S^3-PnP: Plug-and-Play Denoising for Scientific Imaging with Self-Supervised Score Priors

Anonymous authors
Paper under double-blind review

Abstract

Modern image restoration methods increasingly rely on large-scale supervised training, yet clean targets are often prohibitively difficult to obtain in scientific imaging modalities such as hyperspectral imaging (HSI) and magnetic resonance imaging (MRI). Plug-and-play ADMM (PnP-ADMM) with pretrained diffusion priors offers a promising alternative, but the learned prior distribution can mismatch both the target scientific image distribution and the distribution of ADMM denoiser inputs induced by dual feedback. To mitigate these mismatches, we propose S^3-PnP, a self-supervised plug-and-play restoration framework that learns a score prior directly from observation-derived ADMM intermediate states, without clean training targets or externally pretrained weights. Specifically, the learned score corrects the input to the analytical prior update, combining observation-driven corrections with analytical regularization. We further analyze the convergence and reconstruction error of the corrected ADMM algorithm assuming an ideal score and quadratic data fidelity. Experiments on reference-guided HSI restoration, single-image HSI denoising, and MRI denoising demonstrate consistent improvements over competing methods.

1 Introduction

Recovering high-quality images from noisy measurements is a fundamental challenge in scientific imaging (Zhu et al., 2018). In modalities such as hyperspectral imaging (HSI) and magnetic resonance imaging (MRI), noise degrades image quality and obscures important structures. However, clean reference images are often difficult to acquire, limiting the availability of paired noisy and clean data for supervised end-to-end restoration. The alternating direction method of multipliers (ADMM) provides a model-based framework that combines observation fidelity with analytical priors (Boyd et al., 2011). To describe the ADMM formulation, let A denote the image and $f(A)$ the data-fidelity term. Introducing an auxiliary variable V yields:

$$\min_{A,V} \; f(A) + \lambda \Psi(V) \qquad \text{subject to} \qquad A = V. \tag{1}$$

Here Ψ is an analytical regularizer and $\lambda > 0$ controls its strength. With a scaled dual variable U and penalty parameter $\rho > 0$, ADMM performs the following updates:

$$\begin{aligned} &A^{k+1} = \operatorname{prox}_{(f/\rho)}(V^k - U^k) = \arg\min_A f(A) + \frac{\rho}{2}\|A - V^k + U^k\|_2^2, \\ &C^k = A^{k+1} + U^k, \qquad V^{k+1} = \mathcal{T}_k(C^k), \qquad U^{k+1} = U^k + A^{k+1} - V^{k+1}. \end{aligned} \tag{2}$$

In these updates, C^k denotes the *ADMM denoiser input*, combining the data update with dual feedback, and $\mathcal{T}_k$ denotes the analytical prior operator. ADMM-based PnP methods replace this analytical prior operator with a denoising operator $\mathcal{C}_k$, yielding $V^{k+1} = \mathcal{C}_k(C^k)$ (Venkatakrishnan et al., 2013).

Diffusion-based PnP methods and ADMM variants often use diffusion models pretrained on natural RGB images (Zhu et al., 2023; Wu et al., 2024; Zhang et al., 2025; Shrestha & Fu, 2026). Figure 1(a) illustrates how a pretrained diffusion model supplies the prior update within ADMM. Applying such priors to scientific imaging can introduce two distinct mismatches, illustrated in Figure 1(b). First, natural RGB images differ from scientific images in their image statistics and representations, creating

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

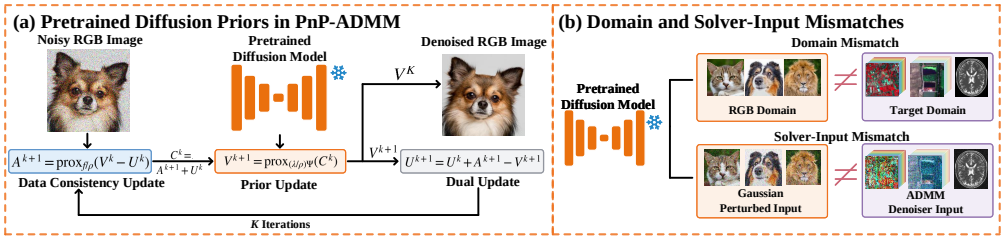

Figure 1: Two sources of mismatch when using pretrained image priors in scientific imaging: differences between the pretraining and target image domains, and differences between score-model training inputs and ADMM denoiser inputs.

a mismatch between the pretraining and target domains. Adapting these priors to modalities such as hyperspectral imaging and MRI may therefore require additional representation or model adaptation (Pang et al., 2024; Xiao et al., 2025; Gao & Zhou, 2024). Second, even within the same modality, ADMM denoiser inputs need not follow the Gaussian perturbation distributions used to train score models, because they also reflect dual feedback and solver-dependent errors. AC–DC mitigates this mismatch by adapting solver inputs to a pretrained score model through noise injection and conditional Langevin correction (Shrestha & Fu, 2026). This motivates a complementary direction: learning score corrections directly from observation-derived ADMM states, so that their statistics inform the prior update without requiring an externally pretrained model.

To address these mismatches in scientific image restoration, we introduce S^3-PnP for settings without domain-matched pretrained priors. The framework learns a self-supervised score approximation at test time from ADMM denoiser inputs generated by the available observations. The learned score corrects the input to the analytical prior update, combining observation-adapted state statistics with analytical regularization. Training requires neither clean targets nor externally pretrained weights. Under a matched quadratic denoising model with ideal score corrections, we establish sufficient conditions for convergence of the corrected ADMM iterations and lower expected reconstruction error relative to the matched analytical ADMM baseline. Because structured interference may enter the learned state statistics, a structure-guided learning strategy combines complementary orthogonal projections with sequential training. We evaluate S^3-PnP on reference-guided HSI restoration, single-image HSI denoising, and MRI denoising. Our contributions are:

- **Self-supervised score prior learning.** For scientific image restoration, S^3-PnP learns score corrections from ADMM states at test time to modify the analytical prior input. Training requires neither clean targets nor externally pretrained weights.
- **Convergence and reconstruction error analysis.** We establish sufficient conditions for S^3-PnP to converge and achieve lower expected reconstruction error than matched analytical ADMM, assuming an ideal score and quadratic data fidelity.
- **Score learning in complementary orthogonal subspaces.** We learn and independently control score corrections in complementary orthogonal subspaces, enabling separate treatment of corrections associated with structured interference.

2 Related Work

Plug-and-Play and Generative Priors for Image Restoration. PnP combines denoisers with observation-model updates (Venkatakrishnan et al., 2013), with subsequent analyses covering fixed-point convergence, Lipschitz conditions, and proximal denoiser constructions (Chan et al., 2017; Ryu et al., 2019; Hurault et al., 2022). For RGB restoration, DDRM uses spectral decompositions of linear observation operators (Kawar et al., 2022), while DDNM uses range–null-space decomposition to enforce data consistency (Wang et al., 2023). DPS and ΠGDM guide diffusion sampling through approximate measurement likelihoods (Chung et al., 2023; Song et al., 2023). DiffPIR combines diffusion denoising with half-quadratic splitting (Zhu et al., 2023), and PnP-DM uses split Gibbs sampling for likelihood and prior updates (Wu et al., 2024). PnP-Flow constructs time-dependent denoisers from pretrained flow-matching models (Martin et al., 2025), while CM4IR uses consistency models for few-step restoration (Garber & Tirer, 2025). ADMMDiff coordinates generative priors

and guidance within ADMM (Zhang et al., 2025). For non-RGB imaging, PLRDiff and HIR-Diff use low-dimensional spectral representations to apply pretrained three-channel diffusion priors to HSI restoration (Rui et al., 2024; Pang et al., 2024). DM-ZS learns guidance and spectral mapping from PAN and low-resolution HSI observations for pansharpening with a pretrained RGB diffusion model (Xiao et al., 2025). For MRI reconstruction, Chung & Ye (2022) combine an MRI-trained score prior with data-consistency updates, while DDS couples diffusion sampling with conjugate-gradient updates (Chung et al., 2024). MRPD transfers a natural-image latent diffusion prior to MRI with optional autoencoder adaptation (Gao & Zhou, 2024), and PnP-CM incorporates MRI-trained consistency models into PnP-ADMM (Gülle et al., 2026). Our framework combines learned score corrections with an analytical prior update within ADMM. The corrections act on the prior input, allowing learned information to complement analytical regularization.

Self-Supervised Prior Learning and Adaptation. Self-supervised learning and prior adaptation address limited clean supervision and mismatch with the target restoration problem. Shoushtari et al. (2024) analyze denoiser mismatch in PnP-ADMM and investigate adaptation using target-domain samples. AC–DC addresses the mismatch between pretrained score models and ADMM inputs shaped by dual feedback through Gaussian noise injection and conditional Langevin corrections (Shrestha & Fu, 2026). To learn without clean targets, Noise2Self constructs self-supervised losses using noise independence and masked prediction (Batson & Royer, 2019). For score estimation, DSM connects perturbation-based denoising regression to distribution scores (Vincent, 2011). Noise2Score learns the score of the noisy-image distribution and uses Tweedie's formula for restoration (Kim & Ye, 2021), while C2S introduces generalized denoising score matching on further-corrupted observations for MRI denoising (Tu et al., 2025). Learning directly from the observations to be restored provides another route to reducing external data requirements. ZS-N2N trains a lightweight network on downsampled pairs from a single noisy image (Mansour & Heckel, 2023), and DDS2M optimizes spatial and spectral networks during reverse diffusion using only the degraded HSI (Miao et al., 2023). For diffusion MRI, DDM^2 combines statistical denoising with generative diffusion in a three-stage framework (Xiang et al., 2023); Di-Fusion uses later diffusion steps and adaptive sampling in a single-stage self-supervised framework (Wu et al., 2025). We learn score corrections at test time from ADMM states generated by the available observations. These corrections modify the input to an analytical prior update, combining observation-driven learning with analytical regularization without externally pretrained weights.

3 METHOD

3.1 SELF-SUPERVISED SCORE APPROXIMATION FROM ADMM ITERATES

Analytical priors in ADMM capture structures such as low rank and sparsity, but may not fully describe the statistics of the denoiser inputs generated during optimization. These statistics can complement analytical regularization, helping suppress noise while preserving image content. Therefore, S^3-PnP learns an implicit regularization correction from ADMM denoiser inputs generated by the current observations. We estimate this correction through self-supervised score regression without externally pretrained weights and apply it before the analytical prior update.

Self-supervised score learning. Firstly, we run Eq. (2) with its analytical prior update and collect the denoiser inputs $C^k = A^{k+1} + U^k$ into a training set $\mathcal{B}$ to capture observation-dependent state statistics. Each state $C \in \mathcal{B}$ is normalized as $Z = \mathcal{S}(C) = (C - m)/s$, using fixed medians m and positive scales s estimated from $\mathcal{B}$. Then, we construct the perturbed state $Z_t = Z + \sqrt{\Delta_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is independent of Z and $\Delta_t > 0$ is the artificial perturbation variance. In order to formulate a correction direction from these state statistics, let p_t denote the density of Z_t and define the state energy $R_t(z) = -\log p_t(z)$. The score of the perturbed state distribution is its log-density gradient, $\nabla_z \log p_t(z) = -\nabla_z R_t(z)$, which provides a correction direction reflecting the reference-state statistics. Since p_t is unknown, we estimate this score through self-supervised regression using the collected states and their artificial perturbations. Specifically, we build on the regression principle of generalized denoising score matching (GDSM) (Tu et al., 2025). The following lemma establishes the relationship between its optimal restoration predictor and the state score, providing the basis for learning the required correction without clean training targets.

Lemma 1. *Under the state construction and fixed normalization above, assume $\mathbb{E}\|Z\|^2 < \infty$. For fixed t and $\kappa_t > \Delta_t > 0$, consider the GDSM interpolation loss*

$$\mathcal{L}_{\mathrm{GDSM}}(\widehat{Z}; t) = \mathbb{E}\left\| \frac{\Delta_t}{\kappa_t} \widehat{Z}(Z_t, t) + \left(1 - \frac{\Delta_t}{\kappa_t}\right) Z_t - Z \right\|^2 . \tag{3}$$

Its minimizer over square-integrable restoration predictors satisfies, almost surely,

$$\widehat{Z}^\star(Z_t, t) = Z_t + \kappa_t s_t^\star(Z_t), \tag{4}$$

where $s_t^\star = \nabla \log p_t = -\nabla R_t$. Moreover, $\mathbb{E}[Z \mid Z_t] = Z_t + \Delta_t s_t^\star(Z_t)$. If additionally $Z = Z_0 + \sigma_0 \epsilon_0$, where Z_0 denotes the underlying clean state in normalized coordinates, $\sigma_0 > 0$, $\epsilon_0 \sim \mathcal{N}(0, I)$, and $Z_0, \epsilon_0, \epsilon$ are mutually independent, choosing $\kappa_t = \sigma_0^2 + \Delta_t$ gives $\widehat{Z}^\star(Z_t, t) = \mathbb{E}[Z_0 \mid Z_t]$.

See Appendix A.1 for the proof.

To learn the state score directly, we parameterize the restoration predictor as $\widehat{Z}_\theta(Z_t, t) = Z_t + \kappa_t s_\theta(Z_t, t)$ in Eq.(4), where θ denotes the trainable parameters of the correction network. When training a shared network across perturbation levels, differences in regression-error scale can affect their relative contributions to optimization. We therefore use noise-level-dependent normalization of the scaled correction error and express the resulting objective as weighted score regression. For fixed t, before task-specific output processing, the GDSM interpolation loss satisfies

$$\mathcal{L}_{\mathrm{GDSM}}(\widehat{Z}_\theta; t) = \frac{\Delta_t^2}{w(t)} \mathcal{L}_{\mathrm{score}}(\theta; t) = \frac{\Delta_t^2}{w(t)} \mathbb{E}\Big[w(t) \left\| s_\theta(Z_t, t) - \Delta_t^{-1}(Z - Z_t) \right\|^2 \Big] . \tag{5}$$

Here $w(t) = \kappa_t^2 (N \gamma \sigma_t^2)^{-1}$, where N is the number of tensor entries per sample, $\gamma > 0$ controls the loss scale, and $\sigma_t > 1$ specifies the perturbation level. For each fixed t, the factor $\Delta_t^2 / w(t)$ is positive and independent of θ, so the two objectives share the same minimizer under this parameterization. Training uses $\mathcal{L}_{\mathrm{score}}$ averaged over the sampled perturbation levels, with $w(t)$ determining their relative emphasis.

We then use the learned score to correct the input to the analytical prior update. With $Z = \mathcal{S}(C)$, the corrected operator is

$$\mathcal{C}_k(C) = \mathcal{T}_k\big(\mathcal{S}^{-1}[Z + \beta_k \kappa_{t_k} s_\theta(Z, t_k)]\big) . \tag{6}$$

Here $\beta_k \geq 0$ controls the correction strength, κ_{t_k} is the score-scaling coefficient introduced in Lemma 1, t_k specifies the smoothing level, and $\mathcal{T}_k$ is the analytical prior update. For the ideal score, $s_{t_k}^\star = -\nabla R_{t_k}$, so the correction is a gradient step on the state energy in normalized coordinates. Setting $\beta_k = 0$ recovers the analytical ADMM solver in Eq. (2). We refer to the resulting ADMM framework with self-supervised score corrections as S^3-PnP.

Convergence and reconstruction error. Next, we establish sufficient conditions under which the corrected ADMM algorithm converges and its limiting estimate has lower expected reconstruction error than that of analytical ADMM. The analysis assumes a matched statistical model with an ideal score, fixed operators, and positive-definite quadratic fidelity.

Theorem 1. Under the clean-state and moment assumptions of Lemma 1, let $A_0 = \mathcal{S}^{-1}(Z_0)$, $Y = \mathcal{S}^{-1}(Z_t)$. Consider ADMM with fixed $\rho > 0$, $H = H^* \succ 0$, and $f(A) = \frac{1}{2}\langle A - Y, H(A - Y)\rangle$. For $\beta_k \equiv \beta \in [0, 1]$, use $V^{k+1} = \mathcal{T}(C^k + \beta d_t(C^k))$, where the ideal correction $d_t(C) = \kappa_t s \odot s_t^\star(\mathcal{S}(C))$ is L-Lipschitz. Set $r_H = \max_j |(\rho - \lambda_j(H))/(\rho + \lambda_j(H))|$, where $\lambda_j(H) > 0$ are the eigenvalues of H.

(i) **Convergence**. Let $C_\beta^k = A_\beta^{k+1} + U_\beta^k$ denote the ADMM denoiser input at iteration k with correction strength β. If $r_H(1 + 2\beta L) < 1$, the iterates converge from any finite initialization and satisfy

$$\|C_\beta^k - C_\beta^\star\| \leq q_\beta^k \|C_\beta^0 - C_\beta^\star\|, \tag{7}$$

where $C_\beta^\star$ is the unique fixed point of the denoiser-input iteration and $q_\beta = \frac{1}{2}[1 + r_H(1 + 2\beta L)] < 1$. Denote the common limit of the primal iterates A_β^k and V_β^k by $A_\beta^\star$.

(ii) **Error improvement**. Under the same stability condition, additionally assume $\|\mathcal{T}(Q) - Q\| \leq b$ for all Q, with $b \geq 0$. Let $a_t = (\mathbb{E}\|Y - \mathbb{E}[A_0 \mid Y]\|^2)^{1/2}$, with δ_β, c_β, and b_H defined in Appendix A.2. For $\beta > 0$, if $\delta_\beta < 1$ and $c_\beta a_t + b_H/(1 - \delta_\beta) < a_t - b_H$, then

$$\mathbb{E}\|A_\beta^\star - A_0\|^2 < \mathbb{E}\|A_{\mathrm{off}}^\star - A_0\|^2. \tag{8}$$

Here $A^\star_{\text{off}}$ is the converged estimate of the analytical ADMM solver in Eq. (2), recovered by setting $\beta = 0$ while keeping the data term, analytical map, and penalty unchanged.

Appendix A.2 gives the remaining constants and proof. We further examine the theoretical predictions on convergence and reconstruction error through controlled experiments in Section 4.4 and Appendix C.1. In practical imaging, structured interference may persist in the reference states and be reflected in the learned score. Therefore, we introduce structure-guided score corrections in Section 3.2.

3.2 Self-Supervised Score Learning and S^3-PnP Inference

The previous subsection described how the state score corrects the input to the analytical prior update. Here, we explain how to learn these corrections, following the two procedures in Figure 2: full-space learning and structure-guided learning. Full-space learning estimates corrections over the entire space, while structure-guided learning separately controls correction components associated with structured interference. For full-space learning, we use the lower module with $P_s = 0$, $P_g = I$, and $\beta_s = 0$. Training begins by running analytical ADMM without learned corrections to collect denoiser inputs $C^k = A^{k+1} + U^k$ into a state training set $\mathcal{B}_0$. We then use these states and their artificial perturbations to train the full-space network D_g through self-supervision, as summarized in Algorithm 1. After training, we freeze the network and add its correction to the current state before the analytical prior update during inference.

However, structured interference, such as column stripes in hyperspectral images, can remain in the ADMM states used for full-space learning and influence the learned score. To control stripe-related corrections separately, we use imaging knowledge to construct a subspace associated with column stripes and its orthogonal complement. These subspaces are represented by orthogonal projectors P_s and $P_g = I - P_s$, satisfying $P_s + P_g = I$ and $P_s P_g = 0$. The two correction components are then learned separately, with their strengths adjusted independently. For a fixed reference-state distribution, let $s^\star$ denote the ideal score-induced correction and define $s^\star_i = P_i s^\star$ for $i \in \{s, g\}$. The correction then decomposes as

$$s^\star = P_s s^\star + P_g s^\star = s^\star_s + s^\star_g. \tag{9}$$

The projector construction is detailed in Appendix B.4. For structure-guided learning, we use two lightweight conditional U-Nets, D_s and D_g, with the same architecture and independent parameters θ_s and θ_g. Following the orange path in Figure 2, we first train the structure network D_s on the state training set $\mathcal{B}_0$ generated by analytical ADMM. We then freeze this network and rerun ADMM with its correction enabled to collect a new state training set $\mathcal{B}_1$ for training the complementary network D_g. The two stages use their respective fixed normalizations and reference-state distributions. We train each network using the self-supervised objective

$$\mathcal{L}_i = \mathcal{L}_{\text{score},i} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux},i}, \qquad i \in \{s, g\}, \tag{10}$$

where the score-regression term follows Eq. (5), with task-specific output processing and auxiliary losses detailed in Appendix B. During inference, the networks, normalizations, and deployment projectors remain fixed. The active networks process the same ADMM input C^k. Let $s_i(C^k, t_k; \theta_i)$ denote the learned correction in the original coordinates after rescaling, output processing, and projection by P_i. The score-scaling factor κ_{t_k} is included in s_i. When $\beta_i = 0$, branch i is skipped and its correction is set to zero. As illustrated on the right of Figure 2, these corrections are combined before a single analytical prior update:

$$V^{k+1} = \mathcal{T}_k\big(C^k + \beta_s s_s(C^k, t_k; \theta_s) + \beta_g s_g(C^k, t_k; \theta_g)\big). \tag{11}$$

Here $\beta_s, \beta_g \geq 0$ control the correction strengths, while the data and dual updates follow Eq. (2). Setting both gains to zero recovers the matched analytical ADMM solver. Algorithms 2 and 3 summarize sequential training and unified inference, respectively.

4 Experiments

We evaluate S^3-PnP on reference-guided HSI restoration, single-image HSI denoising, and MRI denoising. The experiments include quantitative and qualitative comparisons, ablation studies, and controlled tests of the theoretical predictions.

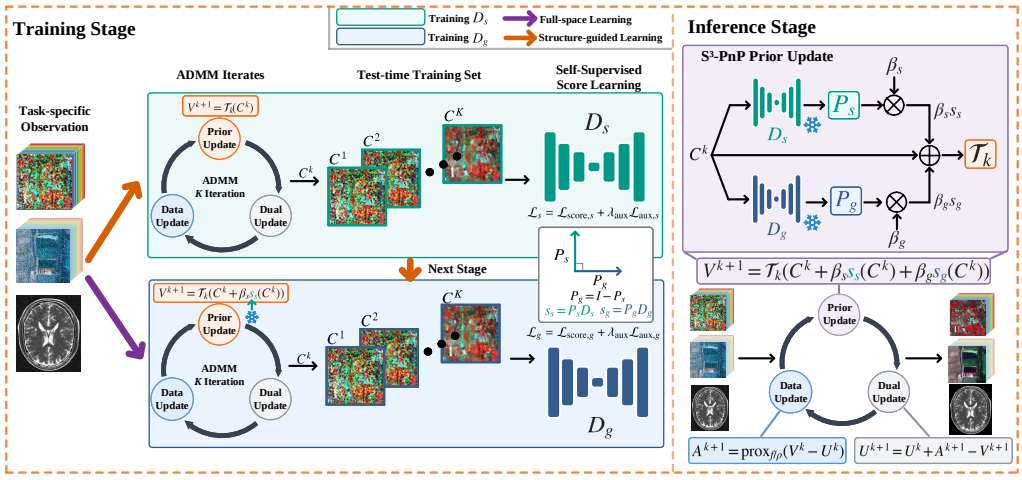

Figure 2: Training and inference of S^3-PnP. Purple arrows indicate full-space learning with $P_s = 0$, $P_g = I$, and $\beta_s = 0$. Orange arrows indicate structure-guided learning: D_s is trained and frozen, and its correction is enabled to collect a new state training set for D_g. At inference, active corrections are evaluated at the same ADMM input and combined before a single analytical prior update. Snowflakes denote frozen networks.

Algorithm 1 LEARNSCORE$(\mathcal{B}, i)$

Input: State training set $\mathcal{B}$, branch i, M

Initialize θ_i
for $n = 1, \ldots, M$ **do**
  Sample Z from $\mathcal{B}$, t, $\epsilon \sim \mathcal{N}(0, I)$
  $Z_t \leftarrow Z + \sqrt{\Delta_t}\epsilon$
  $w(t) \leftarrow \kappa_t^2 (N\gamma\sigma_t^2)^{-1}$
  $\mathcal{L}_i \leftarrow \mathcal{L}_{\text{score},i} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux},i}$
  $\theta_i \leftarrow \text{Adam}(\theta_i, \nabla_{\theta_i}\mathcal{L}_i)$
end for
return θ_i

Algorithm 2 Structure-guided Training

Input: Visible observations, P_s, $P_g = I - P_s$, $\beta_s > 0$

Run ADMM via Eq. (11), $\beta_s = \beta_g = 0$
$\mathcal{B}_0 \leftarrow \{C^k\}_{k \in \mathcal{K}}$
$\theta_s \leftarrow$ LEARNSCORE$(\mathcal{B}_0, s)$ ▷ Alg. 1
Freeze θ_s
Rerun ADMM via Eq. (11) with θ_s, $\beta_g = 0$
$\mathcal{B}_1 \leftarrow \{C^k\}_{k \in \mathcal{K}}$
$\theta_g \leftarrow$ LEARNSCORE$(\mathcal{B}_1, g)$ ▷ Alg. 1
return (θ_s, θ_g)

Algorithm 3 S^3-PnP Inference

Input: Noisy observations, $\{\theta_i, \beta_i\}_{i \in \{s,g\}}$, P_s, ρ, K, $\{t_k\}$

Initialize V^0, U^0, f and $\mathcal{T}_k$
if full-space correction **then**
  $P_s \leftarrow 0, \quad P_g \leftarrow I, \quad \beta_s \leftarrow 0$
  Freeze θ_g ▷ Alg. 1
else
  Fix P_s; $P_g \leftarrow I - P_s$
  Freeze (θ_s, θ_g) ▷ Alg. 2
end if
▷ Trained parameters remain fixed
for $k = 0, \ldots, K-1$ **do**
  $A^{k+1} \leftarrow \text{prox}_{f/\rho}(V^k - U^k)$
  $C^k \leftarrow A^{k+1} + U^k$
  $s_s^k \leftarrow 0, \quad s_g^k \leftarrow 0$
  for $i \in \{s, g\}$ with $\beta_i > 0$ **do**
    $s_i^k \leftarrow s_i(C^k, t_k; \theta_i)$
  end for
  $V^{k+1} \leftarrow \mathcal{T}_k(C^k + \beta_s s_s^k + \beta_g s_g^k)$
  $U^{k+1} \leftarrow U^k + A^{k+1} - V^{k+1}$
end for
return $\widehat{A} = V^K$

4.1 DATASETS AND EXPERIMENTAL SETTINGS

Datasets and Experimental Settings. We evaluate reference-guided hyperspectral restoration on the Dongying dataset from HiPandas (Xu et al., 2025), using a noisy low-resolution HSI as input and a PAN image as guidance. For single-image hyperspectral denoising, we use Qingpu and Xiong'an, with a single noisy HSI as input. Both tasks are evaluated under Gaussian and mixed noise at varying intensity levels. MRI denoising is evaluated on BrainWeb at the official 3%, 5%, and 7% noise levels and on M4Raw using acquired noisy images from all 25 test subjects. For all learning-based methods without pretrained weights, self-supervised training is performed at test time using only the noisy test observations. Dataset and implementation details are provided in Appendix B.5. We report PSNR and SSIM for all tasks, supplemented by SAM and ERGAS for both hyperspectral tasks; higher PSNR/SSIM and lower SAM/ERGAS indicate better performance.

Baselines. For reference-guided hyperspectral restoration, we follow the baseline setup of UHiPandas (Xu et al., 2025). Comparisons include UHiPandas and cascaded pipelines constructed from R-PNN (Guarino et al., 2024), HIR-Diff (Pang et al., 2024), PWRCTV (Xu et al., 2024), PLRDiff (Rui et al., 2024), and SwinIR (Liang et al., 2021). For single-image hyperspectral denoising, baselines include BM4D (Maggioni et al., 2013), HLRTF (Luo et al., 2022), RCTV (Peng et al., 2022), DDS2M (Miao et al., 2023), LRTFR (Luo et al., 2024), HIR-Diff (Pang et al., 2024), and CF-INR (Yu et al., 2026). For MRI denoising, baselines include Noise2Self (Batson & Royer, 2019), Noise2Void (Krull et al., 2019), Noisier2Noise (Moran et al., 2020), R2R (Pang et al., 2021), Noise2Score (Kim & Ye, 2021), ZS-N2N (Mansour & Heckel, 2023), C2S (Tu et al., 2025), and DRDD (Lin et al., 2026). In our comparisons, HIR-Diff, PLRDiff, and DRDD use pretrained diffusion weights, whereas DDS2M performs self-supervised learning during reverse diffusion and C2S trains a denoising predictor through generalized denoising score matching.

4.2 Quantitative and Qualitative Results

Quantitative comparisons. Tables 1–3 and Appendix C.2 summarize the results for the three restoration tasks. For reference-guided HSI restoration, S^3-PnP achieves better average restoration performance than the compared cascaded methods and UHiPandas in both noise groups, with improved spectral fidelity under mixed noise. Similarly, for single-image HSI denoising on Qingpu, it outperforms the reported baselines under both Gaussian and mixed noise. For MRI denoising, S^3-PnP achieves the best reported average performance at all three BrainWeb noise levels. On M4Raw, S^3-PnP learns only at test time and achieves results comparable to pretrained C2S. S^3-PnP generally outperforms the evaluated pretrained diffusion methods and the self-supervised diffusion method.

Table 1: Reference-guided HSI restoration on Dongying. Best in bold; second best underlined.

Method	**Gaussian Mean**				**Mixed Mean**			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
R-PNN + HIR-Diff	30.83	0.7226	8.80	22.55	29.83	0.7139	11.44	31.40
HIR-Diff + SwinIR	32.42	0.7431	7.00	19.42	29.76	0.6776	11.33	34.18
HIR-Diff + R-PNN	35.32	0.9199	6.08	15.35	33.12	0.8995	9.75	<u>29.67</u>
PWRCTV + PLRDiff	36.13	0.8675	6.23	21.54	32.76	0.7747	13.59	153.69
PWRCTV + SwinIR	35.55	0.8773	4.66	14.65	33.80	0.8403	9.90	160.43
UHiPandas	<u>38.24</u>	<u>0.9368</u>	<u>4.05</u>	<u>10.87</u>	<u>37.17</u>	<u>0.9294</u>	<u>7.10</u>	32.91
S^3-PnP (Ours)	**39.75**	**0.9497**	**3.45**	**9.20**	**38.38**	**0.9438**	**4.53**	**11.24**

Table 2: Single-image HSI denoising on Qingpu. Best in bold; second best underlined.

Method	**Gaussian Mean**				**Mixed Mean**			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
BM4D	33.95	0.8214	3.61	9.18	24.56	0.4833	17.50	68.67
HLRTF	34.94	0.7733	3.07	8.97	31.23	0.7944	27.69	117.96
RCTV	28.59	0.5180	7.34	19.50	34.01	0.8741	5.49	16.13
DDS2M	39.97	0.9504	1.40	4.21	29.79	0.8536	13.12	53.51
LRTFR	37.24	0.9020	1.78	6.17	30.84	0.8442	27.00	114.55
HIR-Diff	38.47	0.9157	1.68	4.85	29.06	0.8260	13.20	55.59
CF-INR	<u>40.32</u>	<u>0.9541</u>	<u>1.36</u>	<u>4.01</u>	<u>35.86</u>	<u>0.9339</u>	<u>2.62</u>	<u>8.50</u>
S^3-PnP (Ours)	**40.47**	**0.9562**	**1.33**	**3.95**	**37.88**	**0.9433**	**1.89**	**5.34**

Qualitative comparisons. Figure 3 presents visual comparisons for the three restoration tasks. For reference-guided HSI restoration on Dongying, S^3-PnP recovers clearer spatial boundaries and colors closer to the reference, while several cascaded methods exhibit color shifts or residual artifacts. For single-image HSI denoising on Qingpu, it suppresses noise while preserving fine spatial structures, whereas some competing methods leave visible noise or blur local details. On BrainWeb and M4Raw, it reduces noise while retaining tissue boundaries and local anatomical structures. The enlarged regions highlight these differences in noise suppression and detail preservation, and the error maps reveal the remaining discrepancies from the references.

Table 3: MRI denoising. Best in bold; second best underlined.

Method	BrainWeb						M4Raw					
	3%		5%		7%		T1		T2		FLAIR	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Noise2Self	31.22	0.8613	30.40	0.8396	29.49	0.7952	29.10	0.8537	28.17	0.8095	28.94	0.8150
Noise2Void	32.11	0.8810	30.40	0.8522	29.23	0.8110	31.24	0.8671	30.45	0.8392	30.80	0.8288
Noisier2Noise	34.38	0.8964	31.37	0.8341	28.90	0.7410	31.18	0.8675	30.55	0.8435	31.17	0.8356
R2R	34.02	0.8011	32.03	0.7879	29.50	0.7292	30.82	0.8552	30.48	0.8329	30.97	0.8187
Noise2Score	34.47	0.8405	29.95	0.7621	26.85	0.7133	31.23	0.8444	30.48	0.8139	31.00	0.7975
ZS-N2N	32.63	0.7768	28.73	0.7165	26.07	0.6749	31.66	0.8922	30.96	0.8729	31.89	0.8765
C2S	35.21	0.9287	33.30	0.8504	31.88	0.8173	32.61	0.9152	32.30	0.8880	**33.02**	0.8792
DRDD	35.03	0.8188	30.70	0.7581	27.83	0.7233	32.43	0.9108	31.44	0.8770	32.60	0.8797
S^3-PnP (Ours)	**37.47**	**0.9568**	**34.65**	**0.9413**	**32.72**	**0.9279**	**32.69**	**0.9165**	**32.31**	**0.8922**	32.83	**0.8906**

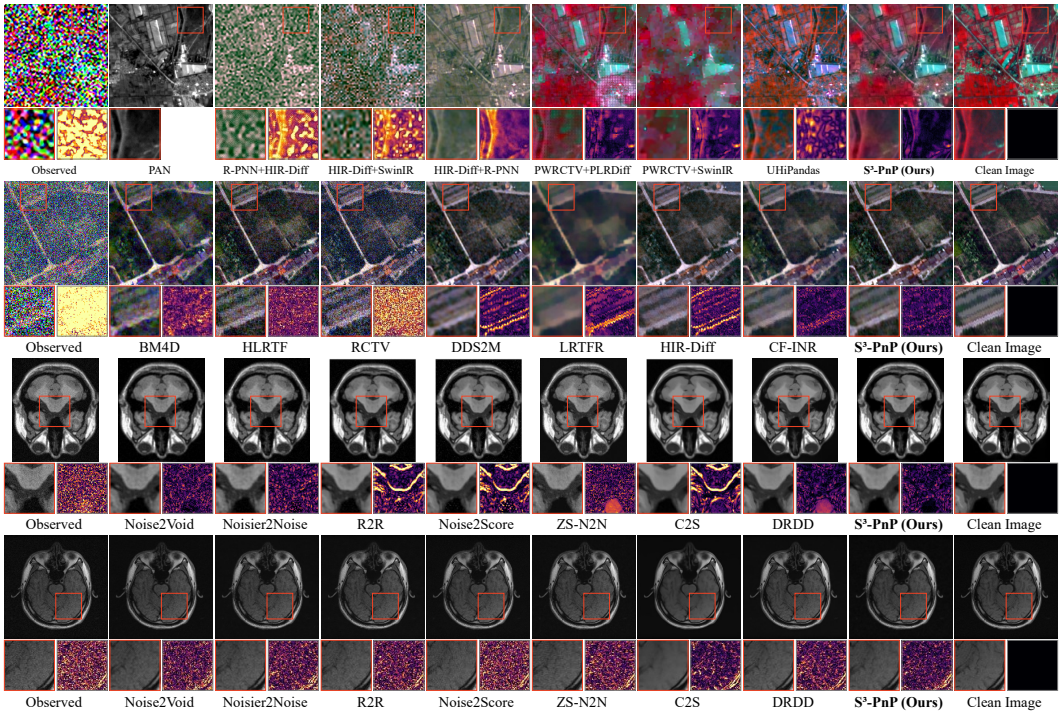

Figure 3: Qualitative comparisons on Dongying (GNI), Qingpu (G50), BrainWeb (T1, 7% noise), and M4Raw (T1), from top to bottom. Enlarged regions and corresponding error maps are shown below each reconstruction.

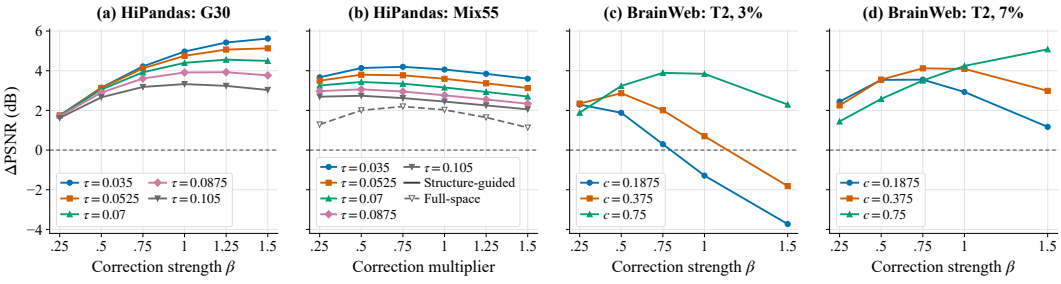

Figure 4: PSNR gains over analytical ADMM at the same prior setting, varying τ in (a)–(b) and c in (c)–(d). In (b), solid curves denote structure-guided correction and the dashed curve denotes full-space correction at $\tau = 0.105$; the horizontal axis scales their respective reference gains. BrainWeb results average three development slices per condition.

4.3 Ablation Studies

Contribution of learned corrections. We compare each trained configuration with analytical ADMM at the same prior setting. The analytical reference uses the initial observation-derived estimates, and network weights remain fixed during each correction-strength sweep. Figure 4(a)–(b) shows positive gains throughout the tested Dongying grids. On BrainWeb, all tested settings improve performance at 7% noise, while strong corrections under weaker priors can reduce performance at 3% noise (c)–(d). These results show that learned corrections can improve analytical ADMM, but their strength should be selected jointly with the analytical prior and noise level.

Structure-guided learning. We compare structure-guided and full-space learning at the same prior strength $\tau = 0.105$. The structure-guided configuration achieves larger gains across all tested correction multipliers (Figure 4(b)). Since the training configurations also differ, this comparison does not isolate the contribution of orthogonal projection alone.

4.4 Consistency with the Theoretical Analysis

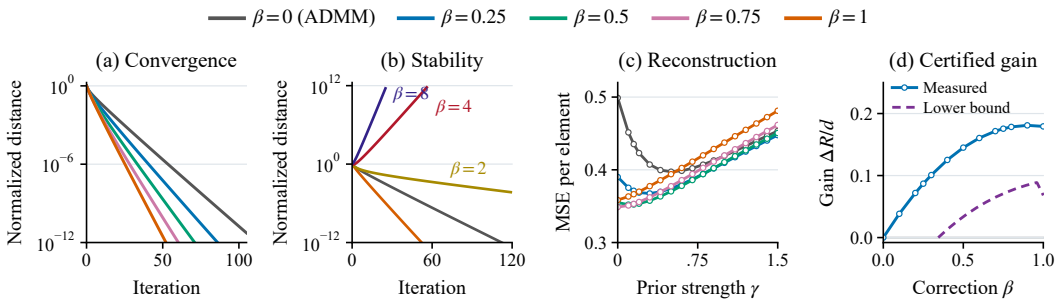

Figure 5: Controlled experiments with 64×64 Gaussian states and ideal scores. (a)–(c) Wide-spectrum fidelity; (d) near-matched fidelity with a theoretical lower bound on error reduction. The relative prior strength $\gamma = b/a_t$ is fixed at 0.15 in (a)–(b) and 0.1 in (d). Shading in (c)–(d) indicates pointwise 95% confidence intervals.

We examine Theorem 1 using matched solvers with positive-definite quadratic fidelity and exact singular-value soft thresholding. Under wide-spectrum fidelity, the tested corrections accelerate the empirical decay of the normalized distance between trajectories from two initializations (Figure 5(a)): at $\beta = 1$, the distance reaches 10^{-10} in 44 iterations, compared with 93 for analytical ADMM. Some settings outside the sufficient contraction region still show decaying distances, whereas $\beta = 4, 8$, beyond the theorem's range, produce growing distances (b). Moderate corrections ($\beta = 0.25, 0.5$) reduce reconstruction error across the tested prior strengths, whereas stronger corrections can increase error under strong analytical regularization (c). Under near-matched fidelity, all eight sampled settings satisfying the improvement condition yield empirical gains above the positive theoretical lower bound (d), consistent with the predicted improvement and illustrating the bound's conservatism. These results support balancing score correction with analytical regularization. Appendix C.1 provides full settings and results across dimensions.

5 Conclusion

We presented S^3-PnP, a self-supervised PnP-ADMM framework for scientific image restoration. By learning score corrections from ADMM states generated by the available observations, the framework connects prior learning to the restoration process. These corrections refine the inputs to analytical prior updates, complementing analytical regularization without clean training targets or externally pretrained weights. We further analyzed the convergence and reconstruction error of the corrected ADMM algorithm under ideal scores and quadratic data fidelity. Through task-specific observation models, the framework supports reference-guided HSI restoration, single-image HSI denoising, and MRI denoising. Our work provides a common approach to integrating observation-driven score learning with analytical optimization for scientific imaging.

AI USE STATEMENT

In this work, we used generative AI tools solely to improve the grammar, clarity, and readability of text written by the authors. Generative AI was not used to generate research ideas, formulate hypotheses or claims, design the methodology or experiments, implement the method, analyze data, or interpret results. The authors reviewed all AI-assisted edits and take full responsibility for the final content of this work.

REFERENCES

Joshua Batson and Loïc Royer. Noise2self: Blind denoising by self-supervision. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 524–533, 2019.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. doi: 10.1561/2200000016.

Stanley H. Chan, Xiran Wang, and Omar A. Elgendy. Plug-and-play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1): 84–98, 2017. doi: 10.1109/TCI.2016.2629286.

Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated MRI. *Medical Image Analysis*, 80:102479, 2022. doi: 10.1016/j.media.2022.102479.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *International Conference on Learning Representations*, 2024.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181.

Ziqi Gao and S. Kevin Zhou. Mrpd: Undersampled mri reconstruction by prompting a large latent diffusion model. *arXiv preprint arXiv:2402.10609*, 2024.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2398–2407, 2025.

Giuseppe Guarino, Matteo Ciotola, Gemine Vivone, and Giuseppe Scarpa. Band-wise hyperspectral image pansharpening using CNN model propagation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–18, 2024. doi: 10.1109/tgrs.2023.3339337.

Merve Gülle, Junno Yun, Yaşar Utku Alçalar, and Mehmet Akçakaya. PnP-CM: Consistency models as plug-and-play priors for inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38464–38474, 2026.

Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9483–9505, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23593–23606, 2022. doi: 10.52202/068431-1714.

Kwanyoung Kim and Jong Chul Ye. Noise2Score: Tweedie's approach to self-supervised image denoising without clean images. In *Advances in Neural Information Processing Systems*, volume 34, pp. 864–874, 2021.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2129–2137, 2019.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 1833–1844, 2021.

Ziyue Lin, Jiahe Hou, Hongyu Xia, Xinrui Xie, Feifei Wang, Yuyin Zhou, Wei Wang, Jiawei Liu, and Liangqiong Qu. Decoupled residual denoising diffusion models for unified and data efficient image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35967–35977, 2026.

Yisi Luo, Xi-Le Zhao, Deyu Meng, and Tai-Xiang Jiang. HLRTF: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19303–19312, 2022.

Yisi Luo, Xile Zhao, Zhemin Li, Michael K. Ng, and Deyu Meng. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3351–3369, 2024. doi: 10.1109/TPAMI.2023.3341688.

Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, and Alessandro Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133, 2013. doi: 10.1109/TIP.2012.2210725.

Youssef Mansour and Reinhard Heckel. Zero-Shot Noise2Noise: Efficient image denoising without any data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14018–14027, 2023.

Ségolène Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. PnP-Flow: Plug-and-play image restoration with flow matching. In *International Conference on Learning Representations*, pp. 45466–45492, 2025.

Yuchun Miao, Lefei Zhang, Liangpei Zhang, and Dacheng Tao. DDS2M: Self-supervised denoising diffusion spatio-spectral model for hyperspectral image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12052–12062, 2023.

Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2Noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12061–12069, 2020. doi: 10.1109/cvpr42600.2020.01208.

Li Pang, Xiangyu Rui, Long Cui, Hongzhong Wang, Deyu Meng, and Xiangyong Cao. HIR-Diff: Unsupervised hyperspectral image restoration via improved diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3005–3014, 2024.

Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2043–2052, 2021.

Jiangjun Peng, Hailin Wang, Xiangyong Cao, Xinling Liu, Xiangyu Rui, and Deyu Meng. Fast noise removal in hyperspectral images via representative coefficient total variation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2022. doi: 10.1109/TGRS.2022.3229012.

Xiangyu Rui, Xiangyong Cao, Li Pang, Zeyu Zhu, Zongsheng Yue, and Deyu Meng. Unsupervised hyperspectral pansharpening via low-rank diffusion model. *Information Fusion*, 107:102325, 2024. doi: 10.1016/j.inffus.2024.102325.

Ernest K. Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-play methods provably converge with properly trained denoisers. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5546–5557, 2019.

Shirin Shoushtari, Jiaming Liu, Edward P. Chandler, M. Salman Asif, and Ulugbek S. Kamilov. Prior mismatch and adaptation in PnP-ADMM with a nonconvex convergence analysis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45154–45182, 2024.

Rajesh Shrestha and Xiao Fu. Taming score-based denoisers in ADMM: A convergent plug-and-play framework. In *International Conference on Learning Representations*, pp. 13523–13568, 2026.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

Jiachen Tu, Yaokun Shi, and Fan Lam. Score-based self-supervised MRI denoising. In *International Conference on Learning Representations*, pp. 102262–102293, 2025.

Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, 2013. doi: 10.1109/GlobalSIP.2013.6737048.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations*, 2023.

Chenxu Wu, Qingpeng Kong, Zihang Jiang, and S. Kevin Zhou. Self-supervised diffusion MRI denoising via iterative and stable refinement. In *International Conference on Learning Representations*, pp. 98708–98747, 2025.

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine L. Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. In *Advances in Neural Information Processing Systems*, volume 37, pp. 118389–118427, 2024. doi: 10.52202/079017-3759.

Tiange Xiang, Mahmut Yurt, Ali B. Syed, Kawin Setsompop, and Akshay Chaudhari. DDM^2: Self-supervised diffusion MRI denoising with generative diffusion models. In *International Conference on Learning Representations*, 2023.

Jin-Liang Xiao, Ting-Zhu Huang, Liang-Jian Deng, Guang Lin, Zihan Cao, Chao Li, and Qibin Zhao. Hyperspectral pansharpening via diffusion models with iteratively zero-shot guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12669–12678, 2025.

Shuang Xu, Qiao Ke, Jiangjun Peng, Xiangyong Cao, and Zixiang Zhao. Pan-denoising: Guided hyperspectral image denoising via weighted represent coefficient total variation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024. doi: 10.1109/tgrs.2024.3450888.

Shuang Xu, Zixiang Zhao, Haowen Bai, Chang Yu, Jiangjun Peng, Xiangyong Cao, and Deyu Meng. Hipandas: Hyperspectral image joint denoising and super-resolution by image fusion with the panchromatic image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12002–12011, 2025.

Chang Yu, Yisi Luo, Kai Ye, Xile Zhao, and Deyu Meng. Cross-frequency implicit neural representation with self-evolving parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(8):9467–9485, 2026. doi: 10.1109/tpami.2026.3679791.

Youyuan Zhang, Zehua Liu, Zenan Li, Zhaoyu Li, James J. Clark, and Xujie Si. Decoupling training-free guided diffusion by ADMM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23292–23302, 2025.

Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen, and Matthew S. Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1219–1229, 2023.

A PROOFS

A.1 PROOF OF LEMMA 1

Lemma 1 (GDSM regression on ADMM states; restated)**.** *Let C be an ADMM denoiser input drawn from a fixed reference training distribution, and let $Z = \mathcal{S}(C) = (C - m)/s$, where the channel-wise median m and positive scale s are estimated from the training set and fixed; / denotes element-wise division. Assume $\mathbb{E}\|Z\|^2 < \infty$. For fixed t, construct $Z_t = Z + \sqrt{\Delta_t}\epsilon$, where $\Delta_t > 0$ and $\epsilon \sim \mathcal{S}(0, I)$ is independent of Z. Let p_t denote the density of Z_t and define $R_t = -\log p_t$. For $\kappa_t > \Delta_t$, consider the GDSM interpolation loss (Tu et al., 2025)*

$$\mathcal{L}_{\mathrm{GDSM}}(\widehat{Z}; t) = \mathbb{E}\left\| \frac{\Delta_t}{\kappa_t} \widehat{Z}(Z_t, t) + \left(1 - \frac{\Delta_t}{\kappa_t}\right) Z_t - Z \right\|^2 .$$

Its minimizer over square-integrable restoration predictors satisfies, almost surely,

$$\widehat{Z}^\star(Z_t, t) = Z_t + \kappa_t s_t^\star(Z_t),$$

where $s_t^\star = \nabla \log p_t = -\nabla R_t$. Moreover, $\mathbb{E}[Z \mid Z_t] = Z_t + \Delta_t s_t^\star(Z_t)$. If additionally $Z = Z_{\mathrm{clean}} + \sigma_0 \epsilon_0$, where $\sigma_0 > 0$, $\epsilon_0 \sim \mathcal{N}(0, I)$, and $Z_{\mathrm{clean}}, \epsilon_0, \epsilon$ are mutually independent, choosing $\kappa_t = \sigma_0^2 + \Delta_t$ gives $\widehat{Z}^\star(Z_t, t) = \mathbb{E}[Z_{\mathrm{clean}} \mid Z_t]$. For conditional networks, all densities, expectations, and independence assumptions are understood conditional on the same auxiliary inputs.

Proof. Fix t, the reference distribution, and the normalization. Any auxiliary inputs are held fixed throughout. Assume $\mathbb{E}\|Z\|^2 < \infty$, and let $Z_t = Z + \sqrt{\Delta_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is independent of Z. We first characterize the optimal interpolated prediction, then derive its score representation using Tweedie's identity, and finally establish the clean-state conditional-mean interpretation.

Determine the optimal interpolated prediction. Consider the GDSM loss over all square-integrable restoration predictors $\widehat{Z}$:

$$\mathcal{L}_{\mathrm{GDSM}}(\widehat{Z}; t) = \mathbb{E}\left\| \frac{\Delta_t}{\kappa_t} \widehat{Z}(Z_t, t) + \left(1 - \frac{\Delta_t}{\kappa_t}\right) Z_t - Z \right\|^2 .$$

Since $\Delta_t/\kappa_t > 0$ and Z_t is square-integrable, the affine transformation from the restoration prediction to its interpolation is invertible within the class of square-integrable predictors. Minimizing the GDSM loss is therefore equivalent to regressing the interpolated prediction onto Z under squared error. By the conditional-mean characterization of optimal squared-error regression, the minimizer satisfies

$$\mathbb{E}[Z \mid Z_t] = \frac{\Delta_t}{\kappa_t} \widehat{Z}^\star(Z_t, t) + \left(1 - \frac{\Delta_t}{\kappa_t}\right) Z_t. \tag{A.1}$$

This determines the optimal interpolation. We next solve for the restoration prediction itself.

Derive the score representation of the optimal prediction. Let p_t denote the density of Z_t and define $s_t^\star = \nabla \log p_t$. For the independent Gaussian perturbation above, Tweedie's identity (Efron, 2011) gives

$$\mathbb{E}[Z \mid Z_t] = Z_t + \Delta_t s_t^\star(Z_t). \tag{A.2}$$

Substituting this identity into Eq. (A.1) and isolating $\widehat{Z}^\star$ yields

$$\begin{aligned} \widehat{Z}^\star(Z_t, t) &= \frac{\kappa_t}{\Delta_t} \mathbb{E}[Z \mid Z_t] - \left(\frac{\kappa_t}{\Delta_t} - 1\right) Z_t \\ &= Z_t + \kappa_t s_t^\star(Z_t). \end{aligned} \tag{A.3}$$

Thus, the optimal restoration predictor takes the form of the input plus a scaled state score. This form follows from minimizing the interpolation loss, without imposing a score parameterization beforehand. Furthermore, defining $R_t = -\log p_t$ gives $s_t^\star = -\nabla R_t$, establishing the state-energy interpretation of the ideal correction.

Establish the clean-state conditional-mean interpretation. Under the additional assumption $Z = Z_{\text{clean}} + \sigma_0\epsilon_0$, where Z_{clean}, ϵ_0, and ϵ are mutually independent and $\epsilon_0 \sim \mathcal{N}(0, I)$, we have

$$Z_t = Z_{\text{clean}} + \sigma_0\epsilon_0 + \sqrt{\Delta_t}\epsilon.$$

The clean state is square-integrable because $Z_{\text{clean}} = Z - \sigma_0\epsilon_0$. The combined noise is Gaussian, independent of Z_{clean}, and has covariance $(\sigma_0^2+\Delta_t)I$. Applying Tweedie's identity to this observation model therefore gives

$$\mathbb{E}[Z_{\text{clean}} \mid Z_t] = Z_t + (\sigma_0^2 + \Delta_t)s_t^\star(Z_t).$$

The score is unchanged because the marginal distribution of Z_t remains p_t. Choosing $\kappa_t = \sigma_0^2 + \Delta_t$ and using Eq. (A.3), we conclude that

$$\widehat{Z}^\star(Z_t, t) = \mathbb{E}[Z_{\text{clean}} \mid Z_t].$$

This proves the final statement. Neither the clean-state distribution nor p_t is required to be Gaussian. Thus, the optimal GDSM prediction equals the input plus a scaled state score and, under the additional Gaussian residual assumption and matched coefficient, coincides with the conditional mean of the clean state. □

A.2 Proof of Proposition 1

We restate the statistical construction and give the constants used in the proposition. Let $Z = Z_0 + \sigma_0\epsilon_0$ and $Z_t = Z + \sqrt{\Delta_t}\epsilon$, where $\sigma_0, \Delta_t > 0$, the two noises are standard Gaussian, and $Z_0, \epsilon_0, \epsilon$ are mutually independent. Assume $\mathbb{E}\|Z\|^2 < \infty$. Fix the reference distribution, smoothing level, normalization $\mathcal{S}(C) = (C - m)/s$ with $s > 0$, and analytical map throughout the iterations and statistical expectations. All statements are conditional on any fixed auxiliary inputs.

The clean state in the original coordinates is $A_0 = \mathcal{S}^{-1}(Z_0)$, and its noisy input is $Y = \mathcal{S}^{-1}(Z_t)$. Thus, the subscript 0 in A_0 and Z_0 denotes the same clean state in different coordinates. Define the posterior-mean estimator $\bar{A}(Y) = \mathbb{E}[A_0 \mid Y]$, its distance from the input $a_t = (\mathbb{E}\|Y - \bar{A}(Y)\|^2)^{1/2}$, and the matched coefficient $\kappa_t = \sigma_0^2 + \Delta_t$. The ideal score $s_t^\star = \nabla \log p_t$ induces the original-coordinate correction and prior map

$$d_t(C) = \kappa_t s \odot s_t^\star(\mathcal{S}(C)), \qquad \mathcal{C}_\beta(C) = \mathcal{T}(C + \beta d_t(C)), \qquad \beta \in [0, 1].$$

This is Eq. (6) with $s_\theta = s_t^\star$ and $\eta_k = \beta\kappa_t$; $\beta = 0$ disables the correction. Matrices are treated as elements of a finite-dimensional Hilbert space with the Frobenius inner product, and $\|\cdot\|_{\text{op}}$ denotes the induced operator norm.

Proposition 1 (Ideal GDSM under quadratic fidelity; expanded statement)**.** *Under the construction above, use exact ADMM with fixed penalty $\rho > 0$ and $f(A) = \frac{1}{2}\langle A - Y, H(A - Y)\rangle$, where $H = H^* \succ 0$ is fixed. Suppose d_t is L-Lipschitz and $\mathcal{T}$ is firmly nonexpansive. Define $r_H = \|(\rho I - H)(\rho I + H)^{-1}\|_{\text{op}}$. For any $\beta \in [0, 1]$ satisfying*

$$r_H(1 + 2\beta L) < 1, \tag{A.4}$$

the ADMM iterates converge from any finite initialization, with

$$\|C_\beta^k - C_\beta^\star\| \le q_\beta^k\|C_\beta^0 - C_\beta^\star\|, \qquad q_\beta = \tfrac{1}{2}[1 + r_H(1 + 2\beta L)] < 1. \tag{A.5}$$

Write $A_\beta^\star = \lim_k A_\beta^k = \lim_k V_\beta^k$ for the primal limit. Denote the limit with $\beta = 0$ by $A_{\text{off}}^\star$, to distinguish the correction-disabled estimate from the clean state A_0.

For the risk comparison, additionally assume $\|\mathcal{T}(Q) - Q\| \le b$ for all Q. Set $M = \rho H^{-1}$, $b_H = \|M\|_{\text{op}}b$, and $\delta_\beta = \beta L\|M\|_{\text{op}}\|I - H/\rho\|_{\text{op}}$. When $\delta_\beta < 1$, define $c_\beta = \|I - \beta M\|_{\text{op}} + \beta\|M\|_{\text{op}}\delta_\beta/(1 - \delta_\beta)$. For $\beta > 0$, the sufficient gain conditions

$$\delta_\beta < 1, \qquad c_\beta a_t + \frac{b_H}{1 - \delta_\beta} < a_t - b_H \tag{A.6}$$

ensure

$$\mathbb{E}\|A_\beta^\star - A_0\|^2 < \mathbb{E}\|A_{\text{off}}^\star - A_0\|^2.$$

The two solvers use the same data term, analytical map, and penalty; only the score correction changes.

Proof. The proof has three parts. First, we identify the useful correction supplied by the ideal score. Second, we establish convergence of the complete ADMM iteration. Third, we control the effect of the analytical map and ADMM feedback to compare the limiting reconstruction risks.

Relating the ideal score correction to the posterior mean. Lemma 1 gives $\mathbb{E}[Z_0 \mid Z_t] = Z_t + \kappa_t s_t^\star(Z_t)$ under the matched coefficient. Since $\mathcal{S}$ is fixed, affine, and invertible, conditioning on Y is equivalent to conditioning on Z_t.

Here $d_t(Y) = \kappa_t s \odot s_t^\star(\mathcal{S}(Y))$ denotes the ideal score correction rescaled to the original coordinates, with s denoting the fixed normalization scale. Inverse normalization multiplies by s and adds m, so the normalized input-plus-score prediction becomes $Y + d_t(Y)$. Using this rescaling and the fact that a fixed affine map commutes with conditional expectation, we obtain

$$\begin{aligned} \bar{A}(Y) &= \mathbb{E}[A_0 \mid Y] \\ &= \mathcal{S}^{-1}[\mathbb{E}[Z_0 \mid Z_t]] \\ &= \mathcal{S}^{-1}[\mathcal{S}(Y) + \kappa_t s_t^\star(\mathcal{S}(Y))] \\ &= Y + \kappa_t s \odot s_t^\star(\mathcal{S}(Y)) = Y + d_t(Y). \end{aligned} \tag{A.7}$$

Thus, $d_t(Y) = \bar{A}(Y) - Y$: the ideal correction at Y points toward the posterior mean, which minimizes expected squared reconstruction error. Substituting this identity gives

$$Y + \beta d_t(Y) = (1-\beta)Y + \beta \bar{A}(Y) =: M_\beta. \tag{A.8}$$

The moment assumption makes A_0, Y, and $\bar{A}(Y)$ square-integrable, and $\mathbb{E}\|d_t(Y)\|^2 = a_t^2$. The quadratic fidelity is centered at this same Y, so the correction identified here is calibrated to the input used in the risk comparison.

Establishing ADMM convergence. Fix $\beta \in [0,1]$ satisfying Eq. (A.4). Suppress the subscript β on intermediate iterates for readability. The prior update is

$$V^{k+1} = \mathcal{C}_\beta(C^k) = \mathcal{T}(C^k + \beta d_t(C^k)). \tag{A.9}$$

Write $h = HY$. The first-order condition for the exact data update is $HA - h + \rho(A - V^k + U^k) = 0$, giving

$$A^{k+1} = (H + \rho I)^{-1}\big[h + \rho(V^k - U^k)\big]. \tag{A.10}$$

Using $C^k = A^{k+1} + U^k$ and $U^{k+1} = C^k - V^{k+1}$, the next prior input satisfies

$$\begin{aligned} C^{k+1} &= A^{k+2} + U^{k+1} \\ &= (H + \rho I)^{-1}\big[h + \rho V^{k+1} + HU^{k+1}\big] \\ &= (H + \rho I)^{-1}\big[h + HC^k + (\rho I - H)\mathcal{C}_\beta(C^k)\big] =: \mathcal{F}_\beta(C^k). \end{aligned} \tag{A.11}$$

To prove convergence, we show that the complete ADMM map $\mathcal{F}_\beta$ contracts distances between inputs. We use a reflected-operator representation (Ryu et al., 2019) to exploit the firm nonexpansiveness of $\mathcal{T}$: its reflection $2\mathcal{T} - I$ is nonexpansive, which allows us to bound the effect of the added score correction. Define $R(v) = 2(H + \rho I)^{-1}(h + \rho v) - v$ and $R_\beta = 2\mathcal{C}_\beta - I$ as the reflections of the data and prior updates, respectively. Substitution gives

$$\begin{aligned} \tfrac{1}{2}\big[C + R(R_\beta(C))\big] &= (H + \rho I)^{-1}\big[h + \rho R_\beta(C)\big] + \tfrac{1}{2}\big[C - R_\beta(C)\big] \\ &= (H + \rho I)^{-1}\big[h + \rho(2\mathcal{C}_\beta(C) - C)\big] + C - \mathcal{C}_\beta(C) \\ &= (H + \rho I)^{-1}\big[h + HC + (\rho I - H)\mathcal{C}_\beta(C)\big] = \mathcal{F}_\beta(C). \end{aligned}$$

Thus Eq. (A.11) has the equivalent representation

$$\begin{aligned} \mathcal{F}_\beta &= \tfrac{1}{2}(I + R \circ R_\beta), \\ R(v) &= (\rho I - H)(\rho I + H)^{-1} v + 2(H + \rho I)^{-1} h. \end{aligned} \tag{A.12}$$

We now bound the changes in R and R_β separately to establish contraction of $\mathcal{F}_\beta$. Positive definiteness of H implies

$$\mathrm{Lip}(R) = r_H = \max_j \left| \frac{\rho - \lambda_j(H)}{\rho + \lambda_j(H)} \right| < 1.$$

Firm nonexpansiveness of $\mathcal{T}$ means $\|\mathcal{T}(x) - \mathcal{T}(y)\|^2 \le \langle \mathcal{T}(x) - \mathcal{T}(y), x - y\rangle$, and therefore $R_{\mathcal{T}} = 2\mathcal{T} - I$ is nonexpansive. Using $\mathcal{C}_\beta(C) = \mathcal{T}(C + \beta d_t(C))$ and adding and subtracting the corrected input yields

$$\begin{aligned} R_\beta(C) &= 2\mathcal{C}_\beta(C) - C \\ &= 2\mathcal{T}(C + \beta d_t(C)) - C \\ &= \big[2\mathcal{T}(C + \beta d_t(C)) - (C + \beta d_t(C))\big] + \beta d_t(C) \\ &= R_{\mathcal{T}}(C + \beta d_t(C)) + \beta d_t(C). \end{aligned} \tag{A.13}$$

For any two inputs C, C', the triangle inequality and nonexpansiveness of $R_{\mathcal{T}}$ then give

$$\begin{aligned} &\|R_\beta(C) - R_\beta(C')\| \\ &\le \|R_{\mathcal{T}}(C + \beta d_t(C)) - R_{\mathcal{T}}(C' + \beta d_t(C'))\| + \beta\|d_t(C) - d_t(C')\| \\ &\le \|(C - C') + \beta(d_t(C) - d_t(C'))\| + \beta\|d_t(C) - d_t(C')\| \\ &\le \|C - C'\| + 2\beta\|d_t(C) - d_t(C')\| \\ &\le (1 + 2\beta L)\|C - C'\|. \end{aligned} \tag{A.14}$$

The last inequality uses the L-Lipschitz property of d_t. Thus $\mathrm{Lip}(R_\beta) \le 1 + 2\beta L$. Combining this bound with $\mathrm{Lip}(R) = r_H$ in Eq. (A.12), we obtain

$$\begin{aligned} &\|\mathcal{F}_\beta(C) - \mathcal{F}_\beta(C')\| \\ &= \tfrac{1}{2}\big\|(C - C') + R(R_\beta(C)) - R(R_\beta(C'))\big\| \\ &\le \tfrac{1}{2}\|C - C'\| + \tfrac{1}{2}\|R(R_\beta(C)) - R(R_\beta(C'))\| \\ &\le \tfrac{1}{2}\|C - C'\| + \tfrac{r_H}{2}\|R_\beta(C) - R_\beta(C')\| \\ &\le \tfrac{1}{2}\|C - C'\| + \tfrac{r_H}{2}(1 + 2\beta L)\|C - C'\| \\ &= \tfrac{1}{2}[1 + r_H(1 + 2\beta L)]\|C - C'\| = q_\beta\|C - C'\|. \end{aligned} \tag{A.15}$$

The inequalities successively use the triangle inequality, the bound for R, and Eq. (A.14). Since $q_\beta < 1$, the contraction principle gives a unique fixed point $C^\star_\beta$ and the linear convergence bound in Eq. (A.5). Continuity of $\mathcal{C}_\beta$ and the ADMM updates then imply $A^k, V^k \to A^\star_\beta$ and $U^k \to C^\star_\beta - A^\star_\beta$, with

$$A^\star_\beta = \mathcal{C}_\beta(C^\star_\beta), \qquad \rho C^\star_\beta = h + (\rho I - H)A^\star_\beta. \tag{A.16}$$

This also covers $\beta = 0$, for which $q_0 = (1 + r_H)/2 < 1$.

Comparing the limiting reconstruction risks. Part 2 establishes the limiting estimates. We now compare their expected reconstruction errors using the posterior mean $\bar{A}(Y) = \mathbb{E}[A_0 \mid Y]$ as a common reference. For any square-integrable estimator $\widehat{A}(Y)$, expanding $\widehat{A}(Y) - A_0 = (\widehat{A}(Y) - \bar{A}(Y)) + (\bar{A}(Y) - A_0)$ gives

$$\begin{aligned} \mathbb{E}\|\widehat{A}(Y) - A_0\|^2 &= R_{\mathrm{MMSE}} + \mathbb{E}\|\widehat{A}(Y) - \bar{A}(Y)\|^2, \\ R_{\mathrm{MMSE}} &= \mathbb{E}\|\bar{A}(Y) - A_0\|^2. \end{aligned} \tag{A.17}$$

The cross term vanishes since $\mathbb{E}[\bar{A}(Y) - A_0 \mid Y] = 0$. The two ADMM limits are measurable functions of Y, as they can be obtained by iterating the fixed continuous maps from a fixed initialization. Their square integrability will be verified below. With this property, applying Eq. (A.17) to both estimates cancels the common R_{MMSE} and gives

$$\begin{aligned} &\mathbb{E}\|A^\star_{\mathrm{off}} - A_0\|^2 - \mathbb{E}\|A^\star_\beta - A_0\|^2 \\ &\qquad = \mathbb{E}\|A^\star_{\mathrm{off}} - \bar{A}(Y)\|^2 - \mathbb{E}\|A^\star_\beta - \bar{A}(Y)\|^2. \end{aligned} \tag{A.18}$$

Our goal is therefore to make the difference on the right strictly positive. We derive a lower bound for its first expectation and an upper bound for its second, then compare them using the gain condition. For the intermediate estimates, write $\|X\|_{L^2} := (\mathbb{E}\|X\|^2)^{1/2}$, so $a_t = \|Y - \bar{A}(Y)\|_{L^2}$. We first bound these root-mean-square distances using triangle inequalities, then square the bounds to compare the expectations.

We begin with the correction-disabled estimate. To determine how far it can move from Y, we express the output displacement using the fixed-point equations from Part 2. Let $B = I - H/\rho$. Since $h = HY$, Eq. (A.16) gives, for either solver,

$$C_\beta^\star - Y = B(A_\beta^\star - Y). \tag{A.19}$$

Define the displacement of the analytical map at its actual input by

$$e_\beta = \mathcal{T}(C_\beta^\star + \beta d_t(C_\beta^\star)) - (C_\beta^\star + \beta d_t(C_\beta^\star)), \qquad \|e_\beta\| \le b.$$

The prior fixed-point equation gives $A_\beta^\star - C_\beta^\star = \beta d_t(C_\beta^\star) + e_\beta$, while Eq. (A.19) gives $A_\beta^\star - C_\beta^\star = (H/\rho)(A_\beta^\star - Y)$. Combining the two identities yields

$$A_\beta^\star - Y = M\big[\beta d_t(C_\beta^\star) + e_\beta\big], \qquad M = \rho H^{-1}. \tag{A.20}$$

Setting $\beta = 0$ gives $A_{\text{off}}^\star - Y = Me_{\text{off}}$, where e_{off} is the corresponding analytical-map displacement. Hence $\|A_{\text{off}}^\star - Y\| \le b_H$, which also makes $A_{\text{off}}^\star$ square-integrable because Y is square-integrable. The reverse-triangle inequality now gives

$$\begin{aligned} \|A_{\text{off}}^\star - \bar{A}(Y)\|_{L^2} &\ge \|Y - \bar{A}(Y)\|_{L^2} - \|A_{\text{off}}^\star - Y\|_{L^2} \\ &\ge a_t - b_H. \end{aligned} \tag{A.21}$$

The gain condition ensures $a_t > b_H$, so squaring this nonnegative lower bound gives $\mathbb{E}\|A_{\text{off}}^\star - \bar{A}(Y)\|^2 \ge (a_t - b_H)^2$. This is the required lower bound for the first expectation in Eq. (A.18).

Nextly, we seek an upper bound for the corrected estimate's distance to the posterior mean. Part 1 identifies $d_t(Y) = \bar{A}(Y) - Y$, but at convergence ADMM applies the score correction to $C_\beta^\star$, which generally differs from Y. Adding and subtracting $\beta M d_t(Y)$ in Eq. (A.20) separates the calibrated score contribution from the remaining effects:

$$\begin{aligned} A_\beta^\star - Y &= \beta M d_t(Y) + \xi_\beta, \\ \xi_\beta &= \beta M\big[d_t(C_\beta^\star) - d_t(Y)\big] + Me_\beta. \end{aligned}$$

Substituting the identity from Part 1 gives

$$A_\beta^\star - \bar{A}(Y) = (I - \beta M)(Y - \bar{A}(Y)) + \xi_\beta.$$

The first term describes how the calibrated score contribution changes the original discrepancy. The remainder ξ_β contains the change in score correction caused by the shifted ADMM input and the displacement due to the analytical map. Bounding this remainder will give the desired upper bound.

For $\beta > 0$, set $D_\beta = A_\beta^\star - Y$. To control the remainder, we first bound this output displacement by separating the correction at Y from the change caused by the ADMM input. Starting from Eq. (A.20), we obtain, pointwise,

$$\begin{aligned} \|D_\beta\| &= \big\|M\big[\beta d_t(C_\beta^\star) + e_\beta\big]\big\| \\ &\le \beta\|M\|_{\text{op}}\|d_t(C_\beta^\star)\| + \|M\|_{\text{op}}\|e_\beta\| \\ &\le \beta\|M\|_{\text{op}}\big(\|d_t(Y)\| + \|d_t(C_\beta^\star) - d_t(Y)\|\big) + b_H \\ &\le \beta\|M\|_{\text{op}}\big(\|d_t(Y)\| + L\|C_\beta^\star - Y\|\big) + b_H \\ &\le \beta\|M\|_{\text{op}}\big(\|d_t(Y)\| + L\|B\|_{\text{op}}\|D_\beta\|\big) + b_H \\ &= \beta\|M\|_{\text{op}}\|d_t(Y)\| + \delta_\beta\|D_\beta\| + b_H. \end{aligned}$$

The first inequality uses the operator-norm bound and the triangle inequality. The second adds and subtracts $d_t(Y)$ and uses $\|e_\beta\| \le b$ with $b_H = \|M\|_{\text{op}} b$. The next two inequalities use the L-Lipschitz property of d_t and the identity $C_\beta^\star - Y = BD_\beta$, respectively. The final equality uses $\delta_\beta = \beta L\|M\|_{\text{op}}\|B\|_{\text{op}}$.

Since $\delta_\beta < 1$, the term involving D_β on the right can be moved to the left, bounding D_β by a square-integrable random variable. Taking the L^2 norm and using $\|d_t(Y)\|_{L^2} = a_t$ gives

$$\|D_\beta\|_{L^2} \le \frac{\beta\|M\|_{\text{op}} a_t + b_H}{1 - \delta_\beta}. \tag{A.22}$$

This proves square integrability of $A^\star_\beta$ and completes the justification of Eq. (A.18). The same Lipschitz estimate bounds the remainder as

$$\begin{aligned}\|\xi_\beta\|_{L^2} &\le \delta_\beta \|D_\beta\|_{L^2} + b_H \\ &\le \frac{\beta\|M\|_{\mathrm{op}}\delta_\beta}{1-\delta_\beta} a_t + \frac{b_H}{1-\delta_\beta}.\end{aligned} \tag{A.23}$$

We now use the decomposition $A^\star_\beta - \bar{A}(Y) = (I - \beta M)(Y - \bar{A}(Y)) + \xi_\beta$ to turn the remainder bound into a bound on the corrected estimate's distance to the posterior mean. Expanding the steps gives

$$\begin{aligned}\|A^\star_\beta - \bar{A}(Y)\|_{L^2} &= \big\|(I-\beta M)(Y-\bar{A}(Y)) + \xi_\beta\big\|_{L^2} \\ &\le \big\|(I-\beta M)(Y-\bar{A}(Y))\big\|_{L^2} + \|\xi_\beta\|_{L^2} \\ &\le \|I-\beta M\|_{\mathrm{op}}\|Y-\bar{A}(Y)\|_{L^2} + \|\xi_\beta\|_{L^2} \\ &= \|I-\beta M\|_{\mathrm{op}} a_t + \|\xi_\beta\|_{L^2} \\ &\le \|I-\beta M\|_{\mathrm{op}} a_t + \frac{\beta\|M\|_{\mathrm{op}}\delta_\beta}{1-\delta_\beta} a_t + \frac{b_H}{1-\delta_\beta} \\ &= \left(\|I-\beta M\|_{\mathrm{op}} + \frac{\beta\|M\|_{\mathrm{op}}\delta_\beta}{1-\delta_\beta}\right) a_t + \frac{b_H}{1-\delta_\beta} \\ &= c_\beta a_t + \frac{b_H}{1-\delta_\beta}.\end{aligned} \tag{A.24}$$

We now substitute the two bounds into the risk difference stated at the start. The gain condition in Eq. (A.6) ensures $a_t - b_H > c_\beta a_t + b_H/(1-\delta_\beta) \ge 0$, so the squared lower bound strictly exceeds the squared upper bound. Consequently,

$$\begin{aligned}&\mathbb{E}\|A^\star_{\mathrm{off}} - A_0\|^2 - \mathbb{E}\|A^\star_\beta - A_0\|^2 \\ &\quad= \mathbb{E}\|A^\star_{\mathrm{off}} - \bar{A}(Y)\|^2 - \mathbb{E}\|A^\star_\beta - \bar{A}(Y)\|^2 \\ &\quad\ge (a_t - b_H)^2 - \left(c_\beta a_t + \frac{b_H}{1-\delta_\beta}\right)^2 > 0.\end{aligned} \tag{A.25}$$

This proves Eq. (8): under the stated gain condition, the corrected limit is closer to the posterior mean in mean square and therefore has strictly lower expected reconstruction error. Together with Part 2, this establishes convergence and the claimed error improvement over the same solver with the score correction disabled. □

B Implementation Details

B.1 Task formulations and analytical ADMM

All three tasks follow the ADMM formulation in Eq. (2), with the common denoiser input $C^k = A^{k+1} + U^k$ and task-specific observation models and analytical prior updates. We detail these components and their implementation below. For matched comparisons, we disable the learned corrections while keeping the data update, analytical prior, and postprocessing fixed.

Reference-guided HSI restoration. We represent the hyperspectral image as $A = EB + \mu$, where B contains the spectral coefficients, and the spectral basis E and mean μ are estimated from the observations and kept fixed. Let $\mathcal{H}$ denote spatial blur and downsampling, and let $\mathcal{R}_{\mathrm{PAN}}$ denote the PAN spectral response. The data-fidelity term combines consistency with the low-resolution HSI and the PAN image:

$$f(A) = \frac{1}{2}\|\mathcal{H}(A) - Y_{\mathrm{LR}}\|^2_{Q(W)} + \frac{1}{2}\|\mathcal{R}_{\mathrm{PAN}}(A) - Y_{\mathrm{PAN}}\|_2^2, \tag{B.1}$$

where $\|r\|^2_{Q(W)} = \langle r, Q(W)r\rangle$, and $Q(W)$ incorporates observation-derived reliability weights. We implement the analytical ADMM solver in coefficient coordinates, with auxiliary variable V_B and scaled dual variable U_B:

$$\begin{aligned} B^{k+1} &= \arg\min_B f(EB+\mu) + \frac{\rho}{2}\|B - V_B^k + U_B^k\|_F^2, \\ C_B^k &= B^{k+1} + U_B^k, \\ V_B^{k+1} &= \mathcal{T}_\tau(C_B^k), \\ U_B^{k+1} &= U_B^k + B^{k+1} - V_B^{k+1}. \end{aligned} \tag{B.2}$$

The data update is a quadratic subproblem solved by preconditioned conjugate gradients. The analytical map $\mathcal{T}_\tau$ applies local singular-value soft thresholding to promote low-rank structure in the coefficient patches. Our learned correction modifies C_B^k before this analytical update, following the prior interface defined in the main text.

Single-image HSI denoising. We represent the hyperspectral image as $A = EB$, where E is a fixed orthonormal spectral basis estimated from the noisy observations and B contains the spectral coefficients. For mixed noise, we additionally introduce a sparse component S and a column component L, with the latter constant along the image height. The weighted data-fidelity term is

$$f(A; S, L) = \frac{1}{2}\|W^{1/2} \odot (Y - A - S - L)\|_F^2, \tag{B.3}$$

where Y is the noisy HSI and W contains observation-derived reliability weights. Gaussian denoising sets $S = L = 0$. We initialize $V_B^0 = E^\top Y$, $U_B^0 = 0$, and $S^0 = L^0 = 0$. Holding the noise estimates fixed during the coefficient update, the analytical ADMM steps are

$$\begin{aligned} B^{k+1} &= \arg\min_B f(EB; S^k, L^k) + \frac{\rho}{2}\|B - V_B^k + U_B^k\|_F^2, \\ C_B^k &= B^{k+1} + U_B^k, \\ V_B^{k+1} &= \mathcal{T}_{\mathrm{HSI}}(C_B^k), \\ U_B^{k+1} &= U_B^k + B^{k+1} - V_B^{k+1}. \end{aligned} \tag{B.4}$$

The coefficient data update is obtained by solving a positive-definite linear system at each pixel. In the LocalSVT/TV configuration, $\mathcal{T}_{\mathrm{HSI}}$ applies local singular-value soft thresholding followed by a finite-step second-order total variation (TV) update. DCT/TV configurations instead use a joint transform-sparsity and TV update. For mixed noise, each ADMM iteration is followed by robust updates using the residual $T^{k+1} = Y - EV_B^{k+1}$: the column component is estimated by soft-thresholding the height-wise medians of $T^{k+1} - S^k$ on the detected column support, with column centering where configured, and the sparse component is updated as $S^{k+1} = \mathrm{soft}(T^{k+1} - L^{k+1}, \tau_{\mathrm{imp}})$, where soft denotes element-wise soft thresholding. These noise estimates enter the next coefficient data update. The restored image is $\widehat{A} = EV_B^K$. Our learned correction modifies C_B^k before the analytical prior update, following the interface defined in the main text.

MRI denoising. We restore the magnitude image A directly from the noisy observation Y. The scaled Rician data-fidelity term is

$$f_{\mathrm{R}}(A) = \sum_p \left[\frac{1}{2} A_p^2 - \widehat{\sigma}_n^2 \log I_0\left(\frac{Y_p A_p}{\widehat{\sigma}_n^2} \right) \right], \qquad A \geq 0, \tag{B.5}$$

where p indexes pixels, $\widehat{\sigma}_n$ is an observation-derived noise estimate, and I_0 is the modified Bessel function of the first kind of order zero. Terms independent of A are omitted. For multicoil M4Raw magnitude images, this data term serves as a likelihood surrogate.

We initialize $V^0 = Y$ and $U^0 = 0$. With the nonnegative data-step center $b^k = \max(V^k - U^k, 0)$ used in our implementation, the analytical ADMM updates are

$$\begin{aligned} A^{k+1} &= \arg\min_{A \geq 0} f_{\mathrm{R}}(A) + \frac{\rho}{2} \|A - b^k\|_F^2, \\ C^k &= A^{k+1} + U^k, \\ V^{k+1} &= \mathcal{T}_{\mathrm{MRI}}(C^k), \\ U^{k+1} &= U^k + A^{k+1} - V^{k+1}. \end{aligned} \tag{B.6}$$

The data subproblem is approximately solved by fixed-point/MM iterations. The analytical map $\mathcal{T}_{\mathrm{MRI}}$ combines sparsity of the non-DC coefficients of the two-dimensional DCT with second-order total variation (TV), promoting transform-domain sparsity and spatial regularity. This joint prior update is approximately computed by primal–dual iterations. Our learned correction modifies C^k before the analytical prior update, following the interface defined in the main text.

B.2 Network architecture and state-bank construction

Correction network. The correction network is a shallow conditional U-Net with feature widths 48 and 96, four residual blocks, and one downsampling–upsampling pair. Each residual block uses GroupNorm and SiLU activations and is conditioned on a 128-dimensional embedding of the perturbation level. The input concatenates the normalized state, a one-channel guide, and three scalar condition maps. The final convolution is zero-initialized. The state has $r = 6$, 12, and 1 channels for reference-guided HSI restoration, single-image HSI denoising, and MRI denoising, respectively. HiPandas uses the observed PAN image as guidance; HSI denoising and MRI use guidance derived from their noisy observations. Each network contains approximately 0.64–0.65 million trainable parameters, with the architecture and exact counts given in Table 4. Dual-track inference uses two networks with the same architecture and independent weights, doubling the parameter count.

Table 4: Architecture of one correction network. r is the number of state channels, and $H \times W$ is the input spatial size. Parameter counts include all trainable weights and biases.

Stage	Operation	Output size	Parameters
Input	State, guide, and condition maps	$(r+4) \times H \times W$	–
Conditioning	MLP: $129 \to 128 \to 128$	128	33,152
Input projection	3×3 convolution	$48 \times H \times W$	$432r + 1{,}776$
Encoder block 1	Residual block	$48 \times H \times W$	47,952
Downsampling	4×4 convolution, stride 2	$96 \times \frac{H}{2} \times \frac{W}{2}$	73,824
Encoder block 2	Residual block	$96 \times \frac{H}{2} \times \frac{W}{2}$	178,848
Bottleneck	Residual block	$96 \times \frac{H}{2} \times \frac{W}{2}$	178,848
Upsampling	4×4 transposed convolution, stride 2	$48 \times H \times W$	73,776
Skip connection	Add encoder block 1 output	$48 \times H \times W$	–
Decoder block	Residual block	$48 \times H \times W$	47,952
Output projection	GroupNorm, SiLU, 3×3 convolution	$r \times H \times W$	$433r + 96$
Total: reference-guided HSI restoration ($r = 6$)			**0.641M**
Total: single-image HSI denoising ($r = 12$)			**0.647M**
Total: MRI denoising ($r = 1$)			**0.637M**

Training states. We first run the analytical ADMM solver and collect its denoiser inputs into a state bank $\mathcal{B}_0$. For single-track learning, the correction network is trained directly on $\mathcal{B}_0$. For dual-track

learning, we first train the structure network on $\mathcal{B}_0$ and freeze its weights. We then run ADMM with the structure correction enabled to construct a second bank $\mathcal{B}_1$, on which the complementary network is trained. This sequential procedure allows the complementary network to adapt to states generated after the structure correction. Each network is trained on a fixed bank without backpropagating through the ADMM iterations, and all network weights are fixed during final inference.

Fixed normalization. For each bank and channel, we compute a median center m_i and scale

$$s_i = \max\left\{ \frac{q_{0.99} - q_{0.01}}{2}, 10^{-4} \right\}, \qquad \mathcal{S}_i(C) = (C - m_i) \oslash s_i. \tag{B.7}$$

The statistics are fixed before training and reused at inference. In dual-track learning, the two banks may have different normalization statistics; branch outputs are therefore restored to the original coefficient scale before projection.

B.3 Self-supervised objectives and optimization

Score regression. For a normalized bank state Z, we sample $\epsilon \sim \mathcal{N}(0, I)$ and construct

$$Z_t = Z + \sqrt{\Delta_t}\epsilon, \qquad \Delta_t = \nu^2(\sigma_t^2 - 1), \qquad \kappa_t = \nu^2\sigma_t^2, \tag{B.8}$$

where $\nu > 0$ controls the artificial perturbation scale and $\sigma_t > 1$ specifies its level. We train the predicted correction $D_i(Z_t, t) = \kappa_t s_{\theta_i}(Z_t, t)$ to match the observable target $\kappa_t(Z - Z_t)/\Delta_t$. Before output bounding and branch projection, the regression objective is

$$\mathcal{L}_{\mathrm{score},i} = \mathbb{E}\Big[w_i(t) \left\| D_i(Z_t, t) - \kappa_t \Delta_t^{-1}(Z - Z_t) \right\|_F^2 \Big], \qquad w_i(t) = (N_i \gamma_i \sigma_t^2)^{-1}. \tag{B.9}$$

Here N_i is the number of tensor entries per training sample, and $\gamma_i = 0.01 a_i^{-1}$ incorporates the external score-loss coefficient a_i. When output bounding or branch projection is used, the same processing is applied to both predictions and targets before computing the squared error.

Auxiliary losses. When a held-out observation loss is used, a fixed set Ω is masked when constructing the state bank and guidance. The original noisy measurements at these locations, denoted by Y_Ω, serve as training targets. Let $\widehat{Y}_{\Omega,i} = [\mathcal{H}(\widehat{A}_i)]_\Omega$ denote the corresponding predicted measurements. For reference-guided HSI restoration, we minimize their weighted absolute difference:

$$\mathcal{L}_{\mathrm{held},i} = |\Omega|^{-1}\mathbb{E}\Big[\Big\| w_\Omega \odot (\widehat{Y}_{\Omega,i} - Y_\Omega) \Big\|_1 \Big], \tag{B.10}$$

where w_Ω contains the observation-reliability weights. For MRI, we instead average the scaled Rician discrepancy in Eq. (B.5) over the same held-out locations. The auxiliary objective is

$$\mathcal{L}_{\mathrm{aux},i} = \mathcal{L}_{\mathrm{held},i} + \omega_{\mathrm{p},i}\mathcal{L}_{\mathrm{PAN},i} \tag{B.11}$$

The PAN term is $\mathcal{L}_{\mathrm{PAN},i} = \mathbb{E}\,\mathrm{mean}\,|\mathcal{R}_{\mathrm{PAN}}(\widehat{A}_i) - Y_{\mathrm{PAN}}|$ and is used only for reference-guided HSI restoration.

Total objective. Consistent with the main text, each network is trained using

$$\mathcal{L}_i = \mathcal{L}_{\mathrm{score},i} + \lambda\mathcal{L}_{\mathrm{aux},i}. \tag{B.12}$$

The weights depend on the task and training configuration.

Optimization. HiPandas and MRI use AdamW with learning rate 10^{-4}, zero weight decay, batch size four, gradient-norm clipping at one, and an EMA decay of 0.999. Inference uses frozen EMA weights. MRI trains for 5,000 updates per network.

B.4 Projectors and inference settings

We construct two complementary orthogonal projectors to specify the correction subspace of each branch. The structure projector P_s selects directions associated with observation-supported column interference, while P_g selects their orthogonal complement. Under vectorization of the coefficient maps, let Q contain an orthonormal basis of the structural subspace. The projectors are defined as

$$P_s = QQ^\top, \qquad P_g = I - P_s, \qquad Q^\top Q = I. \tag{B.13}$$

They satisfy $P_s + P_g = I$ and $P_s P_g = 0$, so their ranges are complementary and mutually orthogonal.

For HSI column artifacts, the structural subspace combines spectral support with low-frequency variation along the image height. For each spatial column j, let M_j be a diagonal mask indicating the spectral bands with detected interference support. The eigenvectors associated with the numerically nonzero eigenvalues of $E^\top M_j E$ form an orthonormal basis Q_j in the spectral coefficient coordinates. We combine these directions with the first q orthonormal height-wise DCT modes to construct the structural subspace independently for each column. This construction captures supported patterns that are constant or slowly varying along the column.

Both networks receive the complete ADMM denoiser input. Their corrections are rescaled to the original coefficient coordinates and projected by P_s and P_g, respectively, before entering the analytical prior update. The structural subspace may also contain image signal, so the network learns the correction within the selected directions. When reliable structural support is unavailable, we use $P_s = 0$ and $P_g = I$.

B.5 Datasets and Implementation Details

Reference-guided HSI restoration. We use the Dongying dataset[1] from HiPandas (Xu et al., 2025). It contains 81 samples with 32 spectral bands. Each sample consists of a noisy $64 \times 64 \times 32$ low-resolution HSI and a 256×256 PAN image, with a target HSI of size $256 \times 256 \times 32$. The spatial scale factor is four. The Gaussian conditions include G10 and G30, as well as GNI with band-dependent, non-i.i.d. Gaussian noise. The mixed conditions, Mix35, Mix55, and Mix55-G30, combine Gaussian noise with stripe and bad-line interference at different severity levels. For each condition, all 81 noisy samples contribute states to a shared training bank, and the resulting network or network pair is shared across the samples. Training uses 64×64 crops in high-resolution state coordinates, corresponding to 16×16 low-resolution observation regions. Inference restores each complete sample. Clean high-resolution images are excluded from the training losses and are used for parameter selection and evaluation. Metrics are computed per sample and then averaged over all 81 samples.

Single-image HSI denoising. We use Qingpu[2] and Xiong'an[3], with one noisy HSI as the input to each restoration problem. The experimental scenes have spatial dimensions of 1580×3750, with 251 and 256 spectral bands for Qingpu and Xiong'an, respectively. Both datasets are evaluated under Gaussian conditions G30/G50/G70 and mixed conditions M30/M50/M70. The mixed conditions combine band-dependent Gaussian noise with stripe noise, dead lines, and salt-and-pepper noise. The scenes are divided into 256×256 spatial patches, which are restored independently using a network shared within each scene and noise condition. Training uses crops of solver states generated from the noisy observations.

MRI denoising. We use BrainWeb[4] and M4Raw[5] to evaluate denoising under simulated and acquired noise. BrainWeb provides T1, T2, and PD images at 1 mm spatial resolution, with a native axial image size of 217×181 pixels. We use the official 3%, 5%, and 7% noise levels, yielding nine conditions, and evaluate 34 slices per condition, giving 306 slice–condition pairs. For M4Raw, we evaluate all 25 test subjects, including the three used for parameter selection. Each subject provides 18 slices of size 256×256 for each of the T1, T2, and FLAIR contrasts, totaling 1,350 slices. We use the acquired noisy images without additional corruption. References are averages of repeated acquisitions that include the input repetition and therefore remain noisy and correlated with the input.

HiPandas parameters. Table 5 lists the HiPandas parameters. K is the number of outer ADMM iterations and ρ is the ADMM penalty. τ is the singular-value soft-threshold of the local low-rank prior. d is the componentwise correction cap in normalized state coordinates before gain scaling. β_s and β_g are the nominal gains of the structure and full-space/complementary corrections, respectively. N_s and N_g are the optimizer update counts of the corresponding networks; – indicates an unused network.

[1] https://github.com/shuangxu96/Hipandas

[2] https://www.3sobs.cn/qingpu-hsi-dataset

[3] https://doi.org/10.3974/geodb.2021.01.02.V1

[4] https://brainweb.bic.mni.mcgill.ca/selection_normal.html

[5] https://github.com/mylyu/M4Raw

Gaussian and GNI conditions use Full-space inference; mixed conditions use Structure-guided inference.

Table 5: HiPandas solver and correction parameters.

Condition	K	ρ	τ	d	β_s	β_g	N_s	N_g
G10	4	1	0.105	0.10	0	1	–	5000
G30	8	8	0.105	0.10	0	1	–	5000
GNI	8	4	0.105	0.10	0	1	–	5000
Mix35	8	8	0.070	0.10	0.75	1.25	10000	5000
Mix55	8	4	0.070	0.10	0.75	1.25	5000	5000
Mix55-G30	8	4	0.070	0.10	0.50	0.50	5000	5000

HSI denoising parameters. Table 6 lists the parameters for Qingpu and Xiong'an, using the same notation as above. Gaussian conditions use Full-space inference; mixed conditions use Structure-guided inference.

Table 6: HSI denoising solver and correction parameters.

Condition	K	ρ	d	β_s	β_g	N_s	N_g
Qingpu G30	30	2	0.1	0	0.5	–	5000
Qingpu G50	30	2	0.1	0	0.75	–	6000
Qingpu G70	30	2	0.1	0	0.75	–	5000
Qingpu M30	30	2	0.1	0.125	0.5	2000	2000
Qingpu M50	30	2	0.1	0.375	0.75	3000	3000
Qingpu M70	30	2	0.1	0.1875	0.75	2000	2000
Xiong'an G30	30	2	0.1	0	0.5	–	4000
Xiong'an G50	30	2	0.1	0	0.5	–	4000
Xiong'an G70	30	2	0.1	0	0.75	–	6000
Xiong'an M30	30	2	0.1	0.125	0.5	2000	2000
Xiong'an M50	30	2	0.1	0.75	0.75	2500	2500
Xiong'an M70	30	2	0.1	0.1875	0.75	2000	2000

MRI parameters. Tables 7 and 8 list the parameters for BrainWeb and M4Raw, respectively. Both datasets use Full-space inference. K, ρ, and d have the same meanings as above; N_g is the number of optimizer updates for the full-space network. c and a are the noise-scaled prior coefficients for DCT sparsity and TGV regularization, respectively, and J is the number of inner iterations used to evaluate their joint proximal map. γ_σ multiplies the estimated noise standard deviation before it is used by the solver. β is the nominal correction gain, and η_k is its iteration-dependent multiplier, so the applied gain is $\beta\eta_k$ at outer iteration $k \in \{0, \dots, K-1\}$. In Table 8, an arrow denotes the training value followed by the inference value; a single value is shared by both stages. $\mathbf{1}_{\{k\geq 10\}}$ activates the correction during the last 20 of 30 iterations, and $[x]_0^1 = \min\{1, \max\{0, x\}\}$ defines the ramp schedule.

Table 7: BrainWeb solver and correction parameters.

Condition	K	ρ	c	a	J	γ_σ	d	β	N_g	η_k
T1, 3%	30	2	0.1875	0.20	20	1	0.10	0.25	5000	1
T1, 5%	30	2	0.1875	0.20	20	1	0.10	0.50	5000	1
T1, 7%	30	2	0.1875	0.20	20	1	0.10	0.50	5000	1
T2, 3%	30	2	0.1875	0.20	20	1	0.10	0.25	5000	1
T2, 5%	30	2	0.1875	0.20	20	1	0.10	0.50	5000	1
T2, 7%	30	2	0.1875	0.20	20	1	0.10	0.75	5000	1
PD, 3%	30	2	0.1875	0.20	20	1	0.10	0.25	5000	1
PD, 5%	30	2	0.1875	0.20	20	1	0.10	0.50	5000	1
PD, 7%	30	2	0.1875	0.20	20	1	0.10	0.75	5000	1

Table 8: M4Raw solver and correction parameters.

Contrast	K	ρ	c	a	J	γ_σ	d	β	N_g	η_k
T1	30	2	$0.03125 \rightarrow 0.015625$	$0.350 \rightarrow 0.175$	20	1.25	0.20	1.00	5000	$[(k-15)/15]_0^1$
T2	30	2	0.09375	0.175	20	1.25	0.10	0.50	5000	$\mathbf{1}_{\{k \geq 10\}}$
FLAIR	30	2	0.125	0.050	20	1.25	0.10	0.50	5000	1

C Additional Experimental Results

C.1 Controlled Experiments for the Theoretical Analysis

Experimental setup. We use 16×16, 32×32, and 64×64 Gaussian states with a known ideal score. Normalization is the identity. For $d = m^2$, the clean covariance Σ is diagonal with entries 0.2, 0.5, 1, and 2 repeated equally; independent Gaussian corruptions have total variance $\kappa_t = 0.5$. The data fidelity is $\frac{1}{2}\langle A - Y, H(A - Y)\rangle$, where H has eigenvalues 0.25, 0.5, 2, and 4, repeated equally in a fixed orthogonal basis, and $\rho = 1$. The analytical map is exact singular-value soft thresholding on the complete matrix, with threshold $\tau = \gamma a_t/\sqrt{m}$, where $a_t^2 = \kappa_t^2 \operatorname{tr}((\Sigma + \kappa_t I)^{-1})$. Thus, $\gamma = b/a_t$ measures relative analytical-prior strength. Corrected and correction-disabled solvers share observations, initialization, data fidelity, prior, and penalty.

Convergence and stability. At $\gamma = 0.15$, we run original ADMM on 16 observations from two different initializations and measure the root-mean-square distance between their prior-input trajectories, normalized by its initial value. Figures 6(a)–(c) compare five correction strengths across dimensions; larger tested corrections yield faster decay in this setting. For this Hessian, the proposition's sufficient contraction condition holds for $\beta < 7/15$. The tested larger corrections within $[0, 1]$ also converge, demonstrating the conservatism of that condition. Panel (d) extends the sweep beyond the proposition's range: $\beta = 4$ and 8 produce growing trajectory distances and primal residuals, whereas $\beta = 2$ continues to decrease during the recorded iterations. Failure of the sufficient condition alone does not imply divergence.

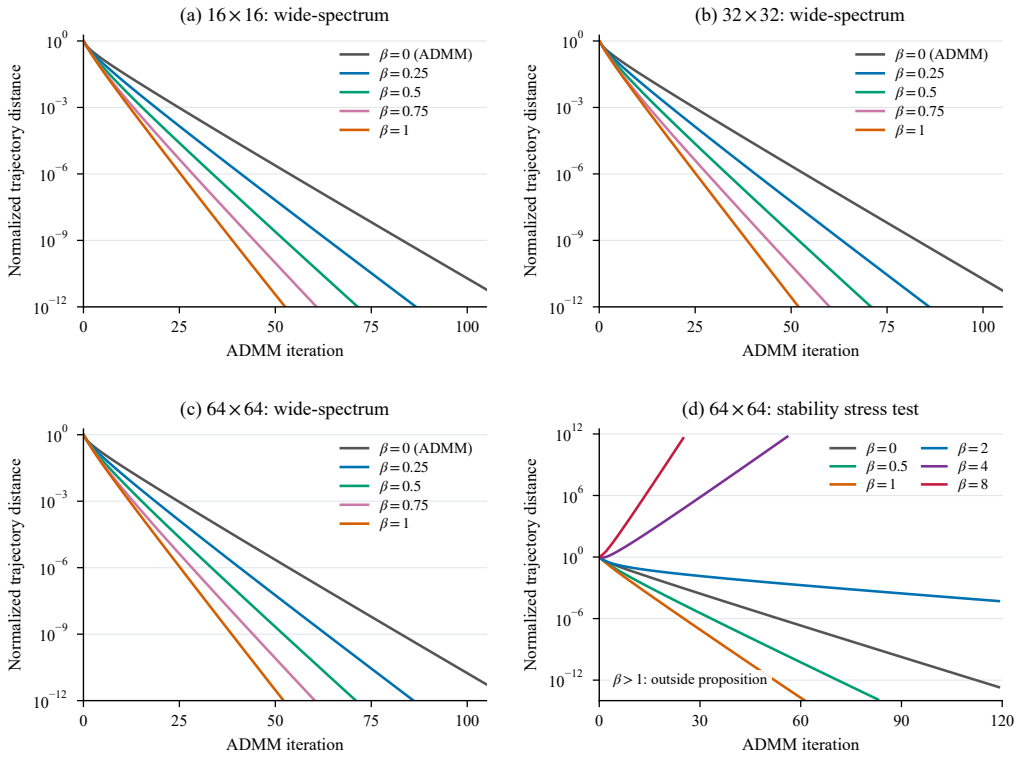

Figure 6: Wide-spectrum convergence and stability. (a)–(c) Original ADMM trajectories across dimensions at $\gamma = 0.15$. (d) Additional correction strengths at 64×64; $\beta > 1$ lies outside the proposition's range. Growing runs stop at a prescribed numerical safety threshold. Panel (d) uses wider axis ranges.

Reconstruction error. We evaluate 512 independent test matrices per dimension, reusing the same clean signals and noisy observations across parameter settings. Figure 7 plots per-element reconstruction error against γ for $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$. The 17 evaluated prior strengths span 0 to 1.5 in increments of 0.1, with 0.15 additionally included. The main text shows the 64×64 case,

using the same data and correction strengths. For each setting, we report

$$\widehat{R}_d(\beta,\gamma) = \frac{1}{Nd}\sum_{j=1}^{N} \|A^{\star}_{\beta,j} - A_{0,j}\|_F^2, \tag{C.1}$$

where $A^{\star}_{\beta,j}$ is computed at prior strength γ, and $\beta = 0$ gives analytical ADMM. Lower values indicate better reconstruction. The error curves illustrate the interaction between analytical regularization and score correction: stronger correction is not uniformly better. The tested nonzero correction strengths lie outside the sufficient risk-improvement region, so these comparisons characterize empirical performance beyond the certified settings.

Final risks are computed with Anderson acceleration of the ADMM state map, checking both original fixed-point and primal residuals below 10^{-10}. Selected estimates are also checked against direct ADMM. All convergence curves use original ADMM iterations.

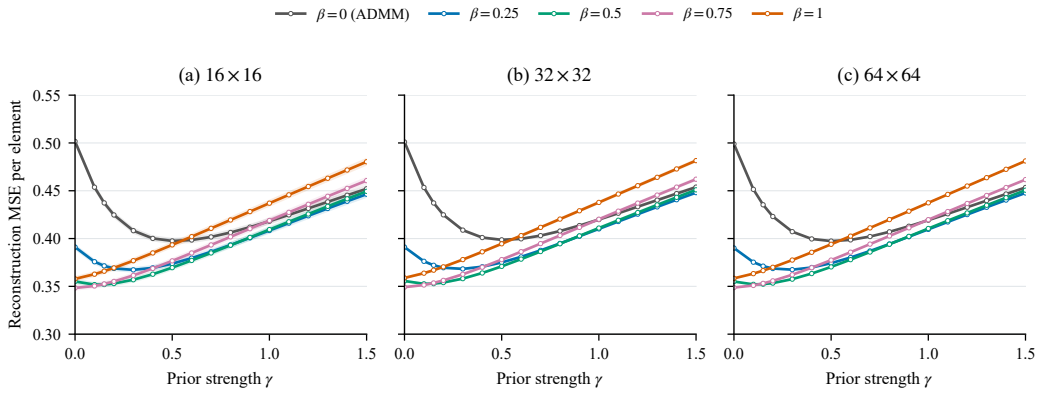

Figure 7: Reconstruction error under wide-spectrum fidelity. Each panel fixes the state dimension; lower curves indicate better restoration. The gray curve is analytical ADMM ($\beta = 0$). Markers denote evaluated prior strengths, connecting segments are visual guides, and shading shows pointwise 95% intervals computed across independent test matrices. All panels share the same vertical scale.

C.2 Additional quantitative results

Reference-guided HSI restoration. Tables 9 and 10 report G10/G30/GNI and mixed conditions, respectively, with 81 patches per condition. Bold and underlined entries indicate the best and second-best distinct values in each column.

Table 9: Reference-guided HSI restoration on Dongying under Gaussian/GNI noise.

Method	G10				G30				GNI			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
PLRDiff + HIR-Diff	34.52	0.8215	6.73	16.49	32.20	0.8115	8.96	20.36	31.01	0.7670	12.73	23.66
PLRDiff + PWRCTV	38.10	0.9445	5.42	12.22	33.43	0.9156	9.02	18.93	32.73	0.9082	12.05	21.31
SwinIR + HIR-Diff	33.88	0.8512	4.69	22.02	27.79	0.4978	10.21	33.18	27.18	0.4962	12.66	35.11
SwinIR + PWRCTV	31.66	0.7815	8.33	65.58	21.16	0.3662	29.96	422.81	22.62	0.4464	31.18	489.23
R-PNN + HIR-Diff	33.80	0.8068	6.22	17.49	29.10	0.6743	9.28	25.03	29.59	0.6866	10.90	25.12
R-PNN + PWRCTV	36.71	0.9240	5.01	13.53	29.58	0.6375	14.46	28.57	30.96	0.7174	15.31	29.07
HIR-Diff + PLRDiff	33.48	0.7606	8.77	42.15	27.52	0.5859	17.65	338.12	23.06	0.3942	31.89	467.35
HIR-Diff + SwinIR	36.23	0.8612	3.88	13.07	31.00	0.6963	7.59	21.15	30.03	0.6716	9.53	24.05
HIR-Diff + R-PNN	37.90	0.9390	4.12	11.52	34.82	0.9157	6.08	15.97	33.25	0.9048	8.04	18.57
PWRCTV + PLRDiff	38.52	0.9116	3.88	11.28	35.46	0.8691	5.97	30.24	34.41	0.8217	8.84	23.09
PWRCTV + SwinIR	37.50	0.9080	3.10	11.33	34.54	0.8638	5.12	15.96	34.60	0.8602	5.76	16.66
UHiPandas	<u>40.54</u>	<u>0.9543</u>	<u>3.04</u>	<u>8.52</u>	<u>36.82</u>	<u>0.9261</u>	<u>4.66</u>	<u>12.34</u>	<u>37.35</u>	<u>0.9300</u>	<u>4.46</u>	<u>11.74</u>
S^3-PnP (Ours)	**41.91**	**0.9614**	**2.58**	**6.89**	**38.52**	**0.9428**	**3.97**	**10.57**	**38.84**	**0.9448**	**3.80**	**10.13**

Single-image HSI denoising. Tables 11–14 report six conditions per dataset, each averaged over 85 patches. BM4D denotes the evaluated GPU approximation.

Table 10: Reference-guided HSI restoration on Dongying under mixed noise.

Method	Mix35				Mix55				Mix55-G30			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
PLRDiff + HIR-Diff	30.90	0.7649	12.30	26.04	30.70	0.7687	13.35	35.69	31.63	0.7997	11.03	34.60
PLRDiff + PWRCTV	32.58	0.9079	11.79	23.82	32.13	0.9000	12.92	34.07	32.82	0.9059	10.94	33.24
SwinIR + HIR-Diff	27.88	0.5443	12.53	34.58	27.62	0.5328	13.47	43.14	28.20	0.5456	11.84	42.82
SwinIR + PWRCTV	25.43	0.5544	27.50	393.81	23.98	0.4886	29.39	662.34	22.44	0.4233	30.17	685.00
R-PNN + HIR-Diff	30.52	0.7378	10.72	25.12	29.48	0.7004	12.24	34.86	29.48	0.7036	11.36	34.20
R-PNN + PWRCTV	32.45	0.7877	13.26	26.32	30.43	0.7176	16.23	73.68	29.56	0.6762	15.72	67.29
HIR-Diff + PLRDiff	24.58	0.4578	28.15	196.77	23.94	0.4272	28.41	80.01	25.74	0.5280	22.85	338.78
HIR-Diff + SwinIR	30.13	0.6949	10.51	26.73	29.64	0.6803	11.85	37.69	29.51	0.6577	11.62	38.12
HIR-Diff + R-PNN	33.04	0.9030	9.29	21.81	32.67	0.8946	10.52	33.86	33.64	0.9007	9.44	33.35
PWRCTV + PLRDiff	35.00	0.8303	9.02	24.35	32.02	0.7559	15.44	234.17	31.27	0.7379	16.33	202.54
PWRCTV + SwinIR	35.65	0.8813	5.98	17.46	33.14	0.8249	11.64	219.03	32.60	0.8146	12.07	244.81
UHiPandas	38.65	0.9440	3.92	10.78	36.49	0.9223	8.67	43.82	36.37	0.9218	8.70	44.13
S^3-PnP (Ours)	**39.24**	**0.9493**	**3.67**	**9.48**	**38.98**	**0.9469**	**3.81**	**9.86**	**36.91**	**0.9352**	**6.13**	**14.37**

Table 11: Single-image HSI denoising on Qingpu under Gaussian noise.

Method	G30				G50				G70			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
BM4D	36.44	0.8936	2.60	6.72	33.64	0.8194	3.65	9.29	31.77	0.7511	4.58	11.54
HLRTF	40.70	0.9427	1.28	3.66	34.39	0.7927	2.81	8.10	29.74	0.5845	5.11	15.13
RCTV	33.03	0.7311	4.10	10.87	27.95	0.4866	7.36	19.66	24.79	0.3364	10.55	27.95
DDS2M	41.57	0.9645	1.12	3.49	39.87	0.9507	1.40	4.21	38.47	0.9361	1.68	4.95
LRTFR	38.73	0.9235	1.39	5.11	37.11	0.9000	1.78	6.22	35.89	0.8826	2.17	7.17
HIR-Diff	39.31	0.9260	1.49	4.42	38.84	0.9252	1.60	4.61	37.25	0.8958	1.95	5.51
CF-INR	42.29	0.9686	1.14	3.25	39.94	0.9535	1.37	**4.05**	38.73	0.9403	**1.56**	**4.72**
S^3-PnP (Ours)	**42.60**	**0.9714**	**1.03**	**3.02**	**40.05**	**0.9558**	**1.35**	4.09	**38.77**	**0.9413**	1.60	4.73

Table 12: Single-image HSI denoising on Qingpu under mixed noise.

Method	M30				M50				M70			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
BM4D	22.54	0.3971	23.26	91.66	25.36	0.5140	15.60	61.07	25.77	0.5388	13.63	53.29
HLRTF	35.60	0.9123	27.83	115.44	30.84	0.8113	27.54	117.36	27.25	0.6595	27.70	121.07
RCTV	36.83	0.9183	3.88	11.74	33.64	0.8730	5.51	16.32	31.55	0.8311	7.07	20.32
DDS2M	29.88	0.8430	13.20	53.63	29.78	0.8552	13.12	53.49	29.70	0.8625	13.05	53.41
LRTFR	32.70	0.8733	27.13	112.10	30.62	0.8417	26.95	114.20	29.20	0.8176	26.94	117.34
HIR-Diff	29.17	0.8363	13.20	55.69	29.09	0.8295	13.21	55.69	28.91	0.8123	13.19	55.40
CF-INR	38.71	0.9596	1.82	5.95	35.53	0.9335	2.62	8.56	33.35	0.9086	3.42	10.99
S^3-PnP (Ours)	**39.62**	**0.9648**	**1.56**	**4.35**	**37.51**	**0.9389**	**1.87**	**5.46**	**36.50**	**0.9261**	**2.24**	**6.21**

Table 13: Single-image HSI denoising on Xiong'an under Gaussian noise.

Method	G30				G50				G70			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
BM4D	35.02	0.8637	2.54	5.58	32.40	0.7786	3.49	7.57	30.66	0.7046	4.32	9.28
HLRTF	39.91	0.9502	1.25	3.29	34.77	0.8508	2.25	5.86	30.51	0.6857	3.84	9.86
RCTV	33.03	0.7797	3.48	7.65	27.97	0.5534	6.25	13.88	24.79	0.3981	9.01	20.04
DDS2M	39.75	0.9448	1.22	3.35	38.07	0.9250	1.47	4.02	36.61	0.9001	1.73	4.73
LRTFR	35.57	0.8309	1.61	5.70	34.18	0.7808	1.98	6.77	33.26	0.7510	2.31	7.54
HIR-Diff	36.40	0.8830	1.58	4.90	36.18	0.8840	1.73	4.98	35.22	0.8652	2.01	5.55
CF-INR	39.46	0.9446	1.47	3.50	37.67	0.9153	1.68	4.30	36.52	0.8903	1.85	4.94
S^3-PnP (Ours)	**40.74**	**0.9632**	**1.13**	**2.88**	**38.66**	**0.9418**	**1.42**	**3.65**	**36.92**	**0.9161**	**1.69**	**4.49**

Table 14: Single-image HSI denoising on Xiong'an under mixed noise.

Method	M30				M50				M70			
	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓	PSNR ↑	SSIM ↑	SAM ↓	ERGAS ↓
BM4D	22.38	0.3987	18.29	43.57	24.96	0.5014	12.33	28.99	25.21	0.5095	10.85	25.93
HLRTF	33.61	0.8941	24.44	81.55	30.12	0.8326	24.06	81.94	27.70	0.7451	24.21	83.75
RCTV	34.86	0.8863	3.77	9.16	31.92	0.8267	5.28	12.47	30.00	0.7754	6.75	15.83
DDS2M	29.05	0.8231	10.05	26.26	28.93	0.8250	9.92	26.17	28.81	0.8212	9.81	26.11
LRTFR	29.98	0.7704	23.74	79.14	28.30	0.7195	23.51	80.07	27.23	0.6901	23.48	81.42
HIR-Diff	28.47	0.8351	9.83	27.49	28.41	0.8277	9.82	27.41	28.29	0.8129	9.80	27.21
CF-INR	36.43	0.9285	2.14	5.14	33.65	0.8762	2.95	7.30	31.82	0.8362	3.76	9.39
S^3-PnP (Ours)	**38.00**	**0.9476**	**1.62**	**4.07**	**37.46**	**0.9306**	**1.82**	**4.40**	**35.21**	**0.9035**	**2.21**	**5.57**

MRI denoising. Table 15 reports nine BrainWeb contrast–noise conditions, each averaged over 34 slices. Table 16 reports three M4Raw contrasts, with equal weighting of 25 subjects after averaging 18 slices per subject. C2S on M4Raw uses the official released weights evaluated on our benchmark; BrainWeb retains the local evaluation. DRDD denotes transfer of the official DRDD-AiO5 natural-image pretrained weights.

C.3 Additional qualitative results

The selected examples use the same display mapping, zoom region, and residual scale across methods within each row. The left inset shows the zoomed region; the right shows its absolute residual, with brighter colors indicating larger errors. HSI residuals use displayed RGB channels, BrainWeb uses displayed intensities, and M4Raw uses full-precision intensities.

Reference-guided HSI restoration. Figure 8 shows Gaussian and mixed-noise examples; PAN is guidance and has no HSI residual.

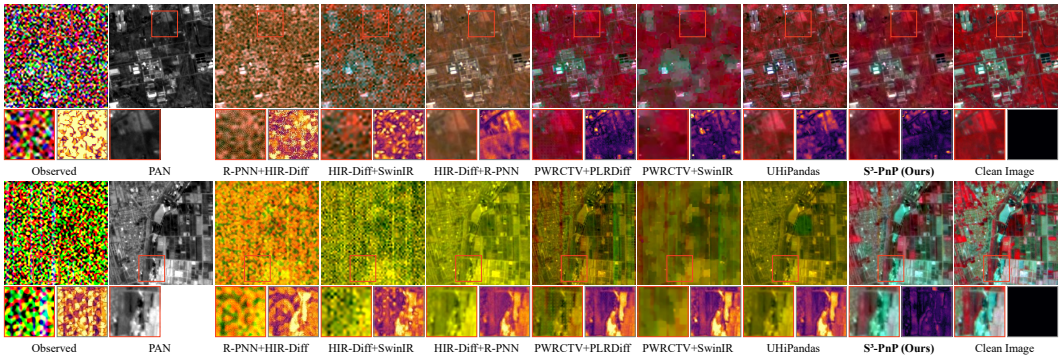

Figure 8: Reference-guided HSI restoration on Dongying: G30 (top) and Mix55 (bottom).

Single-image HSI denoising. Figures 9 and 10 show examples from both datasets.

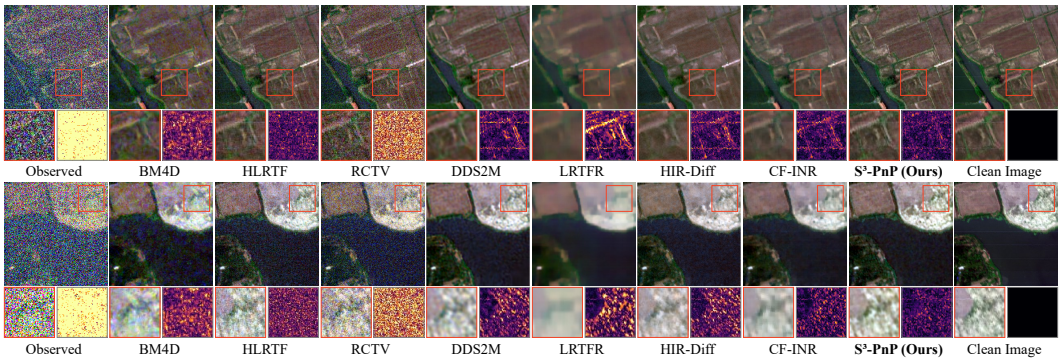

Figure 9: Single-image HSI denoising on Qingpu: G30 (top) and G50 (bottom).

Table 15: MRI denoising on BrainWeb across nine contrast–noise conditions.

T1

Method	T1 3%		T1 5%		T1 7%	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Noise2Void	35.41	0.9062	34.88	0.9425	32.97	0.8833
Noise2Self	33.17	0.7933	33.23	0.9600	30.81	0.7435
Noisier2Noise	37.49	0.9261	34.52	0.9049	31.90	0.7828
R2R	36.20	0.8040	34.83	0.8113	32.26	0.7534
Noise2Score	37.48	0.8741	32.83	0.7883	29.68	0.7370
ZS-N2N	35.38	0.7805	31.36	0.7240	28.70	0.6903
C2S	36.17	0.9301	34.90	0.8402	35.05	0.8866
DRDD	38.23	0.8314	33.85	0.7614	30.89	0.7282
S^3-PnP (Ours)	**40.84**	**0.9824**	**37.97**	**0.9737**	**36.03**	**0.9636**

T2

Method	T2 3%		T2 5%		T2 7%	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Noise2Void	29.46	0.8822	27.60	0.7965	26.03	0.7304
Noise2Self	30.26	0.8962	28.83	0.7829	28.72	0.8620
Noisier2Noise	32.35	0.8728	29.29	0.8076	26.79	0.7151
R2R	33.39	0.8196	31.10	0.8111	28.00	0.7279
Noise2Score	32.79	0.8268	28.42	0.7586	25.30	0.7098
ZS-N2N	30.89	0.7812	27.15	0.7243	24.45	0.6807
C2S	34.39	0.9386	31.85	0.8463	29.44	0.7749
DRDD	32.93	0.8093	28.83	0.7595	25.98	0.7270
S^3-PnP (Ours)	**35.36**	**0.9545**	**32.46**	**0.9339**	**30.54**	**0.9234**

PD

Method	PD 3%		PD 5%		PD 7%	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Noise2Void	31.46	0.8547	28.73	0.8176	28.71	0.8195
Noise2Self	30.23	0.8943	29.14	0.7758	28.93	0.7802
Noisier2Noise	33.32	0.8903	30.32	0.7897	28.00	0.7250
R2R	32.48	0.7798	30.15	0.7413	28.24	0.7063
Noise2Score	33.13	0.8207	28.59	0.7394	25.58	0.6932
ZS-N2N	31.62	0.7686	27.70	0.7011	25.05	0.6538
C2S	35.07	0.9174	33.14	0.8647	31.14	0.7904
DRDD	33.93	0.8157	29.42	0.7534	26.63	0.7148
S^3-PnP (Ours)	**36.21**	**0.9334**	**33.53**	**0.9165**	**31.59**	**0.8966**

Table 16: MRI denoising on M4Raw across three contrasts.

Method	T1		T2		FLAIR	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Noise2Void	31.24	0.8671	30.45	0.8392	30.80	0.8288
Noise2Self	29.10	0.8537	28.17	0.8095	28.94	0.8150
Noisier2Noise	31.18	0.8675	30.55	0.8435	31.17	0.8356
R2R	30.82	0.8552	30.48	0.8329	30.97	0.8187
Noise2Score	31.23	0.8444	30.48	0.8139	31.00	0.7975
ZS-N2N	31.66	0.8922	30.96	0.8729	31.89	0.8765
C2S	32.61	0.9152	32.30	0.8880	**33.02**	0.8792
DRDD	32.43	0.9108	31.44	0.8770	32.60	0.8797
S^3-PnP (Ours)	**32.69**	**0.9165**	**32.31**	**0.8922**	32.83	**0.8906**

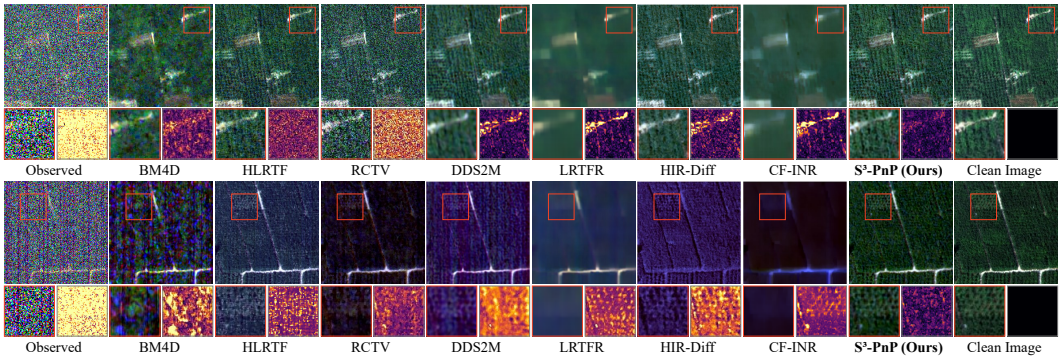

Figure 10: Single-image HSI denoising on Xiong'an: G70 (top) and M70 (bottom).

MRI denoising. Figures 11 and 12 show anatomical details and residuals. The M4Raw reference averages repeated acquisitions, including the input.

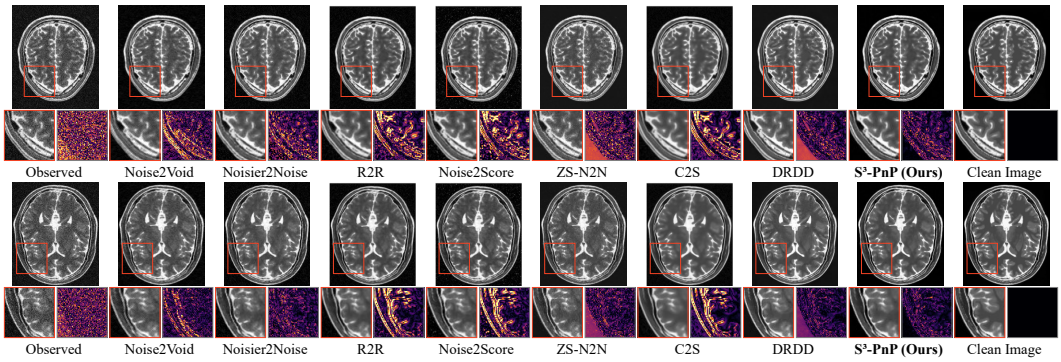

Figure 11: MRI denoising on BrainWeb (T2, 7% noise).

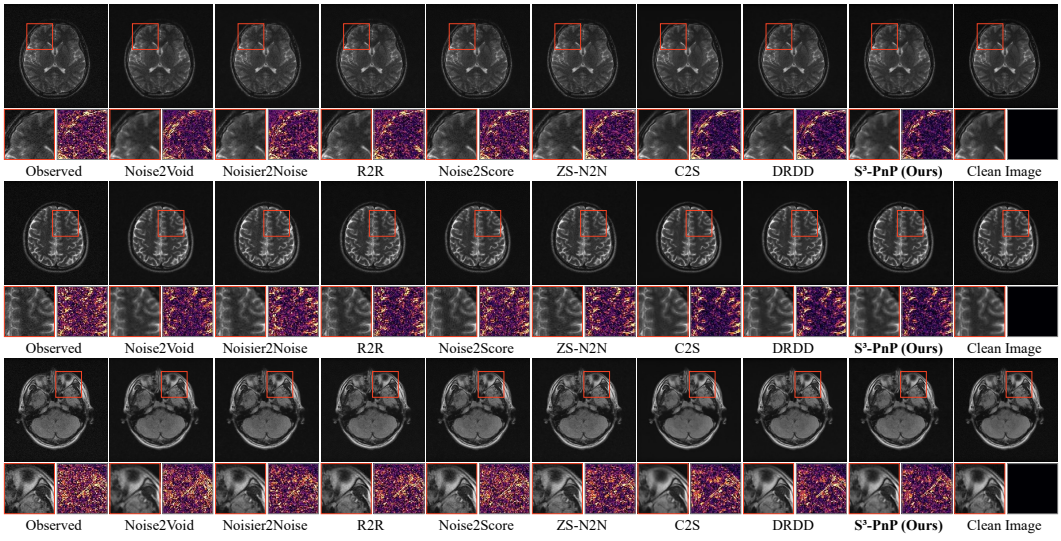

Figure 12: MRI denoising on M4Raw: T2, T2, and FLAIR from top to bottom.