

810 A THEORETICAL ANALYSIS OF SHARP

811 A.1 PROOF OF THEOREM 1

812 **Theorem.** Assume that the prior density $p_{\mathbf{x}}$ is non-degenerate over \mathbb{R}^n and let \mathbf{R}^* be the MMSE
813 restoration operator (4) corresponding to the restoration problems (3). Then, we have that

$$814 \nabla h(\mathbf{x}) = \frac{\tau}{\sigma^2} \left(\mathbb{E}_{\mathbf{s} \sim G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}), \mathbf{H} \sim p_{\mathbf{H}}} [\mathbf{H}^\top \mathbf{H}(\mathbf{x} - \mathbf{R}^*(\mathbf{s}, \mathbf{H}))] \right),$$

815 where h is the ShaRP regularizer in (6).

816 *Proof.* The ShaRP regularizer $h(\mathbf{x})$ is defined as

$$817 \begin{aligned} 818 h(\mathbf{x}) &= \tau \mathbb{E}_{\mathbf{s} \sim G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}), \mathbf{H} \sim p_{\mathbf{H}}} [-\log p(\mathbf{s}|\mathbf{H})] \\ 819 &= -\tau \int p(\mathbf{H}) \left[\int G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}) \log p(\mathbf{s}|\mathbf{H}) \, d\mathbf{s} \right] d\mathbf{H}, \end{aligned} \quad (10)$$

820 where G_σ is the Gaussian probability density with variance σ^2 and $p(\mathbf{s}|\mathbf{H})$ is the likelihood function
821 for the degraded observation given the operator \mathbf{H} . The expectation over $p(\mathbf{H})$ accounts for the
822 randomness of the restoration operator \mathbf{H} .

823 We start by relating the MMSE restoration operator to the score of the degraded observation

$$824 \nabla p(\mathbf{s}|\mathbf{H}) = \frac{1}{\sigma^2} \int (\mathbf{H}\mathbf{x} - \mathbf{s}) G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) \, d\mathbf{x},$$

825 where $p_{\mathbf{x}}$ is the prior. By using the definition of the MMSE estimator, we obtain the relationship

$$826 \nabla \log p(\mathbf{s}|\mathbf{H}) = \frac{1}{\sigma^2} (\mathbf{H}\mathbf{R}^*(\mathbf{s}, \mathbf{H}) - \mathbf{s}). \quad (11)$$

827 Consider the function inside the parenthesis in the expression for the ShaRP regularizer (10)

$$828 \rho(\mathbf{z}) := (G_\sigma * \log p_{\mathbf{s}|\mathbf{H}})(\mathbf{z}) = \int G_\sigma(\mathbf{z} - \mathbf{s}) \log p(\mathbf{s}|\mathbf{H}) \, d\mathbf{s},$$

829 where \mathbf{z} has the same dimensions as \mathbf{s} and $*$ denotes convolution. The gradient of ρ is given by

$$830 \begin{aligned} 831 \nabla \rho(\mathbf{z}) &= (\nabla G_\sigma * \log p_{\mathbf{s}|\mathbf{H}})(\mathbf{z}) = (G_\sigma * \nabla \log p_{\mathbf{s}|\mathbf{H}})(\mathbf{z}) \\ 832 &= \frac{1}{\sigma^2} \int G_\sigma(\mathbf{z} - \mathbf{s}) [\mathbf{H}\mathbf{R}^*(\mathbf{s}, \mathbf{H}) - \mathbf{s}] \, d\mathbf{s} \\ 833 &= \frac{1}{\sigma^2} \left(\mathbf{H} \int \mathbf{R}^*(\mathbf{s}, \mathbf{H}) G_\sigma(\mathbf{z} - \mathbf{s}) \, d\mathbf{s} - \mathbf{z} \right) \end{aligned}$$

834 where we used (11). By using $\mathbf{z} = \mathbf{H}\mathbf{x}$, we write the gradient with respect to \mathbf{x}

$$835 \nabla_{\mathbf{x}} \rho(\mathbf{H}\mathbf{x}) = \frac{1}{\sigma^2} \mathbf{H}^\top \mathbf{H} \left(\int \mathbf{R}^*(\mathbf{s}, \mathbf{H}) G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}) \, d\mathbf{s} - \mathbf{x} \right)$$

836 By using this expression in (10), we obtain the desired result

$$837 \begin{aligned} 838 \nabla h(\mathbf{x}) &= -\frac{\tau}{\sigma^2} \left[\int p(\mathbf{H}) \int G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}) (\mathbf{H}^\top \mathbf{H}(\mathbf{R}^*(\mathbf{s}, \mathbf{H}) - \mathbf{x})) \, d\mathbf{s} \, d\mathbf{H} \right] \\ 839 &= \frac{\tau}{\sigma^2} \mathbb{E}_{\mathbf{s} \sim G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}), \mathbf{H} \sim p_{\mathbf{H}}} [\mathbf{H}^\top \mathbf{H}(\mathbf{x} - \mathbf{R}^*(\mathbf{s}, \mathbf{H}))]. \end{aligned}$$

840 □

841 A.2 PROOF OF THEOREM 2

842 **Theorem.** Run ShaRP for $t \geq 1$ iterations using the step-size $0 < \gamma \leq 1/L$ under Assumptions 1-3.
843 Then, the sequence \mathbf{x}^k generated by ShaRP satisfies

$$844 \mathbb{E} \left[\frac{1}{t} \sum_{k=1}^t \|\nabla f(\mathbf{x}^{k-1})\|_2^2 \right] \leq \frac{2}{t} (f(\mathbf{x}^0) - f^*) + \gamma L \nu^2 + \varepsilon^2.$$

864 *Proof.* First note that from the definition of the bias in eq. (9), we have that

$$865 \mathbb{E} \left[\widehat{\nabla} f(\mathbf{x}^{k-1}) \mid \mathbf{x}^{k-1} \right] = \nabla f(\mathbf{x}^{k-1}) + \mathbf{b}(\mathbf{x}^{k-1}), \quad (12)$$

866 where the expectation is with respect to $\mathbf{s} \sim G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}^{k-1})$ and $\mathbf{H} \sim p_{\mathbf{H}}$. In order to simplify the
869 notation, we will drop these subscripts from the expectations in the analysis below.

870 Consider the iteration $k \geq 1$ of ShaRP with inexact MMSE operator

$$871 f(\mathbf{x}^k) \leq f(\mathbf{x}^{k-1}) + \nabla f(\mathbf{x}^{k-1})^\top (\mathbf{x}^k - \mathbf{x}^{k-1}) + \frac{L}{2} \|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2^2 \\ 872 = f(\mathbf{x}^{k-1}) - \gamma \nabla f(\mathbf{x}^{k-1})^\top \widehat{\nabla} f(\mathbf{x}^{k-1}) + \frac{L\gamma^2}{2} \|\widehat{\nabla} f(\mathbf{x}^{k-1})\|_2^2,$$

876 where in the first line we used the Lipschitz continuity of ∇f . By taking the expectation with respect
877 to $\mathbf{s} \sim G_\sigma(\mathbf{s} - \mathbf{H}\mathbf{x}^{k-1})$ and $\mathbf{H} \sim p_{\mathbf{H}}$ on both sides of this expression, we get

$$879 \mathbb{E}[f(\mathbf{x}^k) \mid \mathbf{x}^{k-1}] \leq f(\mathbf{x}^{k-1}) - \gamma \nabla f(\mathbf{x}^{k-1})^\top (\nabla f(\mathbf{x}^{k-1}) + \mathbf{b}(\mathbf{x}^{k-1})) + \frac{L\gamma^2}{2} \mathbb{E} \left[\|\widehat{\nabla} f(\mathbf{x}^{k-1})\|_2^2 \mid \mathbf{x}^{k-1} \right] \\ 880 \leq f(\mathbf{x}^{k-1}) - \frac{\gamma}{2} \|\nabla f(\mathbf{x}^{k-1})\|_2^2 + \frac{\gamma}{2} \|\mathbf{b}(\mathbf{x}^{k-1})\|_2^2 \\ 881 + \frac{L\gamma^2}{2} \left(\mathbb{E} \left[\|\widehat{\nabla} f(\mathbf{x}^{k-1})\|_2^2 \mid \mathbf{x}^{k-1} \right] - \left(\mathbb{E}[\widehat{\nabla} f(\mathbf{x}^{k-1}) \mid \mathbf{x}^{k-1}] \right)^2 \right) \\ 882 \leq f(\mathbf{x}^{k-1}) - \frac{\gamma}{2} \|\nabla f(\mathbf{x}^{k-1})\|_2^2 + \frac{\gamma\varepsilon^2}{2} + \frac{L\gamma^2\nu^2}{2}.$$

888 In the second row, we completed the square, applied eq. (12), and used the assumption that $\gamma \leq 1/L$.
889 In the third row, we used the variance and bias bounds in Assumptions 2 and 3. By rearranging the
890 expression, we get the following bound

$$891 \|\nabla f(\mathbf{x}^{k-1})\|_2^2 \leq \frac{2}{\gamma} (f(\mathbf{x}^{k-1}) - \mathbb{E}[f(\mathbf{x}^k) \mid \mathbf{x}^{k-1}]) + L\gamma\nu^2 + \varepsilon^2$$

894 By taking the total expectation, averaging over t iterations, and using the lower bound f^* , we get
895 the desired result

$$896 \mathbb{E} \left[\frac{1}{t} \sum_{k=1}^t \|\nabla f(\mathbf{x}^{k-1})\|_2^2 \right] \leq \frac{2}{\gamma t} (f(\mathbf{x}^0) - f^*) + L\gamma\nu^2 + \varepsilon^2.$$

900 □

901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

B EXPERIMENT DETAILS

B.1 IMPLEMENTATION DETAILS OF CS-MRI TASKS

Subsampling pattern for CS-MRI. In this paper, we explored two types of subsampling patterns for MRI reconstruction tasks. All undersampling masks were generated by first including a set number of *auto-calibration signal* (ACS) lines, ensuring a fully-sampled central k-space region.

Figure 5 illustrates the k-space trajectories for both random and uniform (equidistant) subsampling at acceleration factors of 4, 6, and 8. Notably, different patterns were used for training and testing. During training, our restoration prior was only trained on a uniform mask with a subsampling rate of 6. However, for inference, we employed both uniform and random masks at subsampling rates of 4 and 6, creating a mismatch between the pre-trained restoration prior and the test configurations.

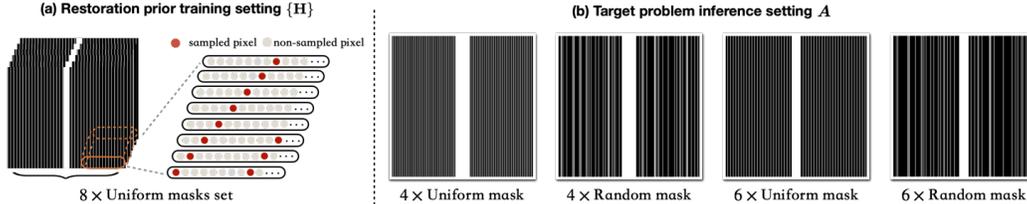


Figure 5: Illustration of the undersampling masks used for CS-MRI in this work. (a) The eight different $8 \times$ uniform masks used for training the restoration prior. (b) The inference setting for ShaRP, demonstrating how the prior trained on the masks in (a) can be applied to solve other problems without retraining.

Algorithm 2 Supervised Training of CS-MRI Restoration Network

Require: dataset: $p(x)$, sampling operator set: $\{M_1, M_2, \dots, M_8\}$, Restoration model: $R_\theta(\cdot, \alpha)$
repeat:
 $x \sim p(x)$, $M \sim \{M_1, M_2, \dots, M_8\}$, $e \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, $\alpha \sim \mathcal{U}([0, 1])$
 $y = Mx + e$
 $\min_\theta \|R_\theta((1 - \alpha)x + \alpha M^T y; \alpha) - x\|_2^2$
until converge

B.1.1 IMPLEMENTATION OF SUPERVISED PRIOR FOR CS-MRI

Models training for supervised case. We use the same U-Net architecture as employed in the official implementation of DDS² for $R(\cdot; \alpha)$. For the supervised learning case, we select 1,000 different α values to train the model, following the α schedule outlined by I²SB (Liu et al., 2023). The model is trained with Adam optimizer with a learning rate of 5×10^{-5} . As shown in Algorithm 2, we train our supervised learning model using eight different masks for $8 \times$ uniform sampling CS-MRI reconstruction. In the pseudocode, $\{M_1, M_2, \dots, M_8\}$ represent the eight different MRI degradation operators, each defined by a unique sampling pattern, as shown in Figure 5 (a). This results in a total of 8,000 possible combinations of α values and sampling masks, effectively creating an ensemble of restoration priors during training.

Inference with a Subset of the Ensemble (Supervised Case). During inference, to simplify computation and focus on the most effective priors, we use only a subset of the supervised trained ensemble. Specifically, we fix the α value to a particular choice (e.g., $\alpha = 0.5$) and use the 8 different sampling masks $\{M_1, M_2, \dots, M_8\}$, resulting in 8 restoration priors.

Step size and regularization parameter. To ensure fairness, for each problem setting, each method—both proposed and baseline—is fine-tuned for optimal PSNR using 10 slices from a validation set separate from the test set. The same step size γ and regularization parameter τ are then applied consistently across the entire test set.

²<https://github.com/HJ-harry/DDS>

Baseline details. We compare ShaRP with several variants of denoiser- and diffusion model-based methods. For denoiser-based approaches, we include PnP-FISTA (Kamilov et al., 2023), PnP-ADMM (Chan et al., 2017). PnP-FISTA and PnP-ADMM correspond to the FISTA and ADMM variants of PnP, both utilizing AWGN denoisers built on DRUNet (Zhang et al., 2022). For diffusion model-based methods, we compare with DPS (Chung et al., 2023) and DDS (Chung et al., 2024), which use pre-trained diffusion models as priors and apply different posterior sampling strategies to address general inverse problems. We use the same pre-trained diffusion model configuration as outlined in the DDS paper. For all baseline methods, we fine-tuned their parameters to maximize the PSNR value. Notably, both the DRUNet denoiser and the diffusion model were trained using the same dataset employed for training our restoration prior. For a fair comparison, the diffusion model pre-trained for DDS and DPS use the same network architecture as our restoration network. All models are trained from scratch on the fastMRI training set, following the architecture settings provided in DDS³. We also compared with method that also use the deep restoration prior to solve general inverse problem: DRP (Hu et al., 2024c). For DRP, we utilize the same pre-trained restoration network as in ShaRP. However, instead of employing a set of degradation priors, DRP uses a single fixed prior. For a fair comparison, we selected the optimal fixed prior—defined by a fixed α and subsampling mask—based on PSNR performance on the validation set, and applied it accordingly.

B.1.2 IMPLEMENTATION OF SELF-SUPERVISED PRIOR FOR CS-MRI

Algorithm 3 Self-Supervised Training of CS-MRI Restoration Network

Require: dataset: $p(\mathbf{y}_i, \mathbf{M}_i, \mathbf{y}_j, \mathbf{M}_j)$, Restoration model: $R_\theta(\cdot)$
repeat:
 $\mathbf{y}_i, \mathbf{M}_i, \mathbf{y}_j, \mathbf{M}_j \sim p(\mathbf{y}_i, \mathbf{M}_i, \mathbf{y}_j, \mathbf{M}_j), \mathbf{e} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
 $\min_{\theta} \|\mathbf{M}_j R_\theta(\mathbf{M}_i^\top (\mathbf{y}_i + \mathbf{e})) - \mathbf{y}_j\|_{\mathbf{W}}^2$
until converge

Models training for (Self-Supervised Case). For self-supervised training, the ground truth reference x is not available as a label. Instead, as shown in Algorithm 3, we work with pairs of subsampled measurements, y_i and y_j , along with their corresponding sampling operators, \mathbf{M}_i and \mathbf{M}_j . These paired measurements exhibit significant overlap within the shared *auto-calibration signal* (ACS) region, which increases the weighting of these overlapping k-space regions. Following the approach proposed by SSDEQ (Gan et al., 2023b), we introduce a diagonal weighting matrix \mathbf{W} to account for the oversampled regions in the loss function. By incorporating this weighted loss, we are able to train our MMSE restoration operator using incomplete measurements alone. Furthermore, unlike the supervised case where we use the combination of α values to form an ensemble, in the self-supervised setting, we construct the ensemble using only eight different sampling masks across the entire dataset.

Inference Using All Restoration Priors (Self-Supervised Case). During inference in the self-supervised setting, we utilize all 8 restoration priors corresponding to the different sampling masks. By incorporating the entire ensemble, we fully leverage its capacity to remove the artifacts and enhance reconstruction performance.

Step size and regularization parameter. To ensure fairness, for each problem setting, each method—both proposed and baseline—is fine-tuned for optimal PSNR using 10 slices from a validation set separate from the test set. The same step size γ and regularization parameter τ are then applied consistently across the entire test set.

Baseline details. In the self-supervised setting, we compared ShaRP with two widely used traditional methods: TV (Block et al., 2007) and GRAPPA (Griswold et al., 2002), both of which address the restoration problem without requiring fully-sampled references. Additionally, we included SPICER (Hu et al., 2024d), a recent state-of-the-art self-supervised deep unrolling method designed for MRI reconstruction using only pairs of undersampled measurements. To ensure consistency, we trained the SPICER model on the same amount of paired data used for training our restoration prior in the $8 \times$ uniform CS-MRI setting and applied it to other CS-MRI configurations.

³<https://github.com/HJ-harry/DDS>

B.2 IMPLEMENTATION DETAILS OF SISR TASKS

Algorithm 4 Gaussian Deblurring Restoration network training**Require:** dataset: $p(\mathbf{x}, \mathbf{y})$, Gaussian blur operator: \mathbf{K} , $R_\theta(\cdot, \alpha)$ **repeat:**

$$\mathbf{x} \sim p(\mathbf{x}), \mathbf{e} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \alpha \sim \mathcal{U}([0, 1])$$

$$\min_{\theta} \|R_\theta((1 - \alpha)\mathbf{x} + \alpha\mathbf{K}\mathbf{x}; \alpha) - \mathbf{x}\|_2^2$$

until converge

Restoration Model training. We use the same U-Net architecture as the Gaussian deblurring model provided by I²SB⁴. Utilizing the pre-trained checkpoints from their repository, we fine-tune our model accordingly. Specifically, we align with their codebase and configure the model type to OT-ODE to satisfy our MMSE restoration operator assumption.

To create an ensemble of restoration priors, we consider a family of degradation operators that are convex combinations of the identity mapping \mathbf{I} and the Gaussian blur operator \mathbf{K} . The blurring operator \mathbf{K} corresponds to convolution with a Gaussian blur kernel of size 31×31 and standard deviation 3. Specifically, we define the degradation operator as $\mathbf{H}_\alpha = (1 - \alpha)\mathbf{I} + \alpha\mathbf{K}$, where $\alpha \in [0, 1]$ controls the degradation level. By varying α , we generate multiple degradation operators, allowing us to train the restoration network R to handle all these operators, expressed as $R(\mathbf{s}, \mathbf{H}_\alpha) = \mathbb{E}[\mathbf{x}|\mathbf{s}, \mathbf{H}_\alpha]$, where \mathbf{s} is the degraded image and \mathbf{x} is the original image.

We select 1,000 different α values from the interval $[0, 1]$, following the α schedule outlined by I²SB (Liu et al., 2023). This results in 1,000 different degradation operators \mathbf{H}_α , effectively creating an ensemble of restoration priors during training. The model is trained using the Adam optimizer with a learning rate of 5×10^{-5} .

Inference with a Subset of the Ensemble. During inference, to simplify computation and focus on the most effective priors, we use only a subset of the supervised trained ensemble. Specifically, we select 6 α values, resulting in 6 restoration priors.

Step size and regularization parameter. To ensure fairness, for each problem setting, each method—both proposed and baseline—is fine-tuned for optimal PSNR using 5 images from a validation set separate from the test set. The same step size γ and regularization parameter τ are then applied consistently across the entire test set.

Baseline details. We compare ShaRP against several denoiser- and diffusion model-based methods. For denoiser-based approaches, we evaluate DPIR (Zhang et al., 2022), which relies on half-quadratic splitting (HQS) iterations with DRUNet denoisers. For diffusion model-based methods, we compare with DPS (Chung et al., 2023), DDNM (Wang et al., 2023), and DiffPIR (Zhu et al., 2023). These methods all use the same pre-trained diffusion models as priors, but each employs a distinct posterior sampling strategy to solve general inverse problems. We specifically use the pre-trained diffusion model from DiffPIR. We also compared with method that also use the deep restoration prior to solve general inverse problem: DRP (Hu et al., 2024c). For DRP, we utilize the same pre-trained deblurring network as in ShaRP. However, instead of employing a set of degradation priors, DRP uses a single fixed prior. For a fair comparison, we selected the optimal fixed prior—defined by a fixed α based on PSNR performance on the validation set, and applied it accordingly. For all baselines, we fine-tuned their parameters to maximize PSNR performance. Notably, the diffusion model backbone for all diffusion-based baselines was trained on the same dataset used to train our restoration prior.

⁴<https://github.com/NVlabs/I2SB>

C ADDITIONAL RESULTS FOR CS-MRI

C.1 PERFORMANCE OF SHaRP FOR RANDOM SUBSAMPLING SETTING

Due to space constraints, we present only the quantitative performance for the uniform subsampling setting in the main paper. In this section, we further evaluate ShaRP’s performance on random subsampling setting, with two sub-sampling rates ($4\times$ and $6\times$), and three noise levels ($\sigma = 0.005$, 0.01 , and 0.015).

Table 4 provides a quantitative comparison of reconstruction performance across different acceleration factors and noise levels using a uniform sub-sampling mask. In all configurations, ShaRP consistently outperforms the baseline methods. The use of a set of restoration operators clearly enhances ShaRP’s performance, highlighting the effectiveness of employing multiple operators to maximize the regularization information provided by the restoration model. Figure 6 presents visual reconstructions for two test scenarios, where ShaRP accurately recovers fine brain details, particularly in the zoomed-in regions, while baseline methods tend to oversmooth or introduce artifacts. These results highlight ShaRP’s superior ability to manage structured artifacts and preserve fine details, outperforming both denoiser-based and diffusion model-based methods.

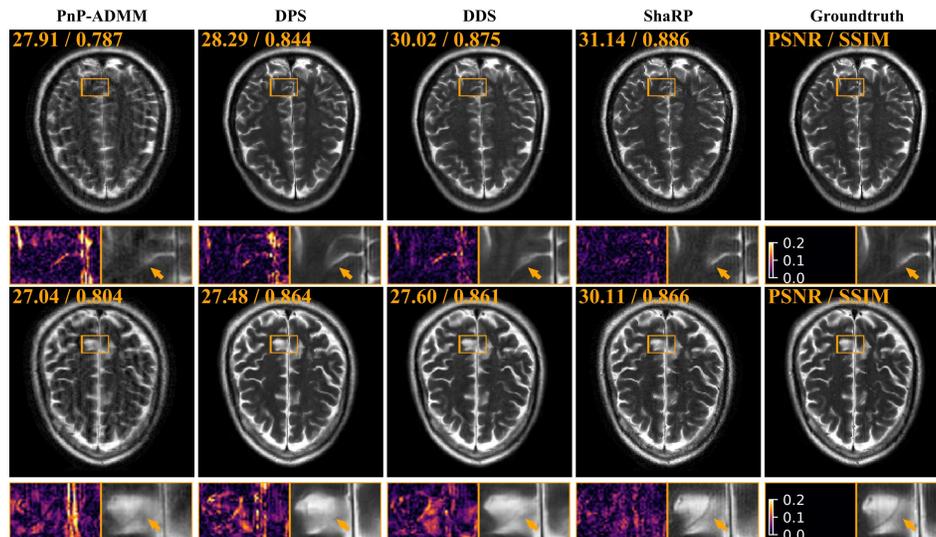


Figure 6: Visual comparison of ShaRP with baseline methods on CS-MRI for $6\times$ random sampling mask with noise $\sigma = 0.015$. PSNR and SSIM values are in the top-left corner of each image. Error maps and zoomed-in areas highlight differences. Notably, ShaRP with stochastic priors outperforms state-of-the-art methods using denoiser and diffusion model priors.

1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187

Noise level	4× Random						6× Random					
	$\sigma = 0.005$		$\sigma = 0.010$		$\sigma = 0.015$		$\sigma = 0.005$		$\sigma = 0.010$		$\sigma = 0.015$	
Metrics	PSNR	SSIM										
Zero-filled	25.83	0.815	25.81	0.812	25.76	0.807	22.68	0.724	22.67	0.722	22.67	0.719
TV	28.14	0.866	28.06	0.863	27.96	0.859	24.55	0.782	24.33	0.750	24.28	0.736
PnP-FISTA	29.31	0.863	28.40	0.817	27.49	0.799	26.01	0.797	25.63	0.756	24.94	0.717
PnP-ADMM	28.83	0.842	28.39	0.816	27.70	0.786	25.59	0.776	25.19	0.740	24.93	0.728
DRP	29.97	0.880	29.37	0.839	28.31	0.794	26.98	0.866	26.78	0.853	26.49	0.821
DPS	31.72	0.874	30.45	0.857	29.50	0.843	30.32	0.856	29.36	0.824	27.99	0.810
DDS	<u>32.41</u>	<u>0.910</u>	<u>32.37</u>	<u>0.906</u>	<u>32.25</u>	<u>0.901</u>	<u>30.59</u>	<u>0.876</u>	<u>30.35</u>	<u>0.874</u>	<u>30.31</u>	<u>0.879</u>
ShaRP	34.66	0.949	33.57	0.920	33.18	0.931	31.53	0.924	31.46	0.918	31.45	0.914

Table 4: Quantitative comparison of ShaRP with several baselines for CS-MRI using random masks at undersampling rates of 4 and 6 on fastMRI dataset. The **best** and second best results are highlighted. Notably, ShaRP outperforms SOTA methods based on denoisers and diffusion models.

C.2 PERFORMANCE OF ADDITIONAL BASELINE METHODS ON MATCHED AND MISMATCHED SETTINGS

In this section, we highlight an important observation: pre-trained restoration networks typically exhibit poor generalization to mismatched settings. We chose two commonly used methods (SwinIR (Liang et al., 2021) and E2E-VarNet (Sriram et al., 2020)) for the specific setting of CS-MRI. We trained them on the same $8\times$ uniform subsampling setting as our restoration prior and directly applied them to solve both matched and mismatched problems, as ShaRP did. As shown in the Table 5, the baseline method’s performance drops significantly under mismatched conditions, whereas ShaRP maintains stable performance and convergence guarantees. This demonstrates ShaRP’s ability to adapt pre-trained restoration models as priors and use it to solve problems under mismatched settings. As shown in the Figure 7, due to the mismatched settings, the two baseline methods suffer from over-smoothing, lack important details, and exhibit artifacts, whereas ShaRP still provides high-quality reconstruction performance. This indicates that ShaRP can balance data fidelity and the artifact removal capabilities of the prior model, leading to an artifact-free reconstruction that preserves important details.

Settings	4× Uniform		4× Random		6× Uniform		6× Random		8× Uniform	
Metrics	PSNR	SSIM								
SwinIR	24.78	0.849	25.09	0.841	29.55	0.907	27.98	0.819	29.37	0.898
E2E-VarNet	<u>35.40</u>	<u>0.957</u>	33.48	0.945	<u>32.79</u>	<u>0.936</u>	<u>31.02</u>	<u>0.913</u>	32.59	0.919
ShaRP	37.59	0.963	34.66	0.949	33.42	0.940	31.53	0.924	<u>32.37</u>	<u>0.907</u>

Table 5: Quantitative comparison of ShaRP with task-specific baselines trained on the $8\times$ uniform mask. Baselines perform well in matched settings (highlighted in the table) but show a significant drop under mismatched conditions. In contrast, ShaRP remains robust, handling both matched and mismatched scenarios effectively.

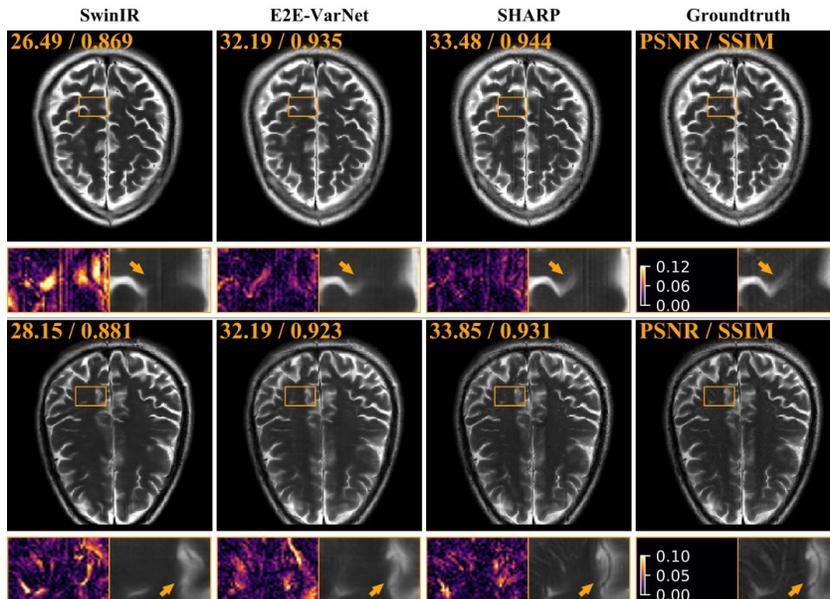


Figure 7: Visual comparison of ShaRP with task-specific baseline methods on CS-MRI for $6\times$ random sampling mask with noise $\sigma = 0.015$. PSNR and SSIM values are in the top-left corner of each image. Error maps and zoomed-in areas highlight differences. Notably, ShaRP with stochastic priors outperforms state-of-the-art methods using denoiser and diffusion model priors.

D ADDITIONAL VISUAL RESULTS FOR SISR

In this section, we present additional visual results to numerical comparisons for the SISR task.

D.1 ADDITIONAL VISUAL RESULTS AGAINST BASELINES

As illustrated in Figure 8 and Figure 9, ShaRP outperforms all baseline approaches under both blur kernel settings, achieving higher PSNR and SSIM values. Moreover, we maintain superior data consistency with the measurements while achieving enhanced perceptual quality. The use of an ensemble of deblurring priors enables our method to recover fine details at varying corruption levels, contributing to the improved performance.

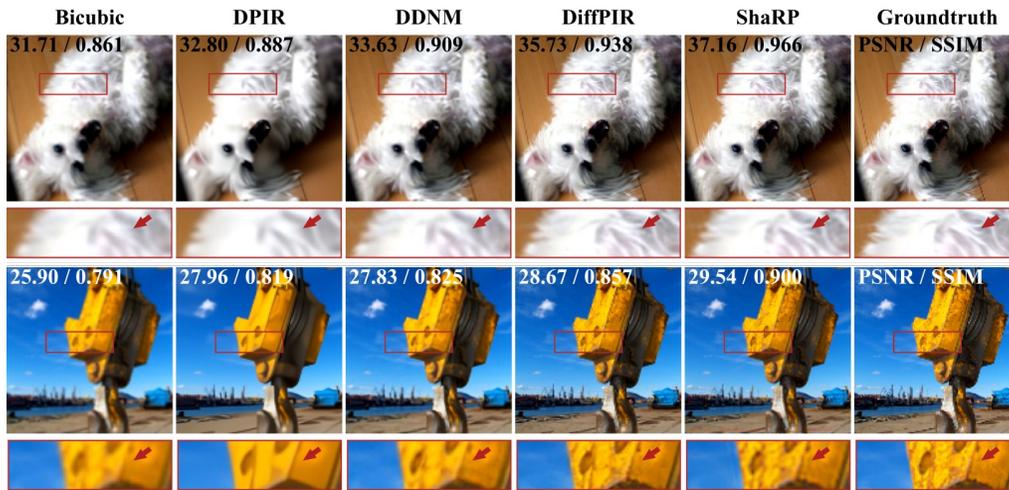


Figure 8: Visual comparison of ShaRP with several well-known methods on $2\times$ SISR with gaussian blur kernel with $\sigma = 1.5$. The quantities in the top-left corner of each image provide PSNR and SSIM values for each method. The squares at the bottom of each image visualize the zoomed area in the image.

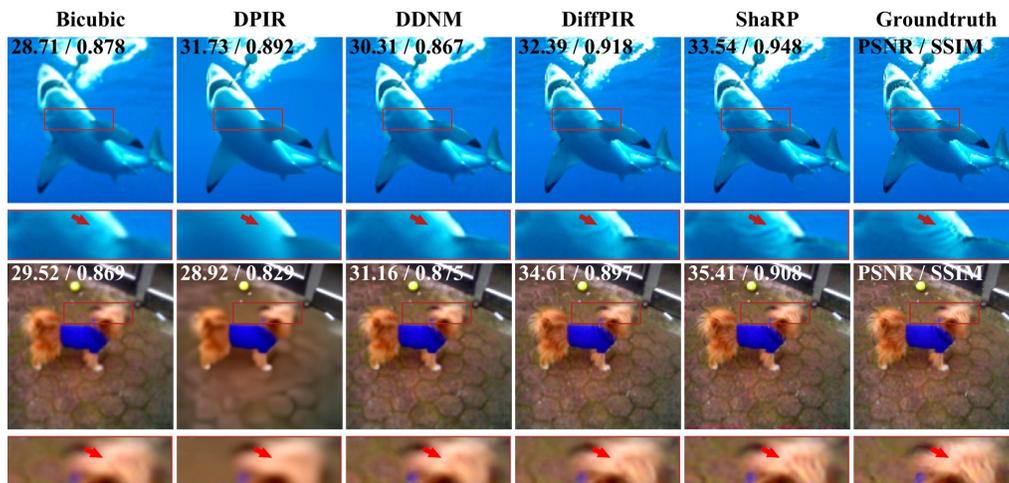


Figure 9: Visual comparison of ShaRP with several well-known methods on $2\times$ SISR with gaussian blur kernel with $\sigma = 1.5$. The quantities in the top-left corner of each image provide PSNR and SSIM values for each method. The squares at the bottom of each image visualize the zoomed area in the image.

1296 D.2 ADDITIONAL VISUAL RESULTS AGAINST DRP
 1297

1298 To further emphasize the necessity and advantages of using an ensemble of deblurring priors, as
 1299 opposed to a fixed prior like in DRP (Hu et al., 2024c), we provide additional visual comparison
 1300 results. As shown in Figure 10, ShaRP consistently recovers finer details, resulting in improved
 1301 PSNR and SSIM scores, along with enhanced perceptual performance.
 1302



1330 Figure 10: Visual comparison of ShaRP with DRP on $2\times$ SISR with gaussian blur kernel with
 1331 $\sigma = 1.5$. The quantities in the bottom-left corner of each image provide PSNR and SSIM values for
 1332 each method. The squares at the bottom of each image visualize the zoomed area in the image.
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349