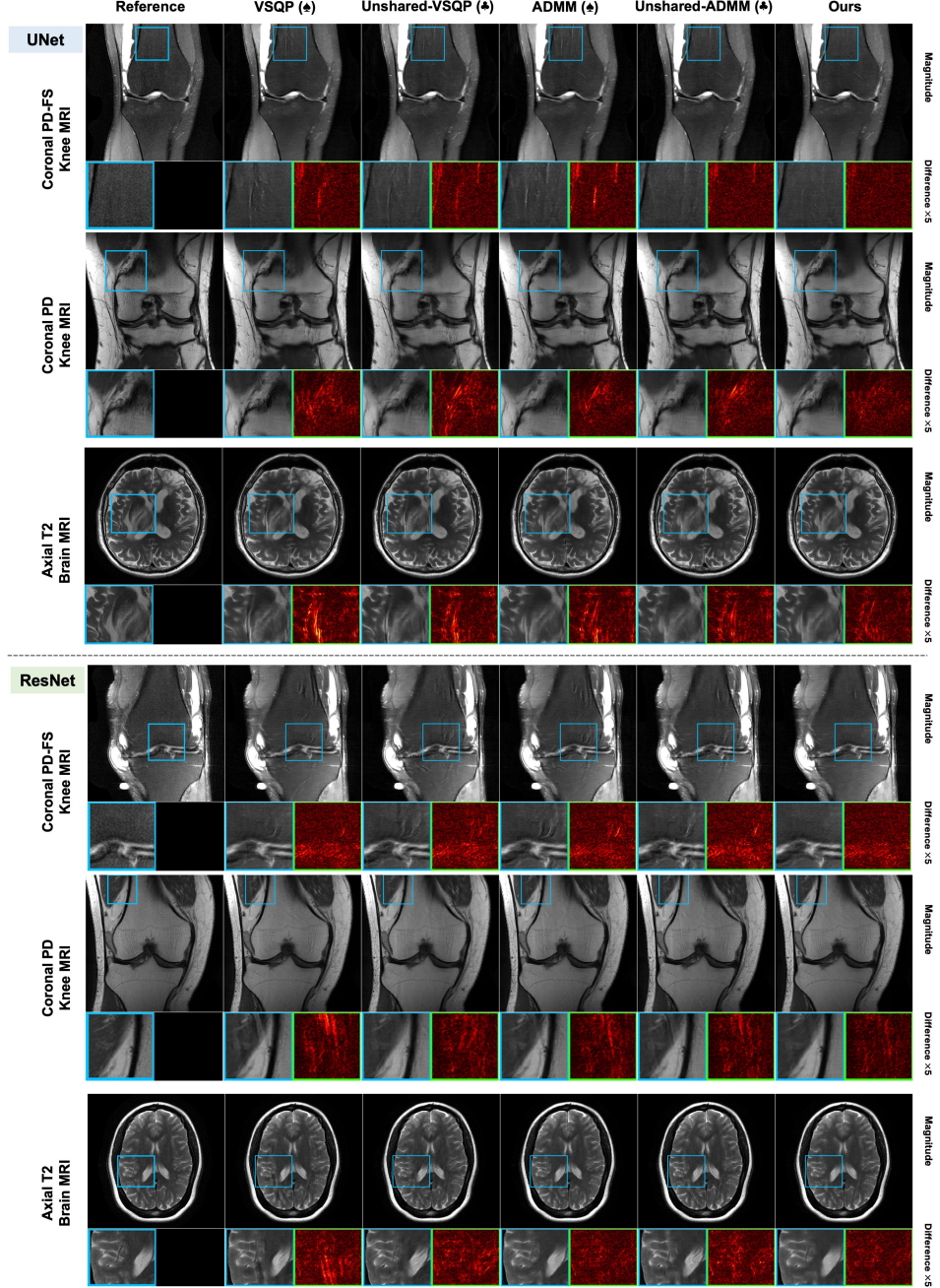


Figure 12: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Qualitative comparisons for $R = 6$ across datasets for each proximal operator ($T = 10$ unrolls). Our proposed method consistently demonstrates superior performance by reducing artifacts. Furthermore, it enhances image sharpness, as shown in the results for the axial T2 data with both U-Net and ResNet proximal operators.

Additional qualitative results for $R=8$

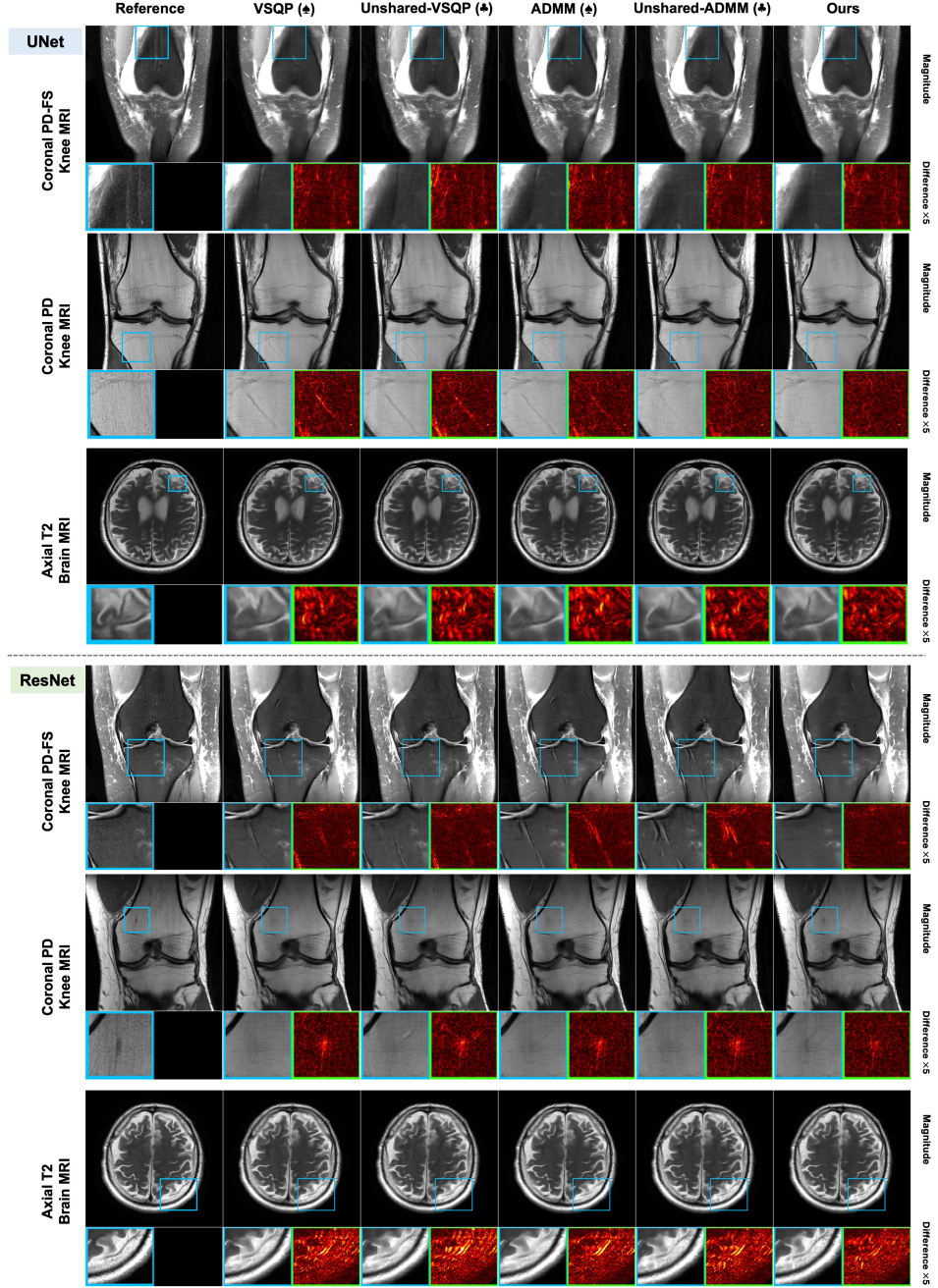


Figure 13: ♠: Shared $\mathcal{R}(\cdot)$ weights, ♣: Unshared $\mathcal{R}(\cdot)$ weights. Qualitative comparisons for $R = 8$ across datasets for each proximal operator ($T = 10$ unrolls). Similar to $R=4$ and $R=6$, $R=8$ also demonstrates artifact reduction and image sharpening. Through Fig. 11 to Fig. 13, our proposed method shows superior performance across all configurations.