TRAJECTORY CONSTRAINTS FOR IMAGING INVERSE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have achieved strong performance in imaging inverse problems, but typically rely on domain-matched pretrained models and external training data. Moreover, errors introduced at intermediate reverse-sampling steps can accumulate and degrade the final reconstruction, with their propagation strongly influenced by how successive states are related. We propose TRACE, a training-free **TRA**jectory-**C**onstrained r**E**construction framework for imaging inverse problems. TRACE replaces the pretrained diffusion model with an untrained overparameterized network optimized directly from the observed measurements and organizes reconstruction as a sequence of intermediate states. At each step, TRACE enforces stochastic proximal consistency (SPC) by inferring each reconstruction from a stochastically perturbed predecessor under measurement consistency, while proximally constraining the resulting estimate to its corresponding unperturbed state. This formulation admits a proximal interpretation, from which we derive stability bounds characterizing the variation between consecutive states under approximate network updates. A local Jacobian analysis further characterizes how SPC regularizes the network-parameterized reconstruction. Experiments on natural-image restoration and CT reconstruction demonstrate competitive performance against both untrained and pretrained neural-prior methods. We further show that the proposed trajectory constraint can be incorporated into existing diffusion-based inverse solvers to improve reconstruction performance.

1 INTRODUCTION

Many imaging tasks can be formulated as inverse problems, where the goal is to recover an unknown signal $\mathbf{x}_0$ from incomplete or indirect measurements $\mathbf{y}$ (Ravishankar et al., 2020; Lustig et al., 2007; Yang et al., 2008). Because inverse problems are often non-injective, multiple signals may agree with the same measurements, making prior information essential. In a Bayesian formulation, reconstruction is characterized by the posterior $p(\mathbf{x}_0 \mid \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{x}_0)p(\mathbf{x}_0)$, where $p(\mathbf{y} \mid \mathbf{x}_0)$ is the measurement model and $p(\mathbf{x}_0)$ is the signal prior.

Diffusion models provide a powerful learned representation of this prior (Ho et al., 2020; Song et al., 2021). In score-based diffusion, reconstruction proceeds through a sequence of progressively denoised states $\mathbf{x}_T, \ldots, \mathbf{x}_0$ generated by a discretized reverse-time process. For inverse problems, the measurements additionally guide these updates, producing a reconstruction trajectory whose intermediate states gradually evolve toward a measurement-consistent solution (Chung et al., 2023a; Zhang et al., 2025). Importantly, each reverse update is conditioned on the preceding state, thereby linking successive reconstructions along the trajectory. However, this approach has two limitations. First, diffusion-based inverse problem solvers typically rely on domain-matched pretrained models and external training data (Wang et al., 2023b; Kawar et al., 2022; Alkhouri et al., 2025). Second, errors introduced at intermediate reverse-sampling steps can accumulate across the reconstruction trajectory and degrade the final reconstruction (Alkhouri et al., 2025). Recent work further shows that the manner in which successive states are related can critically affect the evolution of this trajectory (Zhang et al., 2025). These drawbacks motivate a training-free reconstruction framework that adapts directly to the observed measurements without external training data, while explicitly regulating how successive reconstruction states evolve along the trajectory.

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.

This motivates our first question: *Can we retain the benefits of the diffusion-based reconstruction without relying on external training data?* Note that deep Image Prior (DIP) offers a promising starting point by adapting an untrained network directly to the observed measurements (Ulyanov et al., 2018). Moreover, DIP and diffusion sampling exhibit a related coarse-to-fine frequency progression, with lower-frequency structure emerging before finer details (Shi et al., 2021; Lee et al., 2025). This similarity makes the untrained neural prior a natural candidate to replace the pretrained diffusion model in a diffusion-inspired reconstruction process. However, this replacement does not resolve how the trajectory of intermediate states should evolve. In diffusion-based solvers, the pretrained reverse model not only provides a learned prior but also governs the evolution between successive states. This raises our second question: *How should the resulting intermediate reconstructions be constrained?*

To address these questions, we propose TRACE, a training-free **tra**jectory-**c**onstrained r**e**construction framework for imaging inverse problems. TRACE replaces the pretrained diffusion model with an instance-adaptive untrained neural network optimized directly from the observed measurements. At each step, TRACE perturbs the current estimate and updates the network using both measurement fidelity and a penalty on the change between consecutive states. The construction admits a proximal interpretation. After fixing one state, updating its predecessor corresponds to a proximal step involving measurement fidelity, implicit regularization, and stochastic proximal consistency to the neighboring state. Our contributions are as follows.

- We introduce TRACE, which retains the progressive stochastic reconstruction process of diffusion-based methods while using an instance-adaptive untrained network optimized directly from the measurements. This removes the dependence on external pretraining and enables explicit control of the reconstruction trajectory.
- We derive a proximal interpretation of TRACE and establish stability bounds showing how stochastic proximal consistency controls transitions between consecutive states under approximate neural updates. We further develop a local Jacobian analysis that characterizes the stochastic proximal consistency and relate this interaction to the spectral bias of the untrained network.
- We evaluate TRACE across a range of imaging inverse problems and demonstrate competitive reconstruction performance against leading baselines. We further show that trajectory consistency can be incorporated into existing diffusion-based solvers to improve reconstruction performance.

2 BACKGROUND

Problem setup. Consider the linear inverse problem $\mathbf{y} = \mathcal{A}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$ is the unknown signal, $\mathbf{y} \in \mathbb{R}^m$ denotes the observed measurements, and $\mathcal{A} \in \mathbb{R}^{m\times n}$ is a known forward operator. In many practical settings, $m < n$, making the problem underdetermined and allowing multiple signals to be consistent with the same measurements. Regularization is therefore commonly used to incorporate prior assumptions about the unknown signal, restrict the solution space, and promote stable reconstructions (Benning & Burger, 2018). A standard variational formulation is

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathcal{A}\mathbf{x}\|_2^2 + R(\mathbf{x}).$$

where the first term enforces consistency with the observed measurements and R is a regularizer encoding prior knowledge about $\mathbf{x}$. Existing reconstruction methods differ substantially in how such prior knowledge is represented and incorporated into the reconstruction. In the following, we review three lines of work particularly relevant to our setting, proximal and plug-and-play methods, which impose priors through proximal or denoising operators within iterative reconstruction algorithms; diffusion-based methods, which leverage pretrained generative priors for posterior sampling conditioned on the measurements; and DIP, which exploits the implicit prior induced by the structure and optimization of an overparameterized neural network fitted directly to the measurements.

Proximal methods iteratively update the optimization variable by minimizing an objective together with a quadratic penalty that discourages large deviations from the preceding iterate (Moreau, 1965;

Rockafellar, 1976). This gives a natural method for controlling the evolution of an iterative reconstruction. Plug-and-play (PnP) methods replace the proximal map of an explicit regularizer with an image denoiser, enabling powerful denoising models to be incorporated into iterative reconstruction (Venkatakrishnan et al., 2013; Zhang et al., 2021). Modern PnP methods often employ denoisers pretrained on external data, introducing dependence on the training distribution.

Diffusion models provide powerful learned priors through a stochastic reverse process. For inverse problems, measurement consistency is incorporated into this process to guide the reconstruction toward a solution that matches the observed data (Song et al., 2022; Chung et al., 2023a; Alkhouri et al., 2025; Zhang et al., 2025). However, they typically require pretrained diffusion models, and errors introduced at intermediate reverse steps may propagate through the reconstruction trajectory.

DIP optimizes an untrained network directly from the measurements (Ulyanov et al., 2018), with subsequent methods introducing sequential, self-guided, or pretrained diffusion-guided inputs and transferable weights (Alkhouri et al., 2024; Liang et al., 2025c;b;a), related work has also combined untrained neural priors with diffusion-inspired reconstruction (Luo et al., 2026). The network architecture and optimization dynamics act as an implicit image prior, removing the need for external training data.

TRACE is positioned at the intersection of these three directions. It uses an untrained neural prior for instance-adaptive reconstruction, follows a stochastic trajectory inspired by diffusion, and introduces an explicit stochastic proximal consistency between consecutive states that admits a proximal interpretation.

3 METHOD

Motivated by the reverse-trajectory structure of diffusion models, we represent reconstruction as a sequence of states $\{\mathbf{x}_t\}_{t=0}^{T}$, where $\mathbf{x}_T$ denotes the initial state and $\mathbf{x}_0$ the final reconstruction. We adopt the backward indexing convention of diffusion and retain a prescribed noise schedule along this trajectory, while removing the pretrained diffusion model. To regulate the evolution of the resulting trajectory, we penalize large changes between neighboring reconstruction states. Combining this inter-state constraint with the classical regularized inverse-problem formulation $f(\mathbf{x}) + \lambda R(\mathbf{x})$ gives

$$\min_{\{\mathbf{x}_t\}_{t=0}^{T}} \sum_{t=0}^{T} \left(f(\mathbf{x}_t) + \lambda_t R(\mathbf{x}_t)\right) + \sum_{t=0}^{T-1} \frac{\beta_t}{2} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2, \tag{1}$$

where $f(\mathbf{x}) = \frac{1}{2}\|\mathcal{A}\mathbf{x} - \mathbf{y}\|_2^2$ enforces measurement consistency, $R(\cdot)$ denotes a regularization functional, and $\beta_t > 0$ controls the stochastic proximal consistency between neighboring states.

We do not solve equation 1 as a full joint optimization problem over all states. Instead, TRACE uses a backward one-step approximation induced by this trajectory energy. Starting from $\mathbf{x}_T$, we construct the trajectory sequentially for $t = T-1, \ldots, 0$ by fixing the already available state $\mathbf{x}_{t+1}$ and updating $\mathbf{x}_t$ through

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \lambda_t R(\mathbf{x}) + \frac{\beta_t}{2} \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2. \tag{2}$$

Letting $F_t(\mathbf{x}) = f(\mathbf{x}) + \lambda_t R(\mathbf{x})$, equation 2 takes the form of a proximal update (Moreau, 1965; Rockafellar, 1976)

$$\mathbf{x}_t = P_t(\mathbf{x}_{t+1}) := \mathrm{prox}_{\frac{1}{\beta_t} F_t}(\mathbf{x}_{t+1}). \tag{3}$$

Since $R(\cdot)$ is not explicitly specified, the proximal map in equation 3 cannot be evaluated directly. Motivated by deep plug-and-play methods, where proximal updates are replaced by neural denoisers (Zhang et al., 2021; Wu et al., 2024a; Huang et al., 2025), we approximate this step using an untrained neural mapping. Unlike standard plug-and-play methods that employ a fixed pretrained denoiser, the network is optimized directly from the observed measurements at each trajectory step, providing an instance-specific implicit regularization.

We next incorporate the second retained ingredient from diffusion, scheduled stochastic perturbation. Before solving the local neural reconstruction problem at step t, we perturb the available state

Algorithm 1 TRACE: **TRA**jectory-**C**onstrained r**E**construction framework

Input: $\mathbf{y}$, $\mathbf{x}_T$, θ_T^K, trajectory steps T, inner network iterations K, forward operator $\mathcal{A}$, perturbation $\{\sigma_t\}_{t=0}^{T-1}$, parameters $\{\beta_t\}_{t=0}^{T-1}$.
for $t = T-1, \dots, 0$ **do**
  Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, I)$, $\mathbf{u}_t = \mathbf{x}_{t+1} + \sigma_t \boldsymbol{\epsilon}_t$.
  Initialize $\theta_t^0 = \theta_{t+1}^K$.
  for $k = 1, \dots, K$ **do**
    Update θ_t^k from θ_t^{k-1} by one optimizer step on $\mathcal{L}_{\theta_t}$ in equation 5.
  end for
  Set $\mathbf{x}_t = D_{\theta_t^K}(\mathbf{u}_t)$.
end for
return $\mathbf{x}_0$

as

$$\mathbf{u}_t = \mathbf{x}_{t+1} + \sigma_t \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

where σ_t is determined by the prescribed noise schedule. The next trajectory state is parameterized as $\mathbf{x}_t = D_{\theta_t}(\mathbf{u}_t)$.

We summarize the resulting TRACE construction through three design principles that govern each trajectory step:

- **P1** **Measurement Consistency.** The intermediate reconstruction is adapted to agree with the observed measurements.
- **P2** **Stochastic Proximal Consistency.** The resulting reconstruction is stochastically perturbed according to the prescribed noise schedule, while the updated reconstruction is constrained to remain close to its unperturbed state.

Together, these principles lead to the practical TRACE objective

$$\mathcal{L}_{\theta_t} = \frac{1}{2} \left\| \mathcal{A} D_{\theta_t}(\mathbf{u}_t) - \mathbf{y} \right\|_2^2 + \frac{\beta_t}{2} \left\| D_{\theta_t}(\mathbf{u}_t) - \mathbf{x}_{t+1} \right\|_2^2, \tag{5}$$

The trajectory is initialized by an uncoupled DIP reconstruction. Given a random input $\mathbf{z}$, the network weights θ_T^0 are randomly initialized as in DIP (Ulyanov et al., 2018) and optimized for K iterations:

$$\theta_T^K \approx \arg\min_{\theta} \frac{1}{2} \| \mathcal{A} D_\theta(\mathbf{z}) - \mathbf{y} \|_2^2, \qquad \mathbf{x}_T = D_{\theta_T^K}(\mathbf{z}). \tag{6}$$

For each subsequent trajectory step, the network is warm-started from the preceding optimized weights, $\theta_t^0 = \theta_{t+1}^K$, where $k = 0, \dots, K$ indexes the inner optimization iterations. The complete procedure is summarized in Algorithm 1.

4 Theoretical Analysis

4.1 Effect of Proximal Consistency

We analyze the local update in equation 2 and study how the proximal consistency term affects the change between two consecutive reconstruction states. Recall that $F_t(\mathbf{x}) = f(\mathbf{x}) + \lambda_t R(\mathbf{x})$, so that the local objective can be written as

$$\Phi_t(\mathbf{x}; \mathbf{x}_{t+1}) = F_t(\mathbf{x}) + \frac{\beta_t}{2} \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2. \tag{7}$$

In TRACE, the practical state $\mathbf{x}_t = D_{\theta_t^K}(\mathbf{u}_t)$ is obtained from a finite neural optimization and therefore need not exactly minimize equation 7. We measure the discrepancy between the ideal proximal model in equation 2 and the implemented neural update by the suboptimality gap

$$\varepsilon_t := \Phi_t(\mathbf{x}_t; \mathbf{x}_{t+1}) - \inf_{\mathbf{x}} \Phi_t(\mathbf{x}; \mathbf{x}_{t+1}). \tag{8}$$

Proposition 4.1. *Suppose that F_t is bounded below and that $F_t(\mathbf{x}_{t+1}) < \infty$. Let $\underline{F}_t := \inf_{\mathbf{x}} F_t(\mathbf{x}) > -\infty$. For any $\beta_t > 0$, the state $\mathbf{x}_t$ satisfies*

$$\frac{\beta_t}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2 \le F_t(\mathbf{x}_{t+1}) - F_t(\mathbf{x}_t) + \varepsilon_t. \tag{9}$$

Consequently,

$$\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2 \le \sqrt{\frac{2\left(F_t(\mathbf{x}_{t+1}) - \underline{F}_t + \varepsilon_t\right)}{\beta_t}}. \tag{10}$$

Proof. From the definition of ε_t in equation 8,

$$\Phi_t(\mathbf{x}_t; \mathbf{x}_{t+1}) = \inf_{\mathbf{x}} \Phi_t(\mathbf{x}; \mathbf{x}_{t+1}) + \varepsilon_t \le \Phi_t(\mathbf{x}_{t+1}; \mathbf{x}_{t+1}) + \varepsilon_t = F_t(\mathbf{x}_{t+1}) + \varepsilon_t,$$

where the second inequality follows by using $\mathbf{x}_{t+1}$ as a feasible point. Substituting equation 7 gives equation 9. Since $F_t(\mathbf{x}_t) \ge \underline{F}_t$, equation 10 follows directly. □

Proposition 4.1 provides direct control on the change between two consecutive reconstruction states, up to the local suboptimality ε_t. It does not require the realized trajectory difference to decrease monotonically across reconstruction steps, since the local objective, coupling strength, and approximation error may vary with t. In Appendix B, we empirically compare the coupled and uncoupled trajectories under the same stochastic perturbation schedule.

When $\varepsilon_t = 0$, the result reduces to the exact proximal update in equation 2. Stronger results for the ideal proximal trajectory and the propagation of approximation errors are given in Appendix A.1.

Remark 4.2. *The bound above requires $\beta_t > 0$. When $\beta_t = 0$, the local objective in equation 2 contains no term involving $\|\mathbf{x} - \mathbf{x}_{t+1}\|_2$, and the local objective alone does not control the change relative to $\mathbf{x}_{t+1}$.*

4.2 Local Analysis of Stochastic Proximal Consistency

While Proposition 4.1 characterizes how proximal consistency controls the change between consecutive reconstruction states, SPC applies this constraint when the network is driven by a stochastically perturbed trajectory state. We next analyze the local regularization induced by this stochastic proximal formulation.

Proposition 4.3. *Fix a trajectory state x and network parameters θ where $J_D(x) = \left.\frac{\partial D_\theta(u)}{\partial u}\right|_{u=x}$ denotes the input Jacobian of D_θ at x. Consider the first-order local approximation*

$$D_\theta(x + \sigma\epsilon) \approx D_\theta(x) + \sigma J_D(x)\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

Then the expected locally linearized TRACE objective satisfies

$$\mathbb{E}_\epsilon[\mathcal{L}] \approx \underbrace{\frac{1}{2}\|AD_\theta(x) - y\|_2^2 + \frac{\beta}{2}\|D_\theta(x) - x\|_2^2}_{\text{unperturbed TRACE objective}} + \underbrace{\frac{\sigma^2}{2}\|AJ_D(x)\|_F^2}_{\text{measurement-space sensitivity}} + \underbrace{\frac{\beta\sigma^2}{2}\|J_D(x)\|_F^2}_{\text{SPC-induced regularization}}.$$

The full proof is given in Appendix A.2. The first two terms in Proposition 4.3 recover the unperturbed TRACE objective, while the remaining terms characterize the local effect of stochastic perturbation. The measurement-space sensitivity term arises through the data-fidelity term and controls perturbation sensitivity after application of the forward operator. In contrast, the SPC-induced regularization acts directly on the full Jacobian, thereby controlling local perturbation sensitivity in the reconstruction space. To understand how this regularization acts on measurement-visible and measurement-invisible directions, consider

$$R_{\text{pert}} = \frac{\sigma^2}{2}\|AJ_D(x)\|_F^2 + \frac{\beta\sigma^2}{2}\|J_D(x)\|_F^2.$$

Let $P_{\mathcal{R}}$ and $P_{\mathcal{N}}$ denote the orthogonal projections onto $\mathrm{range}(A^\top)$ and $\mathrm{null}(A)$, respectively. Since $P_{\mathcal{R}} + P_{\mathcal{N}} = I$ and $AP_{\mathcal{N}} = 0$,

$$R_{\text{pert}} = \frac{\sigma^2}{2}\|AP_{\mathcal{R}}J_D(x)\|_F^2 + \frac{\beta\sigma^2}{2}\|P_{\mathcal{R}}J_D(x)\|_F^2 + \frac{\beta\sigma^2}{2}\|P_{\mathcal{N}}J_D(x)\|_F^2.$$

The noise-coupling interaction introduces an additional null-space regularization $\frac{\beta\sigma^2}{2}\|P_{\mathcal{N}}J_D(x)\|_F^2$, which is absent when temporal coupling is removed. This motivates the null-space analysis in Section C.1, where we further examine the source of the reconstruction improvement by evaluating the PSNR of the reconstructions projected onto $\mathrm{null}(A)$. We also approximately estimate the denoiser Jacobian and verify that SPC suppresses its null-space component, consistent with the additional regularization predicted by the analysis and absent under stochastic perturbation alone.

Another way to interpret this proposition is through spectral bias. Untrained network priors are known to learn low-frequency image components before higher-frequency details Shi et al. (2021). Function smoothness is closely related to this frequency bias, and input–output Jacobian regularization has been used to promote smoothness and control the spectral bias of untrained neural networks (Liu et al., 2024). The SPC-induced term directly regularizes this input-output Jacobian. Thus, beyond the regularization induced by stochastic perturbation alone, SPC provides an additional mechanism for modulating the network's inherent spectral bias, with strength jointly controlled by β and σ^2. This motivates the spectral-bias analysis in Section C.2, where we use frequency-band correspondence to empirically examine how SPC changes the low-to-high frequency progression relative to standard DIP, a sequential variant that passes each reconstruction as the input to the next outer step, and stochastic perturbation alone.

5 Experiments

We design the experiments around the two questions motivating TRACE: whether competitive reconstruction can be achieved without relying on domain-matched pretrained priors, and how explicitly constraining successive states affects the reconstruction process.

We first consider CT reconstruction, where several competing methods have access to domain-specific pretrained diffusion models or supervised reconstruction networks. We then evaluate natural-image inverse problems, where strong pretrained diffusion and CNN priors are readily available. Together, these settings evaluate whether TRACE can remain competitive without external training across domains where competing methods benefit from domain-specific learned priors.

Beyond final reconstruction performance, we provide controlled analyses of stochastic perturbation and proximal coupling in the appendix to examine how SPC affects the reconstruction trajectory and how the observed behavior relates to Section 4. Finally, we add the same SPC constraint to several pretrained diffusion-based inverse solvers to test whether its effect extends to other reconstruction frameworks. Additional ablations, sensitivity studies, and runtime comparisons are provided in the Appendix G.

5.1 Experimental Setup

We evaluate TRACE on two CT reconstruction tasks, sparse-view and limited-angle CT, and three natural-image restoration tasks: inpainting, super-resolution, and deblurring. For CT, we use 24 abdominal slices from the AAPM dataset. Sparse-view CT uses 60 projection views over 180°, whereas limited-angle CT uses 120 projection views within the angular range 0°–119°. For natural-image restoration, we use 14 test images from BSDS68 and Set14. We compare TRACE with plug-and-play methods using pretrained CNN denoisers: PnP-DnCNN Zhang et al. (2017a), PnP-FFDNet Zhang et al. (2018), and IRCNN Zhang et al. (2017b). For CT, we also evaluate the supervised networks RED-CNN Chen et al. (2017) and EDCNN Liang et al. (2020). Methods using pretrained diffusion priors include SITCOM Alkhouri et al. (2025), DPS Chung et al. (2023a), DAPS Zhang et al. (2025), RED-Diff Mardani et al. (2024), PnP-DM Wu et al. (2024b), and DDS Chung et al. (2023b). The hybrid method uDiG-DIP Liang et al. (2025b) combines diffusion purification with DIP optimization. Methods that optimize a randomly initialized network for each test instance include DIP Ulyanov et al. (2018), aSeqDIP Alkhouri et al. (2024), Self-Guided DIP Liang et al. (2025c), and SDI Luo et al. (2026). We also include UGoDIT Liang et al. (2025a), which uses transferable weights learned from a group of degraded measurements.

For each task, the evaluated methods use the same test images, measurement realizations, and forward operator. Unless stated otherwise, TRACE uses $T = 40$ trajectory states and $K = 150$ inner optimization steps per state. Further experimental settings and runtime comparisons are provided in Appendix D.

Table 1: Average results on CT reconstruction. Best and second-best values in each row are shown in **bold** and underlined, respectively. DDS and uDiG-DIP use CT-domain pretrained diffusion priors, whereas SITCOM uses a natural-image pretrained diffusion prior. IRCNN uses a pretrained natural-image CNN denoiser, while RED-CNN and EDCNN are supervised CT reconstruction networks.

Setting	Metric	CNN-based methods			Diffusion-prior methods			Neural-prior methods				
		IRCNN	RED-CNN	EDCNN	DDS	uDiG-DIP	SITCOM	DIP	ASeqDIP	Self-Guided DIP	SDI	TRACE
Sparse-view	PSNR↑	29.44	29.32	29.11	**30.55**	27.79	21.44	25.25	27.15	27.43	24.36	29.55
	SSIM↑	0.873	0.841	0.825	0.832	0.807	0.746	0.691	0.822	0.809	0.789	**0.894**
	LPIPS↓	0.055	0.062	**0.050**	0.124	0.138	0.174	0.211	0.127	0.103	0.153	0.057
Limited-angle	PSNR↑	26.22	24.91	23.73	25.51	25.12	18.67	23.36	25.64	26.33	25.36	**27.34**
	SSIM↑	**0.806**	0.766	0.719	0.797	0.733	0.600	0.626	0.760	0.782	0.768	0.804
	LPIPS↓	0.108	0.115	0.104	**0.082**	0.196	0.240	0.276	0.164	0.113	0.151	0.128

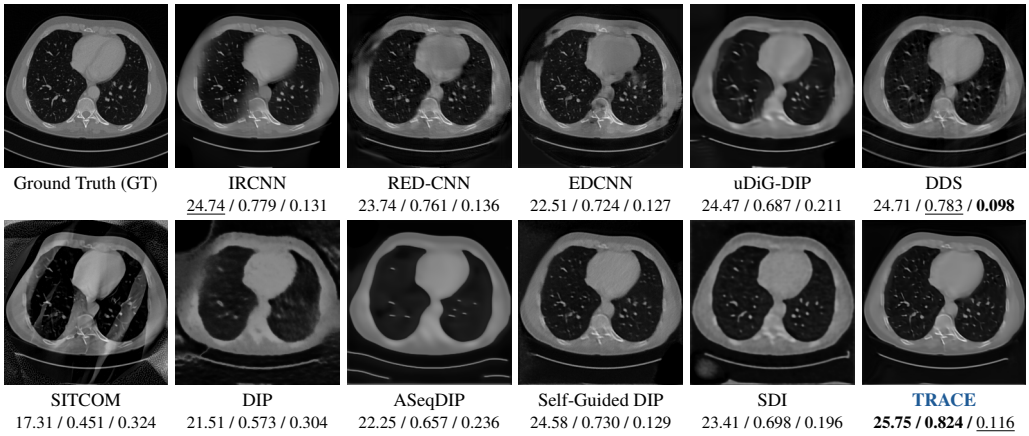


Figure 1: Representative limited-angle CT reconstructions (PSNR↑ / SSIM↑/ LPIPS↓). TRACE preserves anatomical structures more clearly and achieves the highest PSNR and SSIM in this example.

5.2 CT Reconstruction

We first consider CT reconstruction, where the compared methods use different sources of prior information. DDS and uDiG-DIP use diffusion models pretrained on CT images, while RED-CNN and EDCNN are supervised on CT data. IRCNN uses a CNN denoiser pretrained on natural images, and SITCOM uses a diffusion model pretrained on natural images. TRACE and other compared neural-prior methods, in contrast, are optimized directly from the observed measurements and do not require external training data or a pretrained model. This setting allows us to compare training-free reconstruction with both domain-matched learned priors and generic pretrained priors.

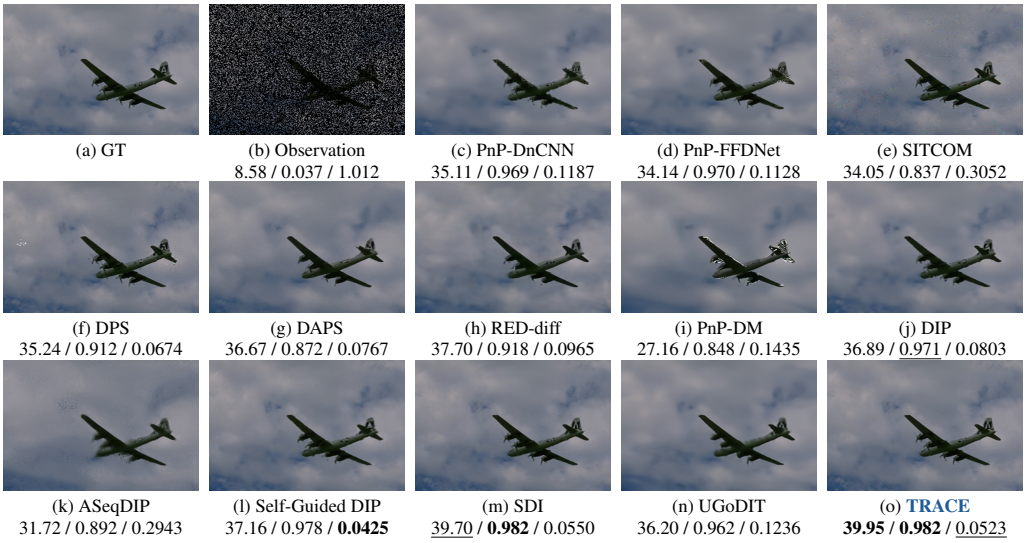
Table 1 shows that TRACE remains competitive with methods using externally learned priors. For sparse-view CT, DDS achieves the highest PSNR, while TRACE obtains the second-best one. DDS benefits from a domain-matched pretrained diffusion prior, whereas TRACE uses no CT training data or pretrained reconstruction model. Despite this difference, TRACE remains within 1.00 dB of DDS and achieves the highest SSIM. For limited-angle CT, TRACE obtains the highest PSNR, 1.01 dB above the second-best one, while our SSIM is close to the best value obtained by IRCNN.

The comparison with SITCOM further illustrates the role of domain matching. SITCOM performs strongly on natural images when its pretrained prior is matched to the image domain, as shown in the next subsection 5.3. When the same natural-image prior is directly applied to CT, its PSNR drops to 21.44 dB for sparse-view and 18.67 dB for limited-angle reconstruction. This is different from DDS, which uses a CT-pretrained diffusion prior and performs strongly on sparse-view CT. TRACE does not require a domain-specific pretrained model and is optimized directly from the measurements.

Figure 1 shows a representative limited-angle reconstruction. The cross-domain SITCOM reconstruction largely fails to recover the underlying CT anatomy, with severe structural distortions caused

Table 2: Average results on natural-image inverse problems. Best and second-best values for each setting and metric are shown in **bold** and underlined, respectively. UGoDIT* uses transferable weights learned from unsupervised group training. PnP-DnCNN and PnP-FFDNet plug pretrained CNN denoisers into a plug-and-play proximal-gradient solver.

Task	Setting	Metric	CNN-denoiser methods		Diffusion-prior methods					Neural-prior methods					
			PnP-DnCNN	PnP-FFDNet	SITCOM	DPS	DAPS	RED-diff	PnP-DM	DIP	ASeqDIP	Self-Guided DIP	SDI	UGoDIT*	TRACE
Inpainting	Missing 50%	PSNR↑	31.45	29.67	33.60	30.80	31.48	31.82	29.11	29.08	31.84	**33.91**	32.53	30.93	33.77
		SSIM↑	0.943	0.926	0.940	0.928	0.922	0.929	0.882	0.832	0.919	**0.951**	0.938	0.894	0.948
		LPIPS↓	0.0668	0.1010	0.0372	0.0373	0.0556	0.0762	0.1128	0.2299	0.0773	0.0337	**0.0315**	0.0778	0.0316
	Missing 70%	PSNR↑	28.53	26.82	29.31	28.00	28.79	29.11	25.82	27.94	29.14	30.61	29.27	28.96	**30.68**
		SSIM↑	0.889	0.864	0.855	0.876	0.876	0.891	0.837	0.805	0.871	**0.909**	0.886	0.864	0.903
		LPIPS↓	0.1306	0.1765	0.0883	0.0718	0.1024	0.1171	0.1618	0.2499	0.1166	0.0787	0.0600	0.0938	**0.0553**
SR	×2	PSNR↑	32.07	30.98	**32.55**	29.27	30.42	29.65	25.99	26.98	28.51	28.42	31.28	29.63	32.01
		SSIM↑	**0.933**	0.924	0.929	0.887	0.893	0.885	0.835	0.748	0.827	0.777	0.920	0.872	0.931
		LPIPS↓	0.0812	0.1055	**0.0416**	0.0515	0.0700	0.1254	0.1446	0.3285	0.2519	0.1374	0.0772	0.0926	0.0755
	×4	PSNR↑	27.09	26.78	**27.50**	25.08	26.32	24.63	23.80	26.11	25.88	22.26	26.32	25.91	27.47
		SSIM↑	0.805	0.808	0.806	0.773	0.795	0.762	0.762	0.723	0.744	0.478	0.783	0.755	**0.817**
		LPIPS↓	0.2661	0.2846	0.1655	**0.1304**	0.1902	0.2762	0.2533	0.3828	0.3500	0.5732	0.2713	0.2373	0.2484
Deblurring	Motion	PSNR↑	27.95	27.00	**33.89**	30.03	27.82	25.61	24.78	24.36	24.58	29.21	29.25	25.24	29.94
		SSIM↑	0.795	0.778	**0.932**	0.879	0.824	0.782	0.774	0.652	0.654	0.829	0.832	0.696	0.863
		LPIPS↓	0.2386	0.3007	**0.0588**	0.0875	0.1571	0.2385	0.2240	0.4522	0.4140	0.1760	0.1899	0.2968	0.1840
	Nonlinear	PSNR↑	24.22	24.11	**26.35**	20.53	24.40	23.59	22.55	23.59	22.85	23.11	24.69	23.29	25.52
		SSIM↑	0.645	0.643	**0.733**	0.595	0.730	0.725	0.713	0.612	0.569	0.492	0.665	0.600	0.708
		LPIPS↓	0.4432	0.5140	**0.2591**	0.3119	0.2780	0.3147	0.3094	0.5192	0.4977	0.5298	0.4156	0.4323	0.3986



(a) GT
(b) Observation 8.58 / 0.037 / 1.012
(c) PnP-DnCNN 35.11 / 0.969 / 0.1187
(d) PnP-FFDNet 34.14 / 0.970 / 0.1128
(e) SITCOM 34.05 / 0.837 / 0.3052
(f) DPS 35.24 / 0.912 / 0.0674
(g) DAPS 36.67 / 0.872 / 0.0767
(h) RED-diff 37.70 / 0.918 / 0.0965
(i) PnP-DM 27.16 / 0.848 / 0.1435
(j) DIP 36.89 / 0.971 / 0.0803
(k) ASeqDIP 31.72 / 0.892 / 0.2943
(l) Self-Guided DIP 37.16 / 0.978 / **0.0425**
(m) SDI 39.70 / **0.982** / 0.0550
(n) UGoDIT 36.20 / 0.962 / 0.1236
(o) **TRACE** **39.95** / **0.982** / 0.0523

Figure 2: Inpainting result (PSNR↑ / SSIM↑ / LPIPS↓) with 70% missing pixels.

by the mismatch between its natural-image prior and the CT domain. DDS, despite using a CT-domain pretrained diffusion prior, still exhibits visible streak and texture artifacts. TRACE has better preserved anatomical structures and achieves the highest PSNR and SSIM. The averages in Table 1 are computed over 24 test slices, additional settings and results are provided in Appendix E.

5.3 Natural-Image Inverse Problems

We consider natural-image inverse problems, where pretrained diffusion and CNN priors are readily available. We examine whether removing external pretraining leads to a substantial loss in reconstruction quality. We evaluate image inpainting, super-resolution, and deblurring, with the results summarized in Table 2. Among methods that are optimized without external training, TRACE achieves the highest PSNR in five of the six settings. The advantage is more apparent in the harder reconstruction settings. For 70% missing inpainting, TRACE achieves the best PSNR and LPIPS among all compared methods, while remaining close to the best SSIM. For ×4 super-resolution, TRACE obtains 27.47 dB, nearly matching SITCOM at 27.50 dB, and achieves the highest SSIM.

Table 3: Effect of SPC on pretrained diffusion inverse solvers. Δ denotes the change in mean PSNR (dB), and the number in parentheses reports how many test images or slices obtain higher PSNR with SPC out of the total number evaluated.

	FFHQ Deblur			**MRI** $4\times$			**MRI** $8\times$		
Solver	**Base**	**+SPC**	Δ	**Base**	**+SPC**	Δ	**Base**	**+SPC**	Δ
DPS	24.30	**25.95**	+1.64 (17/20)	33.40	**33.42**	+0.02 (5/10)	29.43	29.43	+0.00 (6/10)
DAPS	28.12	**28.32**	+0.20 (19/20)	31.33	**31.70**	+0.37 (10/10)	29.02	**29.33**	+0.31 (10/10)
RED-diff	27.47	**27.60**	+0.13 (16/20)	30.67	**31.04**	+0.37 (10/10)	28.42	**28.71**	+0.28 (8/10)
PnP-DM	25.39	**25.51**	+0.12 (13/20)	29.35	**29.54**	+0.19 (9/10)	27.58	**27.72**	+0.14 (9/10)
SITCOM	27.91	**28.55**	+0.64 (19/20)	34.54	**34.59**	+0.05 (10/10)	31.97	**32.08**	+0.11 (9/10)
Mean			**+0.55** (84/100)			**+0.20** (44/50)			**+0.17** (42/50)

The pretrained methods remain stronger in several cases. In particular, SITCOM achieves the best results for both deblurring settings. These results are consistent with the CT comparison (Table 1): a well-matched pretrained prior can provide a clear advantage, whereas TRACE does not require such a model and remains competitive across different inverse problems.

Figure 2 shows a representative result for 70% missing inpainting. TRACE recovers the missing structures with fewer visible artifacts and achieves the highest PSNR. Additional results for inpainting, super-resolution, and deblurring are provided in Appendix F.

5.4 Generality to Pretrained Diffusion Solvers

Although TRACE uses an untrained neural network, the trajectory constraint is not specific to this parameterization. To examine whether the same idea extends beyond TRACE, we augment five pretrained diffusion-based inverse solvers with the proximal state-consistency term used in SPC. For each method, we compare the original solver (Base) with its SPC-augmented version (+SPC). We consider Gaussian deblurring on FFHQ and multi-coil MRI reconstruction on fastMRI Knoll et al. (2020) at $4\times$ and $8\times$ acceleration. Experimental settings are provided in Appendix H.

Table 3 reports the effect of adding the state-consistency constraint to the five pretrained diffusion solvers. The average gain is 0.55 dB on FFHQ deblurring, with larger improvements for DPS ($+1.64$ dB) and SITCOM ($+0.64$ dB). For MRI, the average gains are 0.20 dB at $4\times$ acceleration and 0.17 dB at $8\times$ acceleration. The improvement is smaller for DPS on MRI, while DAPS and RED-diff show consistent gains at both acceleration factors. Although SPC does not improve every individual reconstruction, positive changes are observed on most test cases. Across the five solvers, SPC improves PSNR on 84 of the 100 FFHQ deblur evaluations, 44 of the 50 evaluations for $4\times$ MRI, and 42 of the 50 for $8\times$ MRI.

These results show that the trajectory constraint is not tied to the untrained network used in TRACE. This supports the broader role of trajectory constraint for controlling intermediate reconstruction states.

6 Conclusion

We presented TRACE, a training-free framework for imaging inverse problems that reconstructs the image through a sequence of intermediate states rather than relying on a pretrained diffusion model. TRACE uses stochastic proximal consistency to constrain successive states, and the resulting formulation admits both a proximal interpretation and a local Jacobian analysis. Experiments on natural-image restoration and CT reconstruction show that TRACE remains competitive with methods using pretrained or supervised priors while requiring no external training data. We also find that the same state-consistency constraint can improve existing diffusion inverse solvers. These results suggest that explicitly modeling the reconstruction trajectory provides a useful alternative to relying only on a learned prior or a final reconstruction objective.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, improving the clarity of the manuscript, and assisting with formatting the experimental presentation. All technical content, including the methodology, theoretical results, experimental design, implementation, and reported results, was independently checked and verified by the authors. The authors take full responsibility for the final contents of the paper.

REFERENCES

Ismail Alkhouri, Shijun Liang, Cheng-Han Huang, Jimmy Dai, Qing Qu, Saiprasad Ravishankar, and Rongrong Wang. Sitcom: Step-wise triple-consistent diffusion sampling for inverse problems. In *International Conference on Machine Learning*, pp. 1128–1158. PMLR, 2025.

Ismail R Alkhouri, Shijun Liang, Evan Bell, Qing Qu, Rongrong Wang, and Saiprasad Ravishankar. Image reconstruction via autoencoding sequential deep image prior. *Advances in Neural Information Processing Systems*, 37:18988–19012, 2024.

Markus Bachmayr and Martin Burger. Iterative total variation schemes for nonlinear inverse problems. *Inverse Problems*, 25(10):105004, 2009. doi: 10.1088/0266-5611/25/10/105004.

Martin Benning and Martin Burger. Modern regularization methods for inverse problems, 2018. URL https://arxiv.org/abs/1801.09922.

Weike Chang, Nicola D'Ascenzo, and Qingguo Xie. A relaxed iterated tikhonov regularization for linear ill-posed inverse problems. *Journal of Mathematical Analysis and Applications*, 530(2): 127754, 2024. doi: 10.1016/j.jmaa.2023.127754.

Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE Transactions on Medical Imaging*, 36(12):2524–2535, 2017.

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*, 2023a.

Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. *arXiv preprint arXiv:2303.05754*, 2023b.

Damek Davis, Mateo Díaz, and Dmitriy Drusvyatskiy. Escaping strict saddle points of the moreau envelope in nonsmooth optimization, 2021. URL https://arxiv.org/abs/2106.09815.

Zhenghan Fang, Sam Buchanan, and Jeremias Sulam. What's in a prior? learned proximal networks for inverse problems. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 55445–55485, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/f4757db82a02eea015670ecca605d5cc-Paper-Conference.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Chaoyan Huang, Zhongming Wu, Yanqi Cheng, Tieyong Zeng, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Deep block proximal linearized minimization algorithm for nonconvex inverse problems. *SIAM Journal on Mathematics of Data Science*, 7(4):1729–1754, 2025.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1724–1732. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/jin17a.html.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Florian Knoll, Jure Zbontar, Anuroop Sriram, Matthew J. Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, James Pinkerton, Duo Wang, Nafissa Yakubova, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: A publicly available raw k-space and DICOM dataset of knee images for accelerated MR image reconstruction using machine learning. *Radiology: Artificial Intelligence*, 2(1):e190007, 2020. doi: 10.1148/ryai.2020190007.

Anselm Krainovic, Mahdi Soltanolkotabi, and Reinhard Heckel. Learning provably robust estimators for inverse problems via jittering. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 57276–57305. Curran Associates, Inc., 2023. doi: 10.52202/075280-2503. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b3411e30afa6caeefa4d6d39a5ea84cd-Paper-Conference.pdf.

Haeil Lee, Hansang Lee, Seoyeon Gye, and Junmo Kim. Beta sampling is all you need: Efficient image generation strategy for diffusion models using stepwise spectral analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4215–4224, 2025. doi: 10.1109/WACV61041.2025.00414.

Shijun Liang, Ismail Alkhouri, Siddhant Gautam, Qing Qu, and Saiprasad Ravishankar. Ugodit: Unsupervised group deep image prior via transferable weights. *Advances in Neural Information Processing Systems*, 2025a.

Shijun Liang, Ismail Alkhouri, Qing Qu, Rongrong Wang, and Saiprasad Ravishankar. Sequential diffusion-guided deep image prior for medical image reconstruction. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025b.

Shijun Liang, Evan Bell, Qing Qu, Rongrong Wang, and Saiprasad Ravishankar. Analysis of deep image prior and exploiting self-guidance for image reconstruction. *IEEE Transactions on Computational Imaging*, 2025c.

Tengfei Liang, Yi Jin, Yidong Li, and Tao Wang. EDCNN: Edge enhancement-based densely connected network with compound loss for low-dose CT denoising. In *2020 15th IEEE International conference on signal processing (ICSP)*, volume 1, pp. 193–198. IEEE, 2020.

Yilin Liu, Yunkui Pang, Jiang Li, Yong Chen, and Pew-Thian Yap. Towards architecture-agnostic untrained network priors for image reconstruction with frequency regularization, 2024. URL https://arxiv.org/abs/2312.09988.

Guanxiong Luo, Shoujin Huang, and Yanlong Yang. Self-diffusion for solving inverse problems. In *Advances in Neural Information Processing Systems*, 2026.

Michael Lustig, David Donoho, and John M. Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. doi: 10.1002/mrm.21391.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 28027–28053, 2024.

Cynthia H. McCollough, Adam C. Bartley, Rickey E. Carter, Baiyu Chen, Tammy A. Drees, Phillip Edwards, David R. Holmes, III, Alice E. Huang, Farhana Khan, Shuai Leng, Kyle L. McMillan, Gregory J. Michalak, Kristina M. Nunez, Lifeng Yu, and Joel G. Fletcher. Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 low dose CT grand challenge. *Medical Physics*, 44(10):e339–e352, 2017. doi: 10.1002/mp.12345.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965. doi: 10.24033/bsmf.1625.

OpenAI. guided-diffusion. https://github.com/openai/guided-diffusion, 2021. GitHub repository. Accessed: 2026-05-07.

Sathish Ramani, Thierry Blu, and Michael Unser. Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on Image Processing*, 17(9):1540–1554, 2008. doi: 10.1109/TIP.2008.2001404.

Saiprasad Ravishankar, Jong Chul Ye, and Jeffrey A. Fessler. Image reconstruction: From sparsity to data-adaptive methods and machine learning. *Proceedings of the IEEE*, 108(1):86–109, 2020. doi: 10.1109/JPROC.2019.2936204.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976. doi: 10.1137/0314056.

Zenglin Shi, Pascal Mettes, Subhransu Maji, and Cees G. M. Snoek. On measuring and controlling the spectral bias of the deep image prior, 2021. URL https://arxiv.org/abs/2107.01125.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022.

Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11956–11965, June 2021.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

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. IEEE, 2013.

Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun. Early stopping for deep image prior. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=231ZzrLC8X.

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

Zhongming Wu, Chaoyan Huang, and Tieyong Zeng. Extrapolated plug-and-play three-operator splitting methods for nonconvex optimization with applications to image restoration. *SIAM Journal on Imaging Sciences*, 17(2):1145–1181, 2024a.

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. *Advances in Neural Information Processing Systems*, 37:118389–118427, 2024b.

Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. doi: 10.1109/CVPR.2008.4587647.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20895–20905, 2025.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017a.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3929–3938, 2017b.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy Feng, Caifeng Zou, Yu Sun, Nikola Borislavov Kovachki, Zachary E Ross, Katherine Bouman, and Yisong Yue. Inversebench: Benchmarking plug-and-play diffusion models for scientific inverse problems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=U3PBITXNG6.

APPENDIX

The appendix is organized as follows.

- Appendix A provides additional theoretical results for the proximal formulation, including the effect of the coupling strength, the relation between local optimization error and the exact proximal update, error propagation along the trajectory, and the derivation of the noise-coupling interaction in Proposition 4.3.
- Appendix B presents the empirical analysis of stochastic proximal consistency (SPC), including component ablations, controlled experiments that separate stochastic perturbation and proximal coupling, and their sensitivity to the corresponding hyperparameters.
- Appendix C connects the empirical observations to the theoretical analysis through null-space reconstruction, Jacobian sensitivity, and spectral-bias experiments.
- Appendix D gives implementation details for TRACE and the compared methods, together with the evaluation protocol and task-specific settings.
- Appendix E provides detailed settings and additional results for CT reconstruction.
- Appendix F provides detailed settings and additional qualitative reconstruction results for natural-image restoration.
- Appendix G collects additional ablations and computational analyses, including the allocation of the optimization budget between outer and inner iterations, pretrained initialization, and runtime and memory costs.
- Appendix H provides the experimental protocol and solver-specific details for incorporating the state-consistency constraint into pretrained diffusion inverse solvers.

A ADDITIONAL THEORETICAL ANALYSIS

A.1 ADDITIONAL RESULTS FOR STOCHASTIC PROXIMAL CONSISTENCY

In this subsection, we give additional results for the proximal model in equation 2. We first consider the exact proximal update and study the effect of the coupling parameter β_t. We then relate the suboptimality gap ε_t in equation 8 to the distance from the exact proximal update and study how these errors propagate along the reconstruction trajectory.

Recall that

$$\Phi_t(\mathbf{x};\mathbf{v}) = F_t(\mathbf{x}) + \frac{\beta_t}{2}\|\mathbf{x}-\mathbf{v}\|_2^2, \tag{11}$$

where $F_t(\mathbf{x}) = f(\mathbf{x}) + \lambda_t R(\mathbf{x})$. Unless stated otherwise, we assume that F_t is proper, lower semicontinuous, and bounded below. These are the same general conditions used in the main text.

A.1.1 EXACT PROXIMAL UPDATE

We first consider the exact local problem corresponding to $\varepsilon_t = 0$. For a fixed neighboring state $\mathbf{v}$ and coupling parameter $\beta > 0$, define

$$\mathbf{x}^\star(\beta) \in \arg\min_{\mathbf{x}} \left\{ F_t(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{x}-\mathbf{v}\|_2^2 \right\}. \tag{12}$$

Since F_t is bounded below, the objective in equation 12 is coercive for every $\beta > 0$. Together with properness and lower semicontinuity, this guarantees the existence of at least one global minimizer.

Proposition A.1. *Let $0 < \beta_1 < \beta_2$, and let $\mathbf{x}^{\star,1}$ and $\mathbf{x}^{\star,2}$ be any global minimizers of equation 12 corresponding to β_1 and β_2, respectively. Then*

$$\|\mathbf{x}^{\star,2}-\mathbf{v}\|_2 \le \|\mathbf{x}^{\star,1}-\mathbf{v}\|_2, \tag{13}$$

and

$$F_t(\mathbf{x}^{\star,1}) \le F_t(\mathbf{x}^{\star,2}). \tag{14}$$

Proof. Define $d_1 := \|\mathbf{x}^{\star,1} - \mathbf{v}\|_2$, $d_2 := \|\mathbf{x}^{\star,2} - \mathbf{v}\|_2$. Since $\mathbf{x}^{\star,1}$ is a global minimizer for β_1,

$$F_t(\mathbf{x}^{\star,1}) + \frac{\beta_1}{2} d_1^2 \le F_t(\mathbf{x}^{\star,2}) + \frac{\beta_1}{2} d_2^2. \tag{15}$$

Similarly, optimality of $\mathbf{x}^{\star,2}$ for β_2 gives

$$F_t(\mathbf{x}^{\star,2}) + \frac{\beta_2}{2} d_2^2 \le F_t(\mathbf{x}^{\star,1}) + \frac{\beta_2}{2} d_1^2. \tag{16}$$

Hence equation 15 and equation 16 yields

$$\frac{\beta_2 - \beta_1}{2}(d_2^2 - d_1^2) \le 0. \tag{17}$$

Since $\beta_2 > \beta_1$, we obtain

$$d_2 \le d_1, \tag{18}$$

which proves equation 13.

Using equation 15 and $d_2^2 \le d_1^2$,

$$F_t(\mathbf{x}^{\star,1}) - F_t(\mathbf{x}^{\star,2}) \le \frac{\beta_1}{2}(d_2^2 - d_1^2) \le 0, \tag{19}$$

which proves equation 14. □

Proposition A.1 gives a stronger statement than in the main text for the exact local problem. For a fixed neighboring state $\mathbf{v}$, increasing the coupling parameter cannot increase the distance of the proximal solution from $\mathbf{v}$. At the same time, the value of the uncoupled objective F_t cannot decrease. Thus, β_t controls the balance between minimizing F_t and remaining close to the neighboring trajectory state.

Remark A.2. *The comparison in Proposition A.1 is local. The state* $\mathbf{v}$ *is kept fixed when* β *is varied. It does not imply monotonicity of the complete TRACE trajectory when all trajectory states are recomputed for different coupling schedules.*

A.1.2 From Proximal Suboptimality to State Error

The main text defines

$$\varepsilon_t = \Phi_t(\mathbf{x}_t; \mathbf{x}_{t+1}) - \inf_{\mathbf{x}} \Phi_t(\mathbf{x}; \mathbf{x}_{t+1}), \tag{20}$$

where $\mathbf{x}_t = D_{\theta_t^K}(\mathbf{u}_t)$ is the implemented neural state. Without additional curvature, a bound on the objective gap ε_t does not in general imply a bound on the distance between $\mathbf{x}_t$ and a minimizer. We therefore consider a stronger condition for the remainder of this section.

Recall that F_t is ρ_t-weakly convex if $F_t(\mathbf{x}) + \frac{\rho_t}{2}\|\mathbf{x}\|_2^2$ is convex.

Lemma A.3. *Suppose that* F_t *is* ρ_t*-weakly convex and* $\beta_t > \rho_t$*. Then, for every* $\mathbf{v}$*,* $\Phi_t(\cdot; \mathbf{v})$ *is* $(\beta_t - \rho_t)$*-strongly convex and therefore has a unique global minimizer. Denote this minimizer by* $P_t(\mathbf{v}) := \arg\min_{\mathbf{x}} \Phi_t(\mathbf{x}; \mathbf{v})$*. Then the implemented state satisfies*

$$\|\mathbf{x}_t - P_t(\mathbf{x}_{t+1})\|_2 \le \sqrt{\frac{2\varepsilon_t}{\beta_t - \rho_t}}. \tag{21}$$

Proof. Since F_t is ρ_t-weakly convex, $\Phi_t(\mathbf{x}; \mathbf{v}) = F_t(\mathbf{x}) + \frac{\beta_t}{2}\|\mathbf{x} - \mathbf{v}\|_2^2$ is $(\beta_t - \rho_t)$-strongly convex whenever $\beta_t > \rho_t$. Hence, it has a unique global minimizer $P_t(\mathbf{v})$.

By strong convexity and optimality of $P_t(\mathbf{x}_{t+1})$,

$$\Phi_t(\mathbf{x}_t; \mathbf{x}_{t+1}) - \Phi_t(P_t(\mathbf{x}_{t+1}); \mathbf{x}_{t+1}) \ge \frac{\beta_t - \rho_t}{2}\|\mathbf{x}_t - P_t(\mathbf{x}_{t+1})\|_2^2. \tag{22}$$

The left-hand side is ε_t by equation 20. Therefore,

$$\frac{\beta_t - \rho_t}{2}\|\mathbf{x}_t - P_t(\mathbf{x}_{t+1})\|_2^2 \le \varepsilon_t, \tag{23}$$

which gives equation 21. □

A.1.3 Propagation Along the Trajectory

We now compare the implemented TRACE trajectory with the trajectory obtained by exact proximal updates. Starting from the same initial state, define

$$\widetilde{\mathbf{x}}_T = \mathbf{x}_T, \qquad \widetilde{\mathbf{x}}_t = P_t(\widetilde{\mathbf{x}}_{t+1}), \quad t = T-1, \dots, 0. \tag{24}$$

We first record the Lipschitz property of P_t under weak convexity.

Lemma A.4. *Suppose that F_t is ρ_t-weakly convex and $\beta_t > \rho_t$. Then*

$$\|P_t(\mathbf{a}) - P_t(\mathbf{b})\|_2 \le \frac{\beta_t}{\beta_t - \rho_t}\|\mathbf{a} - \mathbf{b}\|_2 \tag{25}$$

for all $\mathbf{a}, \mathbf{b}$.

Proof. Let $\mathbf{p} = P_t(\mathbf{a})$, $\mathbf{q} = P_t(\mathbf{b})$, and define $m_t = \beta_t - \rho_t > 0$. The functions $\Phi_t(\cdot; \mathbf{a})$ and $\Phi_t(\cdot; \mathbf{b})$ are m_t-strongly convex.

Since $\mathbf{p}$ minimizes $\Phi_t(\cdot; \mathbf{a})$,

$$\Phi_t(\mathbf{q}; \mathbf{a}) \ge \Phi_t(\mathbf{p}; \mathbf{a}) + \frac{m_t}{2}\|\mathbf{p} - \mathbf{q}\|_2^2. \tag{26}$$

Similarly,

$$\Phi_t(\mathbf{p}; \mathbf{b}) \ge \Phi_t(\mathbf{q}; \mathbf{b}) + \frac{m_t}{2}\|\mathbf{p} - \mathbf{q}\|_2^2. \tag{27}$$

Adding equation 26 and equation 27, the F_t terms cancel and we obtain

$$m_t\|\mathbf{p} - \mathbf{q}\|_2^2 \le \beta_t\langle \mathbf{a} - \mathbf{b}, \mathbf{p} - \mathbf{q}\rangle. \tag{28}$$

Applying Cauchy–Schwarz gives

$$(\beta_t - \rho_t)\|\mathbf{p} - \mathbf{q}\|_2^2 \le \beta_t\|\mathbf{a} - \mathbf{b}\|_2\|\mathbf{p} - \mathbf{q}\|_2. \tag{29}$$

If $\mathbf{p} \neq \mathbf{q}$, division by $\|\mathbf{p} - \mathbf{q}\|_2$ gives equation 25. If $\mathbf{p} = \mathbf{q}$, the result is immediate. □

Theorem A.5. *Suppose that, for every $t = T-1, \dots, 0$, F_t is ρ_t-weakly convex and $\beta_t > \rho_t$. Then*

$$\|\mathbf{x}_t - \widetilde{\mathbf{x}}_t\|_2 \le \delta_t + \frac{\beta_t}{\beta_t - \rho_t}\|\mathbf{x}_{t+1} - \widetilde{\mathbf{x}}_{t+1}\|_2, \tag{30}$$

where $\delta_t = \sqrt{\frac{2\varepsilon_t}{\beta_t - \rho_t}}$ is the right-hand side of equation 21.

Consequently,

$$\|\mathbf{x}_t - \widetilde{\mathbf{x}}_t\|_2 \le \sum_{s=t}^{T-1} \delta_s \prod_{i=t}^{s-1} \frac{\beta_i}{\beta_i - \rho_i}, \tag{31}$$

where the empty product is defined as 1. *In particular,*

$$\|\mathbf{x}_0 - \widetilde{\mathbf{x}}_0\|_2 \le \sum_{s=0}^{T-1} \delta_s \prod_{i=0}^{s-1} \frac{\beta_i}{\beta_i - \rho_i}. \tag{32}$$

Proof. Using $\widetilde{\mathbf{x}}_t = P_t(\widetilde{\mathbf{x}}_{t+1})$, we have

$$\begin{aligned}\|\mathbf{x}_t - \widetilde{\mathbf{x}}_t\|_2 \le{}& \|\mathbf{x}_t - P_t(\mathbf{x}_{t+1})\|_2 \\ &+ \|P_t(\mathbf{x}_{t+1}) - P_t(\widetilde{\mathbf{x}}_{t+1})\|_2.\end{aligned} \tag{33}$$

By Lemma A.3,

$$\|\mathbf{x}_t - P_t(\mathbf{x}_{t+1})\|_2 \le \delta_t. \tag{34}$$

By Lemma A.4,

$$\|P_t(\mathbf{x}_{t+1}) - P_t(\widetilde{\mathbf{x}}_{t+1})\|_2 \le \frac{\beta_t}{\beta_t - \rho_t}\|\mathbf{x}_{t+1} - \widetilde{\mathbf{x}}_{t+1}\|_2. \tag{35}$$

Combining these two inequalities gives equation 30.

Since $\mathbf{x}_T = \widetilde{\mathbf{x}}_T$, repeated application of equation 30 gives equation 31. Setting $t = 0$ gives equation 32. □

Theorem A.5 controls the deviation between the implemented TRACE trajectory and the exact proximal trajectory.

If F_t is convex, we can take $\rho_t = 0$. Then

$$\delta_t = \sqrt{\frac{2\varepsilon_t}{\beta_t}}, \tag{36}$$

and equation 31 becomes

$$\|\mathbf{x}_t - \widetilde{\mathbf{x}}_t\|_2 \leq \sum_{s=t}^{T-1} \sqrt{\frac{2\varepsilon_s}{\beta_s}}. \tag{37}$$

Thus, in the convex case, the proximal maps are nonexpansive and the local approximation errors accumulate additively over the finite trajectory.

The weak and strong convexity assumptions in this appendix are used only to obtain the stronger trajectory-level results above. Under weak or strong convexity, this suboptimality can be converted into a distance from the exact proximal update, which then allows us to analyze error propagation along the trajectory.

A.2 Derivation of the Noise–Coupling Interaction

We provide the derivation of Proposition 4.3. Consider a single TRACE transition and fix the trajectory state x and network parameters θ. The perturbed network input is

$$x + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

where σ controls the perturbation magnitude. The corresponding TRACE objective is

$$\mathcal{L} = \frac{1}{2} \|AD_\theta(x + \sigma\epsilon) - y\|_2^2 + \frac{\beta}{2} \|D_\theta(x + \sigma\epsilon) - x\|_2^2 . \tag{38}$$

We study the expectation of this objective with respect to the stochastic perturbation. Using a first-order Taylor expansion of the network around x,

$$D_\theta(x + \sigma\epsilon) \approx D_\theta(x) + \sigma J_D(x)\epsilon, \tag{39}$$

where

$$J_D(x) = \left.\frac{\partial D_\theta(u)}{\partial u}\right|_{u=x}$$

denotes the Jacobian of the network output with respect to its input. The following derivation is therefore with respect to this locally linearized network mapping.

Substituting equation 39 into equation 38 gives

$$\begin{aligned} \mathcal{L} \approx \frac{1}{2} &\|A\left(D_\theta(x) + \sigma J_D(x)\epsilon\right) - y\|_2^2 \\ &+ \frac{\beta}{2} \|D_\theta(x) + \sigma J_D(x)\epsilon - x\|_2^2 . \end{aligned} \tag{40}$$

Expanding the data-fidelity term,

$$\begin{aligned} &\frac{1}{2} \|A\left(D_\theta(x) + \sigma J_D(x)\epsilon\right) - y\|_2^2 \\ =\ &\frac{1}{2} \|AD_\theta(x) - y\|_2^2 + \sigma \left(AD_\theta(x) - y\right)^\top AJ_D(x)\epsilon \\ &+ \frac{\sigma^2}{2} \epsilon^\top J_D(x)^\top A^\top AJ_D(x)\epsilon. \end{aligned} \tag{41}$$

Similarly, expanding the temporal-coupling term gives

$$\begin{aligned} &\frac{\beta}{2} \|D_\theta(x) + \sigma J_D(x)\epsilon - x\|_2^2 \\ =\ &\frac{\beta}{2} \|D_\theta(x) - x\|_2^2 + \beta\sigma \left(D_\theta(x) - x\right)^\top J_D(x)\epsilon \\ &+ \frac{\beta\sigma^2}{2} \epsilon^\top J_D(x)^\top J_D(x)\epsilon. \end{aligned} \tag{42}$$

Combining equation 41 and equation 42, we obtain

$$\begin{aligned}\mathcal{L} \approx \frac{1}{2}\left\|AD_\theta(x)-y\right\|_2^2 + \frac{\beta}{2}\left\|D_\theta(x)-x\right\|_2^2 \\ + \sigma\left(AD_\theta(x)-y\right)^\top AJ_D(x)\epsilon + \beta\sigma\left(D_\theta(x)-x\right)^\top J_D(x)\epsilon \\ + \frac{\sigma^2}{2}\epsilon^\top J_D(x)^\top A^\top AJ_D(x)\epsilon + \frac{\beta\sigma^2}{2}\epsilon^\top J_D(x)^\top J_D(x)\epsilon.\end{aligned} \tag{43}$$

We now take the expectation with respect to $\epsilon \sim \mathcal{N}(0, I)$. Since

$$\mathbb{E}_\epsilon[\epsilon] = 0,$$

the terms linear in ϵ vanish. Moreover, for any fixed matrix Q,

$$\mathbb{E}_\epsilon\left[\epsilon^\top Q\epsilon\right] = \mathrm{Tr}\left(Q\,\mathbb{E}_\epsilon[\epsilon\epsilon^\top]\right) = \mathrm{Tr}(Q),$$

since $\mathbb{E}_\epsilon[\epsilon\epsilon^\top] = I$. Therefore,

$$\begin{aligned}\mathbb{E}_\epsilon\left[\mathcal{L}\right] \approx \frac{1}{2}\left\|AD_\theta(x)-y\right\|_2^2 + \frac{\beta}{2}\left\|D_\theta(x)-x\right\|_2^2 \\ + \frac{\sigma^2}{2}\,\mathrm{Tr}\left(J_D(x)^\top A^\top AJ_D(x)\right) \\ + \frac{\beta\sigma^2}{2}\,\mathrm{Tr}\left(J_D(x)^\top J_D(x)\right).\end{aligned} \tag{44}$$

Using

$$\mathrm{Tr}\left(J_D(x)^\top A^\top AJ_D(x)\right) = \left\|AJ_D(x)\right\|_F^2$$

and

$$\mathrm{Tr}\left(J_D(x)^\top J_D(x)\right) = \left\|J_D(x)\right\|_F^2,$$

we finally obtain

$$\begin{aligned}\mathbb{E}_\epsilon\left[\mathcal{L}\right] \approx \underbrace{\frac{1}{2}\left\|AD_\theta(x)-y\right\|_2^2 + \frac{\beta}{2}\left\|D_\theta(x)-x\right\|_2^2}_{\text{unperturbed TRACE objective}} \\ + \underbrace{\frac{\sigma^2}{2}\left\|AJ_D(x)\right\|_F^2}_{\text{measurement-space sensitivity}} + \underbrace{\frac{\beta\sigma^2}{2}\left\|J_D(x)\right\|_F^2}_{\text{noise–coupling interaction}}.\end{aligned} \tag{45}$$

The final term appears only when both stochastic perturbation and temporal coupling are present, establishing the interaction stated in Proposition 4.3.

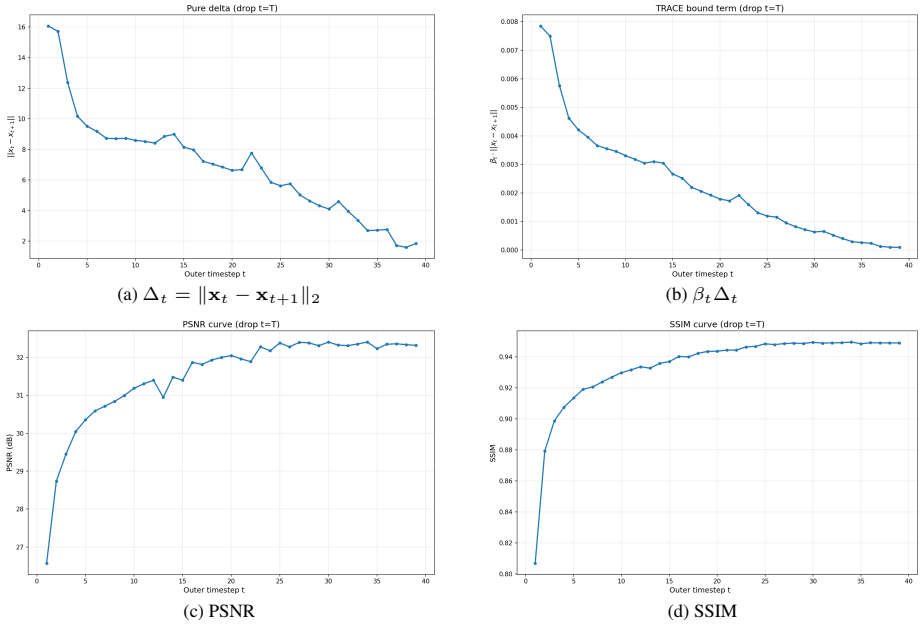
B Empirical Analysis of Stochastic Proximal Consistency

This section examines the empirical behavior of stochastic proximal consistency (SPC). We first evaluate the contribution of the main components of TRACE to the final reconstruction quality. We then focus on the two constituents of SPC, stochastic perturbation and proximal coupling, and study how they affect the reconstruction trajectory separately and jointly. To separate these effects from the neural parameterization, we first consider network-free inverse problems with explicit regularizers and then repeat the analysis under the network-parameterized TRACE formulation. Finally, we examine the sensitivity to the perturbation and proximal coupling strengths.

The purpose of these experiments is complementary to the theoretical analysis in Sec. 4. Proposition 4.1 characterizes the local control provided by the proximal term, while Proposition 4.3 characterizes the additional local regularization that arises when stochastic perturbation and proximal coupling are used together.

Table 4: Component ablation of TRACE on 50% missing-pixel inpainting over three Set14 images. Best and second-best values are shown in **bold** and underlined, respectively.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
TRACE w/o proximal coupling	<u>32.31</u>	<u>0.950</u>	<u>0.0279</u>
TRACE w/o stochastic perturbation	30.59	0.939	0.0395
TRACE w/o weight inheritance	26.01	0.784	0.2166
TRACE	**32.77**	**0.953**	**0.0243**



(a) $\Delta_t = \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2$ (b) $\beta_t \Delta_t$

(c) PSNR (d) SSIM

Figure 3: Representative TRACE trajectory. Both the state variation and its weighted counterpart generally decrease as reconstruction proceeds, while PSNR and SSIM improve and eventually level off. This figure illustrates the behavior of the complete TRACE trajectory; controlled comparisons that isolate the effects of stochastic perturbation and proximal coupling are presented in the next subsection.

B.1 Component Analysis

We first examine the contribution of the main components of TRACE to the final reconstruction quality. We consider three variants obtained by removing (i) proximal coupling, (ii) stochastic perturbation, and (iii) weight inheritance across trajectory states. We evaluate three Set14 images under 50% missing-pixel inpainting.

Table 4 shows that removing any of the three components degrades reconstruction quality. Without proximal coupling, PSNR decreases from 32.77 to 32.31 dB. Removing stochastic perturbation produces a larger drop to 30.59 dB, while removing weight inheritance gives the largest degradation, reaching 26.01 dB. The corresponding SSIM and LPIPS values show the same overall trend. These results indicate that all three design components contribute to the final reconstruction performance.

B.2 Trajectory Behavior and Proximal Coupling

Before separating the individual components of SPC, we first show a representative TRACE trajectory in Fig. 3.

Table 5: Sensitivity to the proximal coupling schedule for super-resolution and deblurring. The first column reports the endpoints of the annealed β_t schedule after scaling by 10^3. Best and second-best values are shown in **bold** and underlined, respectively.

$10^3\beta_t$	Super-Resolution			Deblurring		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
0	27.67	0.891	0.0889	27.26	0.836	0.1757
[1, 0.1]	27.85	0.892	<u>0.0803</u>	27.15	0.832	0.1828
[2, 0.2]	27.68	0.890	0.0826	27.30	0.837	0.1810
[3, 0.3]	27.92	0.896	0.0852	27.19	0.833	0.1755
[4, 0.4]	28.18	**0.898**	**0.0768**	<u>27.31</u>	0.838	**0.1653**
[5, 0.5]	**28.23**	**0.898**	0.0834	**27.49**	**0.846**	<u>0.1686</u>
[6, 0.6]	27.69	0.891	0.0898	27.05	0.828	0.1843
[7, 0.7]	27.99	<u>0.897</u>	0.0832	27.28	<u>0.839</u>	0.1765

The consecutive-state difference Δ_t decreases substantially over the trajectory, although small non-monotone variations remain. The weighted quantity $\beta_t\Delta_t$ shows a similar overall decay. Meanwhile, both PSNR and SSIM improve and gradually saturate as the trajectory progresses.

These observations are consistent with a reconstruction process in which successive states become progressively closer while reconstruction quality continues to improve. However, this single trajectory is descriptive and does not by itself isolate the effect of proximal coupling, since both stochastic perturbation and the coupling schedule vary along the full TRACE trajectory.

This observation raises a related question: how does the strength of the proximal coupling affect the final reconstruction? We therefore vary the β_t schedule while keeping the remaining TRACE configuration fixed.

The parameter β_t determines the strength of the proximal constraint between consecutive reconstruction states. To examine its effect independently of the other TRACE components, we keep the remaining configuration fixed and vary the magnitude of the annealed β_t schedule.

We evaluate three images from ImageNet Large Scale Visual Recognition Challenge (ILSVRC) on super-resolution and deblurring. Table 5 shows that the uncoupled setting ($\beta_t = 0$) is competitive, but several nonzero coupling schedules improve the final reconstruction quality. The schedule $[5, 0.5] \times 10^{-3}$ gives the highest PSNR for both tasks, improving over the uncoupled case by 0.56 dB for super-resolution and 0.23 dB for deblurring. The dependence on the coupling strength is not monotone: increasing β_t beyond the intermediate range does not consistently improve PSNR, SSIM, or LPIPS. Thus, the proximal constraint is beneficial over a range of coupling strengths, while its effect depends on the choice of β_t.

B.3 Trajectory Analysis

TRACE enforces *stochastic proximal consistency* along the reconstruction trajectory through a proximal constraint that keeps the reconstruction generated from each stochastically perturbed intermediate trajectory estimate close to its preceding trajectory state. To understand the mechanism underlying stochastic proximal consistency, we disentangle its stochastic and proximal constituents and study their individual and joint effects.

To address this question, we study stochasticity and proximal consistency across a progression of optimization regimes with different smoothness, convexity, and parameterization properties. Specifically, we first consider network-free formulations, where the image $\mathbf{x}$ is optimized directly under explicit regularization, followed by the full network-parameterized TRACE formulation, where each reconstruction state is represented by an untrained neural network. This progression follows a strategy used in prior work, where a mechanism is first characterized in a controlled and interpretable setting before examining how its behavior changes under higher-dimensional or neural inverse-problem formulations (Krainovic et al., 2023; Fang et al., 2024). This allows us to ask the central question

of our ablation: *which roles of stochasticity and proximal consistency persist across different optimization regimes and which behaviors emerge only after introducing the neural parameterization?*

For each optimization regime, we evaluate four ablation configurations obtained by independently enabling the stochastic and proximal constituents: (i) neither constituent, (ii) stochastic perturbation only, (iii) the proximal constraint only, and (iv) both stochastic perturbation and the proximal constraint. This 2×2 design isolates the individual effects of the two constituents and their interaction.

We characterize these configurations through two complementary aspects of the PSNR trajectory. First, the peak PSNR over the outer iterations measures the best reconstruction quality attained along the trajectory. Second, we examine the peak-aligned evolution of PSNR after the maximum is reached to assess whether the attained reconstruction quality is maintained over subsequent iterations, independently of when the peak occurs. In both classical regularization methods and network-parameterized reconstruction, prior work has shown that reconstruction quality can deteriorate under prolonged optimization, which motivates this second analysis (Chang et al., 2024; Bachmayr & Burger, 2009; Wang et al., 2023a).

For the post-peak analysis, each trajectory is aligned with respect to its own peak. Specifically, letting

$$t_{\mathrm{peak}} = \arg\max_t \mathrm{PSNR}(\mathbf{x}_t), \tag{46}$$

we re-index the subsequent outer iterations by $\tau = t - t_{\mathrm{peak}}$ and measure the degradation relative to the peak as

$$\Delta_{\mathrm{post}}(\tau) = \mathrm{PSNR}(\mathbf{x}_{t_{\mathrm{peak}}}) - \mathrm{PSNR}(\mathbf{x}_{t_{\mathrm{peak}}+\tau}), \qquad \tau \geq 0. \tag{47}$$

Thus, every trajectory starts from $\Delta_{\mathrm{post}}(0) = 0$, allowing post-peak stability to be compared independently of the outer iteration at which the peak is attained.

B.3.1 NETWORK-FREE SETTING.

We first remove the neural parameterization and optimize the image $\mathbf{x}$ directly under an explicit regularizer $R(\mathbf{x})$.To represent the four ablation configurations within a common formulation, let $c_\sigma, c_\beta \in \{0, 1\}$ indicate whether stochastic perturbation and the proximal constraint are active, respectively. At outer iteration t, the inner optimization is initialized as

$$\mathbf{x}_t^0 = \mathbf{x}_{t+1} + c_\sigma \sigma_t \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{48}$$

and we perform K optimization steps on

$$F_t(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda R(\mathbf{x}) + c_\beta \frac{\beta_t}{2}\|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2. \tag{49}$$

The four configurations therefore correspond to $(c_\sigma, c_\beta) = (0, 0), (1, 0), (0, 1)$, and $(1, 1)$, representing plain optimization, perturbation only, coupling only, and the combined setting, respectively.

We evaluate this controlled formulation under three explicit regularizers chosen to vary the structural properties of the resulting optimization problem. Tikhonov regularization yields a smooth convex objective, Total Variation (TV) introduces convex nonsmooth regularization, and the L_p penalty with $p = 0.8$ introduces nonconvexity:

$$R_{\mathrm{Tik}}(\mathbf{x}) = \frac{1}{2}\|D_x\mathbf{x}\|_2^2 + \frac{1}{2}\|D_y\mathbf{x}\|_2^2, \tag{50}$$

$$R_{\mathrm{TV}}(\mathbf{x}) = \|D_x\mathbf{x}\|_1 + \|D_y\mathbf{x}\|_1, \tag{51}$$

$$R_p(\mathbf{x}) = \frac{1}{p} \sum_{d \in \{D_x\mathbf{x}, D_y\mathbf{x}\}} \sum_i |d_i|^p, \qquad p = 0.8, \tag{52}$$

where D_x and D_y denote the horizontal and vertical finite-difference operators. This progression allows us to examine whether the effects of stochasticity and proximal consistency change with the smoothness and convexity of the optimization problem before introducing the nonconvex parameter optimization associated with the neural parameterization. Such distinctions are particularly relevant for perturbation-based and proximal-type optimization mechanisms, whose behavior can depend on the geometry and regularity of the objective (Jin et al., 2017; Davis et al., 2021).

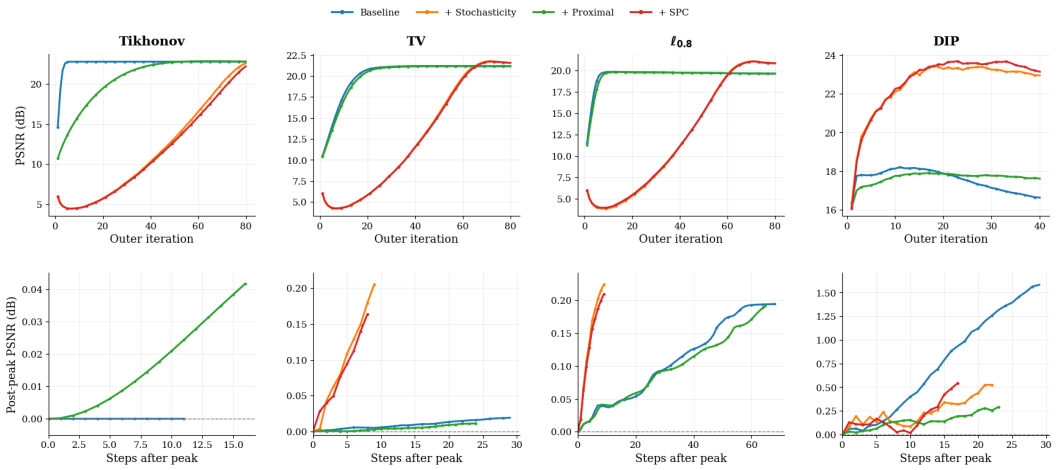


Figure 4: Component ablation of stochasticity and the proximal term for Tikhonov, TV, $\ell_{0.8}$, and DIP reconstruction. Top: PSNR across outer iterations. Bottom: peak-aligned post-peak PSNR degradation. For the network-free experiments, the stochasticity and proximal schedules are $10^{-4} \to 10^{-2}$ and $\beta_t : 10^{-2} \to 10^{-3}$, respectively. For DIP, they are $10^{-4} \to 10^{-2}$ and $\beta_t : 10^{-3} \to 10^{-4}$.

B.3.2 Network-parameterized setting.

We next repeat the ablation under the full network parameterization of TRACE. Using the same indicators $c_\sigma, c_\beta \in \{0, 1\}$, we define the network input at outer iteration t as

$$\mathbf{u}_t = \mathbf{x}_{t+1} + c_\sigma \sigma_t \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{53}$$

and the network parameters are optimized for K iterations using

$$\mathcal{L}_t(\boldsymbol{\theta}_t) = \frac{1}{2} \left\| AD_{\boldsymbol{\theta}_t}(\mathbf{u}_t) - \mathbf{y} \right\|_2^2 + c_\beta \frac{\beta_t}{2} \left\| D_{\boldsymbol{\theta}_t}(\mathbf{u}_t) - \mathbf{x}_{t+1} \right\|_2^2 . \tag{54}$$

As in the network-free setting, $(c_\sigma, c_\beta) = (0, 0), (1, 0), (0, 1)$, and $(1, 1)$ correspond to the plain, stochastic-only, proximal-only, and stochastic-proximal configurations, respectively. The case $(c_\sigma, c_\beta) = (1, 1)$ recovers the full TRACE formulation in Eqs. equation 4 and equation 5, while all remaining optimization and architectural choices are held fixed across the ablation.

B.3.3 Observations

The ablations in Figs. 4–6 reveal distinct roles for stochasticity and the proximal term in shaping the reconstruction trajectory. Across the network-free experiments, stochastic perturbation and temporal coupling exhibit distinct but complementary effects on the reconstruction trajectory. In the smooth convex Tikhonov setting, the different configurations eventually approach similar reconstruction quality, with perturbation and coupling mainly affecting the transient trajectory. Their influence becomes more pronounced under TV and $\ell_{0.8}$ regularization (Fig. 5, TV and $\ell_{0.8}$ rows). Perturbation can lead to higher peak PSNR, but the improvement is generally accompanied by greater post-peak degradation, whereas temporal coupling has a comparatively small effect on the attainable peak while reducing post-peak variation.

The joint sweeps in Fig. 6 further show that temporal coupling can modify this behavior. For the $10^{-6} \to 10^{-4}$ and $10^{-5} \to 10^{-3}$ noise schedules in the DIP row of Fig. 6, appropriate coupling strengths further reduce post-peak degradation relative to stochasticity alone and can also attain higher peak PSNR than both the baseline and stochasticity-only trajectories. This ordering is not uniform across perturbation strengths. At the $10^{-4} \to 10^{-2}$ noise schedule (rightmost column of the DIP block in Fig. 6), stochasticity alone can be more stable than some of the proximally coupled trajectories. The strong dependence on the relative perturbation and coupling strengths highlights the role of stochastic proximal consistency in shaping the reconstruction trajectory under the neural parameterization, while a rigorous characterization of this interaction remains an interesting direction for future work.

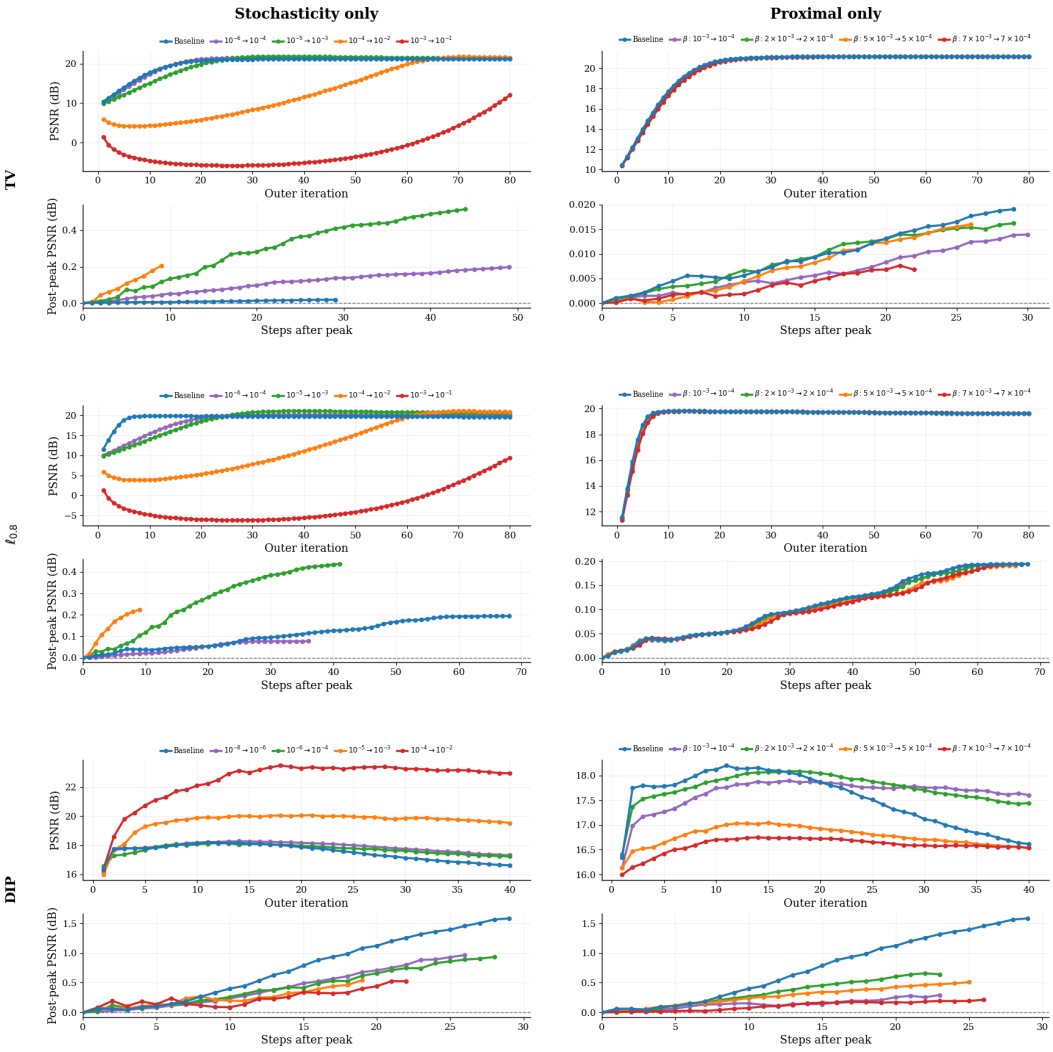


Figure 5: Sensitivity to stochasticity and the proximal term in isolation for TV, $\ell_{0.8}$, and DIP reconstruction. The left column varies c_α with $c_\beta = 0$, while the right column varies c_β with $c_\alpha = 0$. In each setting, the top panel shows PSNR trajectories and the bottom panel shows peak-aligned post-peak degradation.

The network-parameterized DIP experiments reveal a stronger and more nonlinear interaction between the two mechanisms. Increasing the perturbation strength produces substantially higher peak PSNR (Fig. 5, DIP row, stochasticity-only column), while, unlike the network-free experiments, the stronger perturbation levels also exhibit less post-peak degradation than the baseline trajectory. Thus, stochastic perturbation in the neural setting is associated not only with reaching higher-quality reconstructions but also with improved trajectory stability. This contrast is consistent with our local analysis, where input perturbation induces a sensitivity penalty proportional to $\sigma_t^2 \|A J_{D_{\boldsymbol{\theta}_t}}\|_F^2$, and suggests that its interaction with the neural parameterization may contribute to the observed stability.

The joint sweeps in Fig. 6 further show that temporal coupling can modify this behavior. For several perturbation schedules, appropriate coupling strengths further reduce post-peak degradation relative to perturbation alone and can also attain higher peak PSNR than both the baseline and perturbation-only trajectories (see the DIP row of Fig. 6 across the intermediate perturbation schedules). This ordering is not uniform at the strongest perturbation level: for the $10^{-4} \to 10^{-2}$ noise schedule (rightmost column of the DIP block in Fig. 6), perturbation alone can be more stable than some of the coupled trajectories. The strong dependence on the relative perturbation and coupling strengths highlights the role of stochastic proximal consistency in shaping the reconstruction trajectory under

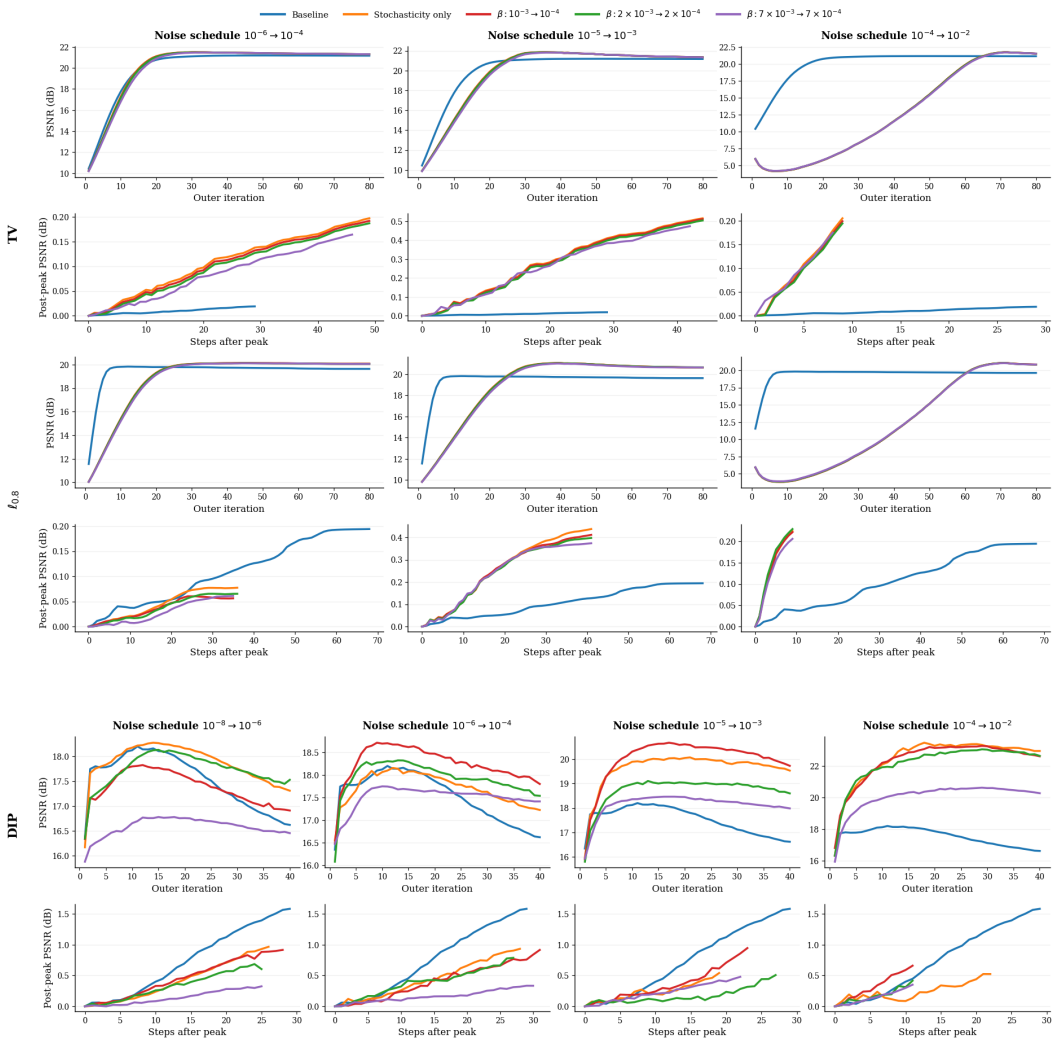


Figure 6: Joint sensitivity to stochasticity and the proximal term for TV, $\ell_{0.8}$, and DIP reconstruction. Each column fixes c_α and compares stochasticity alone ($c_\beta = 0$) with several choices of $c_\beta > 0$, together with the baseline. The top panels show PSNR trajectories and the bottom panels show peak-aligned post-peak degradation. The DIP results are shown separately because they use a different range of c_α.

the neural parameterization, while a rigorous characterization of this interaction remains an interesting direction for future work.

C Empirical Analysis of Stochastic Proximal Consistency

Proposition 4.3 shows that SPC introduces an additional Jacobian-dependent regularization whose strength is jointly controlled by β and σ^2. We next examine how this regularization is reflected in the reconstruction dynamics through two questions. First, does SPC regularize reconstruction components that lie in the null space of A? Second, does SPC alter the spectral bias of the untrained network?

C.1 Null-Space Regularization

We empirically examine whether the predicted null-space regularization is reflected in the reconstruction trajectory. We decompose each reconstruction into its components in $\mathrm{range}(A^\top)$ and $\mathrm{null}(A)$, and track the reconstruction accuracy in the full image space and in each subspace separately. This allows us to examine how SPC affects reconstruction errors that are invisible to the measurement operator.

We consider image inpainting, for which the range–null decomposition can be computed exactly. Let $A(x) = M \odot x$, where M is the binary sampling mask. Since A is an orthogonal projection in this setting, $P_{\mathcal{R}} = M$ and $P_{\mathcal{N}} = I - M$ project onto the observed and missing image components, respectively. At each outer trajectory step, we compute the PSNR of the complete reconstruction, the observed component $P_{\mathcal{R}}\hat{x}$, and the missing component $P_{\mathcal{N}}\hat{x}$ with respect to the corresponding components of the ground-truth image. We compare the four configurations corresponding to fixed-input DIP, trajectory-only reconstruction, stochastic perturbation without SPC, and the full SPC formulation.

As shown in Fig. 7, SPC provides little change in the observed component but consistently improves null-space PSNR. The gain in full-image PSNR is therefore primarily driven by more accurate reconstruction of the measurement-invisible component.

The null-space reconstruction error, however, does not directly measure the Jacobian sensitivity appearing in Proposition 4.3. We therefore approximately estimate $\|P_{\mathcal{N}} J_D(x)\|_F^2$ along the reconstruction trajectory. Following Monte Carlo finite-difference Jacobian estimation (Ramani et al., 2008), we approximate Jacobian-vector products along Gaussian random directions using central finite differences. For $v \sim \mathcal{N}(0, I)$,

$$P_{\mathcal{N}} J_D(x) v \approx P_{\mathcal{N}} \frac{D_\theta(x + hv) - D_\theta(x - hv)}{2h}.$$

Since $\mathbb{E}[vv^\top] = I$,

$$\mathbb{E}_v \left[\|P_{\mathcal{N}} J_D(x) v\|_2^2 \right] = \|P_{\mathcal{N}} J_D(x)\|_F^2.$$

We therefore use the Monte Carlo estimator

$$\widehat{\mathcal{S}}_{\mathcal{N}}(x) = \frac{1}{K} \sum_{k=1}^{K} \left\| P_{\mathcal{N}} \frac{D_\theta(x + hv_k) - D_\theta(x - hv_k)}{2h} \right\|_2^2,$$

where $v_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$. In our experiments, we use $K = 32$ Gaussian random vectors and set $h = 10^{-3}$. At each selected outer trajectory step, we fix the network parameters after the inner optimization and evaluate the finite-difference approximation around the corresponding unperturbed trajectory state x_{prev}. The same Gaussian random vectors are used for the perturbation-only and SPC variants to enable a paired comparison. Due to the computational cost of the finite-difference estimation, we evaluate the Jacobian sensitivity at $t \in \{10, 20, 40\}$.

As shown in Fig. 8, the estimated null-space Jacobian sensitivity is consistently lower with SPC than with stochastic perturbation alone, while the range-space sensitivity remains comparable. This is consistent with Proposition 4.3, as SPC reduces Jacobian sensitivity primarily in the measurement-invisible subspace. Together with the improved null-space PSNR, these results indicate that the additional regularization acts primarily on components that are not constrained by the measurements.

C.2 Spectral Bias Modulation

Deep image priors are known to exhibit a spectral bias, fitting low-frequency image components earlier than high-frequency components. Following Shi et al. (2021), we use frequency-band corre-

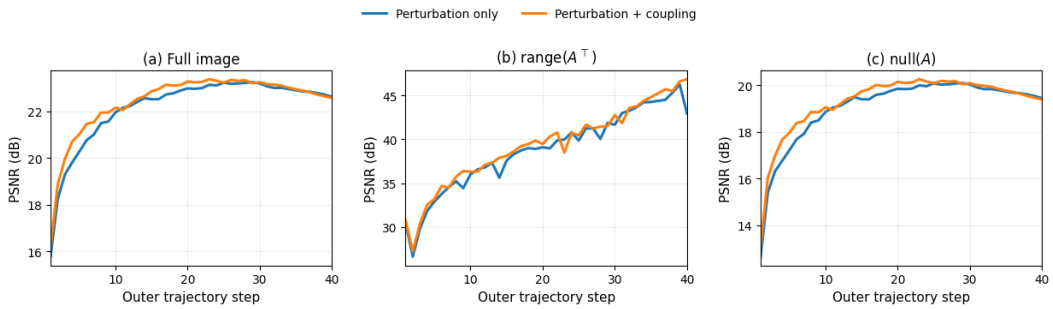


Figure 7: Reconstruction accuracy across the TRACE trajectory for image inpainting. Stochasticity alone is compared with stochasticity combined with the proximal term. PSNR is shown for the full reconstruction and for its components in $\mathrm{range}(A^\top)$ and $\mathrm{null}(A)$. []

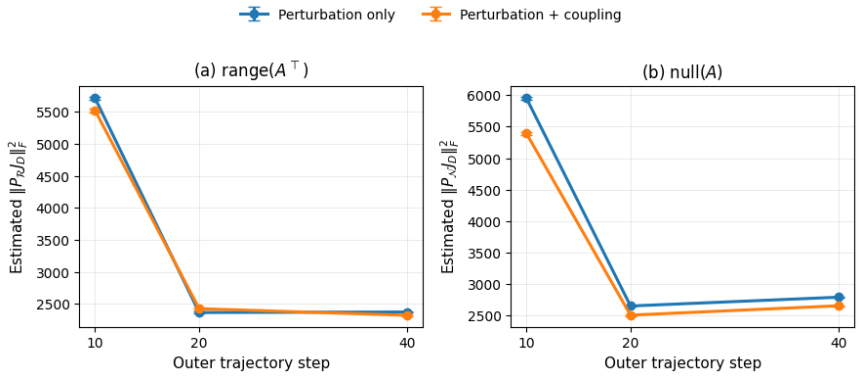


Figure 8: Estimated Jacobian sensitivity along the TRACE trajectory for image inpainting. We compare stochastic perturbation alone with stochastic perturbation combined with temporal coupling. The range-space and null-space Jacobian energies are estimated using the Monte Carlo finite-difference procedure described in Sec. C.1. Temporal coupling reduces the estimated null-space Jacobian energy at all sampled trajectory steps.

spondence (FBC) to examine whether SPC modulates this spectral progression. For a reconstruction $\hat{x}_t$ and ground-truth image $x^\star$, the Fourier-domain correspondence is written as

$$H_t = \frac{\mathcal{F}\{\hat{x}_t\}}{\mathcal{F}\{x^\star\}},$$

where $\mathcal{F}$ denotes the Fourier transform. The correspondence map is divided into non-overlapping frequency bands, and the mean correspondence within each band defines its FBC value $C_n(t)$. We use five bands, with C_1 and C_5 denoting the lowest- and highest-frequency bands, respectively.

Figure 9 compares the FBC trajectories of four variants trained for the same number of optimization steps. Standard DIP shows the expected low-to-high frequency ordering, but the bands quickly approach similar correspondence levels. Sequentially feeding the previous reconstruction as the network input prolongs this separation, although the frequency bands again converge relatively early. At each outer step, stochastic perturbation lowers the frequency correspondence. The inner optimization then progressively restores it, resulting in the repeated sawtooth structure in the FBC trajectories. The perturbation-only and SPC trajectories exhibit the same overall pattern, indicating that stochastic perturbation is primarily responsible for the repeated frequency-dependent reset and recovery.

The perturbation-only and SPC trajectories are qualitatively similar in Fig. 9, making the effect of SPC difficult to distinguish from the full FBC trajectories. The "Low and high frequency fitting" plot in Fig. 10 therefore focuses on C_1 and C_5. To quantify their separation, the "Band 1 - Band 5"

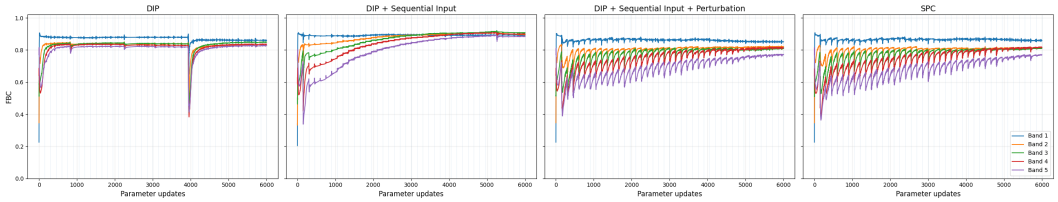


Figure 9: Frequency-band correspondence (FBC) over 6000 optimization updates for four reconstruction variants. Standard DIP uses a fixed network input, while the second variant sequentially feeds the previous reconstruction as the input to the next outer step. Adding stochastic perturbation produces repeated frequency-dependent drops and recovery across outer steps, with larger changes at higher frequencies. SPC exhibits the same overall stochastic spectral dynamics while introducing a more subtle modulation of the frequency progression.

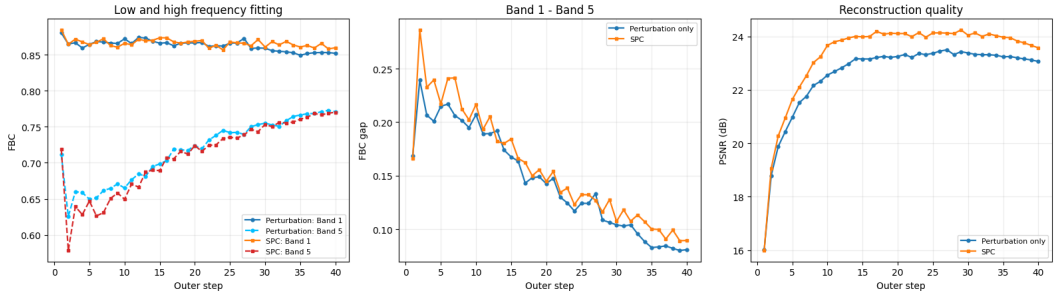


Figure 10: Comparison of spectral progression and reconstruction quality for stochastic perturbation alone and SPC. The "Low and high frequency fitting" plot compares the lowest- and highest-frequency FBC trajectories. The "Band 1 - Band 5" plot reports the low–high FBC gap, $C_1 - C_5$. The "Reconstruction quality" plot compares PSNR across the outer trajectory.

plot reports

$$G(t) = C_1(t) - C_5(t).$$

SPC maintains a slightly larger FBC gap than stochastic perturbation alone over much of the trajectory, indicating a stronger separation between low- and high-frequency fitting.

Importantly, a larger FBC gap is not itself a measure of reconstruction quality. Shi et al. (2021) observed that DIP can attain its best reconstruction quality while low-frequency components are already well matched and high-frequency components remain less fully incorporated, and showed that controlling this spectral progression can mitigate subsequent performance degradation. In our setting, the increased low-high separation with SPC coincides with improved reconstruction PSNR, as shown in the "Reconstruction quality" plot of Fig. 10.

Together, these observations suggest that the additional Jacobian regularization induced by SPC modulates the spectral bias of the untrained network, rather than uniformly changing the fitting rate across frequencies.

D Algorithmic Details

This section summarizes the evaluation protocol and the implementation settings of TRACE and all compared methods. Details specific to CT reconstruction are given in Appendix E, and details of the SPC experiments on diffusion solvers in Appendix H.

Evaluation protocol. All methods are run on the same measurements, generated by the same forward operators. The natural-image benchmark uses 14 images (10 from BSDS68 and 4 from Set14), resized to 256×256, and six tasks: random inpainting with 50% and 70% of the pixels missing (a random pixel-wise mask shared by the three color channels); super-resolution by $\times 2$ and $\times 4$ average-pooling downsampling; motion deblurring with a 21×21 linear motion kernel at

$45°$; and for nonlinear deblurring, we follow DPS (Chung et al., 2023a) and use the BKSE forward model (Tran et al., 2021): the measurement is generated as $y = \mathcal{F}_\phi(x, k) + n$, where $\mathcal{F}_\phi$ is a pretrained blur network and k is a randomly sampled latent blur kernel, with Gaussian noise of standard deviation 0.05. The CT benchmark (24 slices, sparse-view and limited-angle) is described in Appendix E.1. We report PSNR, SSIM and LPIPS for all tasks, computed on images in $[0, 1]$ at 256×256 resolution; for CT, LPIPS is computed on the grayscale slice replicated to three channels.

Reported output. TRACE, SDI, DIP and all diffusion-based and CNN-based methods report their final output; DIP reports the iterate with the lowest data-fidelity loss, which does not use the ground truth. For ASeqDIP, Self-Guided DIP, UGoDIT and uDiG-DIP we use the evaluation mode of their implementations, which returns the iterate with the highest PSNR with respect to the ground truth.

Hardware and software. All experiments are implemented in PyTorch and run on NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs. Random seeds are fixed for all methods.

D.1 TRACE

For all natural-image tasks, TRACE uses $T = 40$ outer optimization steps and $K = 150$ inner Adam iterations per step, with a learning rate of 10^{-3}. The first outer step, indexed by $t = T$, uses the uncoupled DIP objective for K iterations to obtain the initialization x_T. The remaining $T - 1$ outer steps use the SPC objective, with the first SPC step starting at $t = T - 1$. Thus, the complete reconstruction performs

$$K + (T - 1)K = TK = 6000$$

Adam updates. The network weights are inherited from one outer step to the next.

The network is a randomly initialized U-Net with five downsampling levels (64 to 1024 channels), bilinear upsampling, and approximately 88M parameters.

The stochastic perturbation level σ_t follows a DDPM-style noise schedule. Specifically,

$$\sigma_t = \sqrt{1 - \bar{\alpha}_t}, \qquad \bar{\alpha}_t = \prod_{i=0}^{t} (1 - v_i),$$

where the noise-variance parameter v_i increases linearly from 10^{-4} at $i = 0$ to 10^{-2} at $i = T - 1$. After the DIP initialization at $t = T$, the SPC trajectory begins with σ_{T-1} and follows the reverse noise schedule toward the cleanest noise level. For $T = 40$, the noise level decreases from approximately 0.43 to 0.01. We use v_i exclusively for the DDPM-style noise-variance schedule and reserve β_t for the consistency weight.

The consistency weight is defined as

$$\beta_t = \beta_{\min} + \frac{t}{T - 1} \left(\beta_{\max} - \beta_{\min} \right).$$

Consequently, the consistency weight decreases along the SPC trajectory from $\beta_{T-1} = \beta_{\max}$ at the first SPC step to $\beta_0 = \beta_{\min}$ at the final SPC step.

For the main natural-image experiments, we use

$$(\beta_{\max}, \beta_{\min}) = (5 \times 10^{-4}, 5 \times 10^{-5})$$

for inpainting and super-resolution, and

$$(\beta_{\max}, \beta_{\min}) = (5 \times 10^{-2}, 5 \times 10^{-3})$$

for the two deblurring tasks.

We use random seed 42 and report the output of the final outer step. The CT settings are provided in Appendix E.2. Unless explicitly varied, all ablations use the same implementation and hyperparameters.

D.2 COMPARED METHODS

All baselines are run with the authors' released code, or with the reference implementation of a public library where stated, using their default hyperparameters unless noted below. For the neural-prior methods, T denotes the number of outer stages and K the number of inner optimization steps per stage.

DIP (Ulyanov et al., 2018). We use the standard skip-connection encoder–decoder of DIP (5 scales, 128 channels), a 32-channel random input with input-noise regularization of standard deviation $1/30$, and 3000 Adam iterations with learning rate 10^{-2}.

ASeqDIP (Alkhouri et al., 2024). For natural images we use $T = 600$ stages with $K = 2$ inner steps, learning rate 3×10^{-4} and autoencoding-regularization weight $\lambda = 1$. For CT we use $T = 600$ stages, learning rate 2×10^{-4} and $\lambda = 1$.

Self-Guided DIP (Liang et al., 2025c). We run 1200 iterations with learning rate 3×10^{-4} for the network and 10^{-1} for the reference input, 3 smoothing samples and an exponential moving average with weight 0.99. The guidance weight is $\alpha = 2$ for natural images.

SDI (Luo et al., 2026). SDI uses the same U-Net, $T = 40$ trajectory steps, $K = 150$ inner Adam iterations per step with learning rate 10^{-3}, and the same DDPM-style noise-variance schedule $\{v_i\}_{i=0}^{T-1}$ as TRACE.

UGoDIT (Liang et al., 2025a). UGoDIT uses $T = 300$ stages with $K = 2$ inner steps and learning rate 10^{-2}. Its encoder is initialized with the transferable weights learned by unsupervised group training and kept frozen, and only the decoder is optimized.

uDiG-DIP (Liang et al., 2025b). uDiG-DIP is evaluated on CT only, with its released CT diffusion prior (the same prior as DDS), $T = 300$ stages, learning rate 2×10^{-4} and $\lambda = 0$.

SITCOM (Alkhouri et al., 2025). We use the released implementation with 20 sampling steps, 30 inner optimization steps per sampling step and learning rate 0.02. The diffusion prior is the unconditional natural-image pre-trained model released with SITCOM, for both natural images and CT.

DPS, DAPS, RED-diff and PnP-DM. These four solvers are run with the implementations and default configurations of InverseBench (Zheng et al., 2025), all with the same unconditional natural-image diffusion prior: DPS with 1000 steps and guidance scale 10; DAPS with 200 annealing steps, 5 diffusion steps per annealing step and 100 Langevin steps (learning rate 10^{-4}, $\tau = 0.01$); RED-diff with 1000 steps, learning rate 0.02 and regularization weight 0.25; and PnP-DM with 200 annealing steps and 100 Langevin steps (learning rate 10^{-4}, $\tau = 0.05$). The forward operators and measurements are the same as for all other methods, and PSNR and SSIM are those computed by the InverseBench image evaluator.

PnP-DnCNN and PnP-FFDNet. Both use the plug-and-play proximal-gradient solver of the DeepInverse library with the released color DnCNN and FFDNet denoisers, denoiser noise level 0.01, step size 1 (with the forward operator normalized to unit spectral norm), and at most 200 iterations with early stopping.

IRCNN, RED-CNN, EDCNN and DDS. These methods are evaluated on CT only; their settings are given in Appendix E.3.

E Additional Details and Results for CT Reconstruction

E.1 Data and forward operators

We evaluate CT reconstruction in two challenging tomographic settings, sparse-view and limited-angle, using full-dose abdominal CT scans from the AAPM Low-Dose CT Grand Challenge dataset McCollough et al. (2017) (1 mm slice thickness, B30 reconstruction kernel). The test set consists of 24 slices: twelve consecutive slices from each of two patients (L067 and L109). Each slice is normalized to $[0, 1]$ by per-slice min–max scaling of the stored DICOM values and resized to 256×256 by bilinear interpolation. Both tasks use a 2D parallel-beam Radon transform applied to the image masked by its inscribed circle, and the sinogram is divided by the maximum of the ground-truth sinogram. Sparse-view CT uses 60 projection angles uniformly spaced over $[0^\circ, 180^\circ)$, which preserves the full angular coverage but severely reduces the number of measurements. Limited-angle CT uses 120 contiguous angles from 0° to 119° in 1° steps, so that the remaining 60° of the angular range is never observed, which leads to missing-angle artifacts and strong directional ambiguity. Measurements are noiseless. PSNR and SSIM are computed on the $[0, 1]$ single-channel reconstruction with respect to the normalized ground truth. LPIPS uses the AlexNet backbone, with the grayscale image replicated to three channels and scaled to $[-1, 1]$. All metrics in Table 1 are averaged over the 24 test slices.

E.2 Implementation of TRACE

TRACE uses the same U-Net for both tasks, with five downsampling levels (from 64 to 1024 channels), bilinear upsampling, a single input and output channel, and approximately 88M parameters. The network is randomly initialized and optimized with Adam for $K = 150$ inner iterations per outer step. As in Appendix D.1, the first outer step, indexed by $t = T$, uses the uncoupled DIP objective to obtain the initialization x_T. The subsequent trajectory starts at $t = T - 1$ and uses the SPC objective. The network weights are inherited from one outer step to the next, and the complete reconstruction uses TK Adam updates.

The injected noise follows the same DDPM-style noise-variance schedule as in Appendix D.1, with v_i increasing linearly from 10^{-4} at $i = 0$ to 10^{-2} at $i = T - 1$. After the DIP initialization, the SPC trajectory begins with σ_{T-1} and proceeds toward the cleanest noise level.

The SPC weight β_t decreases along the SPC trajectory, with $\beta_{T-1} = \beta_{\max}$ at the first SPC step and $\beta_0 = \beta_{\min}$ at the end of the trajectory. For sparse-view CT, we use $T = 300$, a learning rate of 10^{-3}, and

$$(\beta_{\max}, \beta_{\min}) = (2 \times 10^{-3}, 2 \times 10^{-4}).$$

For limited-angle CT, we use $T = 80$, a learning rate of 5×10^{-4}, and

$$(\beta_{\max}, \beta_{\min}) = (5 \times 10^{-4}, 5 \times 10^{-5}).$$

No training data or pretrained weights are used, and one reconstruction is produced per slice.

E.3 Implementation of the compared methods

All methods are evaluated on the same 24 slices with the same projection angles, and all reconstructions are mapped to the same $[0, 1]$ normalization before computing metrics.

- **Neural-prior methods** (DIP, ASeqDIP, Self-Guided DIP, SDI) are run with their released code, using the same forward operators and measurements as TRACE.
- **IRCNN** is a plug-and-play method whose CNN denoiser is pretrained on natural images; it is used without retraining on CT data.
- **RED-CNN** and **EDCNN** are supervised networks that map the filtered back-projection (FBP) of the measurements to the full-dose image. A separate model is trained for each task on slices from four AAPM patients (L143, L192, L286 and L310) that are disjoint from the test patients, with the measurements simulated by the same forward operators as above.

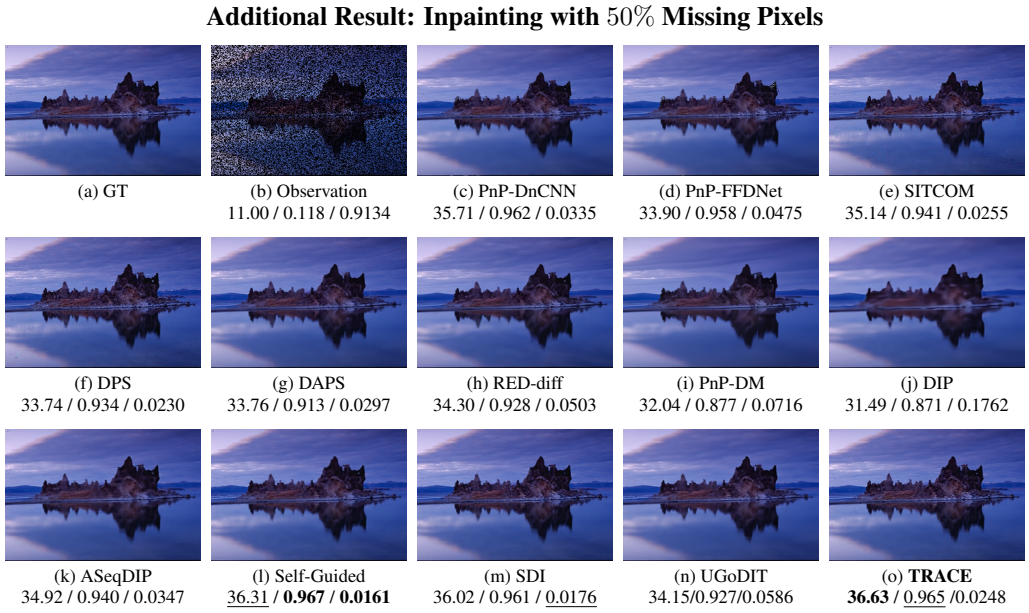


Figure 11: Additional qualitative result for image inpainting with 50% missing pixels. The numbers below each image denote PSNR↑ / SSIM↑ / LPIPS↓; the best and second-best values are shown in **bold** and underlined, respectively. TRACE achieves the highest PSNR on this example and produces a visually consistent reconstruction in the missing regions.

- **DDS** is run with its released code and CT diffusion prior, pretrained on AAPM CT images, using the sampling settings of its released sparse-view evaluation script and changing only the number and range of projection angles to match our tasks. Since this prior operates on a fixed intensity window, each slice is mapped into that window before reconstruction and the result is mapped back to the per-slice $[0, 1]$ normalization for evaluation.
- **uDiG-DIP** is run with its released code and the same pretrained CT diffusion prior as DDS.
- **SITCOM** uses a diffusion model pretrained on natural images. The grayscale slice is replicated to three channels as input, and the first output channel is used for evaluation.

E.4 Additional qualitative results

Figures 16 and 17 show one additional slice for sparse-view and limited-angle CT, respectively, complementing the example in Figure 1. As in the main text, the out-of-domain SITCOM prior leads to background texture artifacts in both tasks and to structural distortions in the limited-angle case. On the sparse-view slice in Figure 16, TRACE attains the best PSNR, SSIM and LPIPS, with IRCNN second in PSNR and SSIM and EDCNN second in LPIPS. On the limited-angle slice in Figure 17, TRACE attains the best PSNR and SSIM and the second-best LPIPS, while Self-Guided DIP, ASeqDIP and DDS attain the second-best PSNR, the second-best SSIM and the best LPIPS, respectively.

F Additional Details and Results for Natural Image

In this section, we provide additional qualitative results to complement the main experiments. These examples further illustrate the behavior of TRACE across different natural-image inverse problems, including image inpainting, super-resolution, and deblurring. For all visual comparisons, the numbers below each image denote PSNR↑ / SSIM↑ / LPIPS↓.

Additional Result: Inpainting with 70% Missing Pixels

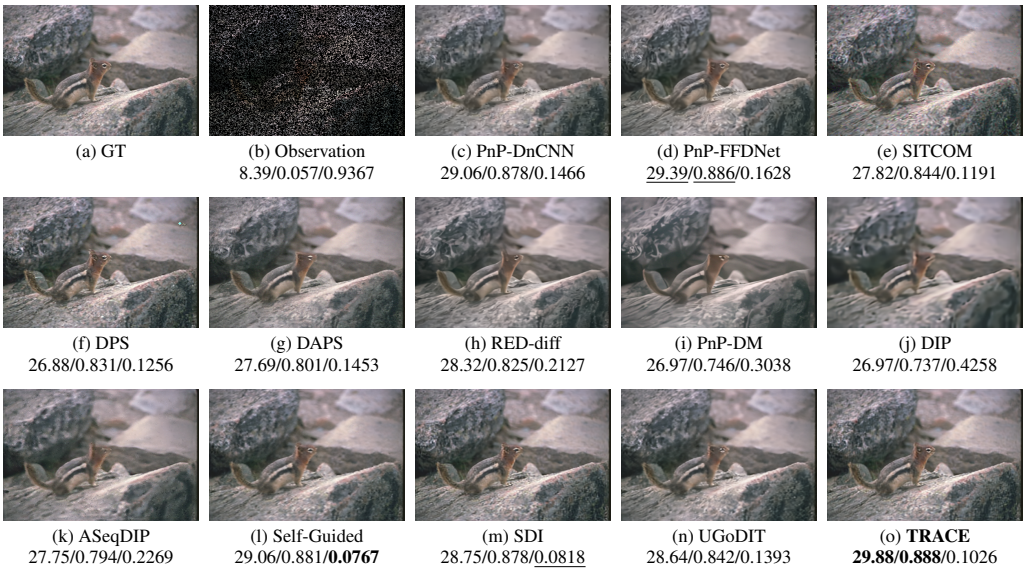


Figure 12: Representative image inpainting result under 70% missing pixels. The numbers below each image denote PSNR↑ / SSIM↑ / LPIPS↓; the best and second-best values are shown in **bold** and underlined, respectively. TRACE achieves the highest PSNR and SSIM on this challenging example, indicating improved pixel-level and structural fidelity.

F.1 Image Inpainting

We provide an additional qualitative comparison for image inpainting with 50% random masking. Inpainting removes a large portion of pixels and therefore requires the reconstruction method to infer missing structures from incomplete observations. This provides a complementary test to deblurring and super-resolution, where the measurements are corrupted but spatially complete.

Figures 11 and 12 show that TRACE produces a visually consistent reconstruction in the missing regions. Compared with DIP and ASeqDIP, TRACE reduces artifacts caused by directly fitting the incomplete observation. Compared with stronger baselines such as Self-Guided DIP and SDI, TRACE achieves the highest PSNR on these examples while maintaining competitive structural and perceptual quality.

F.2 Image Super-Resolution

We further evaluate image super-resolution under $\times 2$ and $\times 4$ downsampling setting. Super-resolution requires recovering high-frequency details from a low-resolution observation, making it sensitive to over-smoothing and unstable texture generation.

Figures 13 and 14 show additional SR examples. TRACE achieves the best PSNR and LPIPS among the compared methods and matches the best SSIM. This indicates that the proposed trajectory-constrained formulation improves pixel-level fidelity while preserving perceptual quality. The result further supports the conclusion in the main paper that TRACE is effective for recovering fine image structures under resolution loss.

F.3 Image Deblurring

We provide an additional qualitative comparison for motion deblurring. Deblurring is highly ill-conditioned because high-frequency information is attenuated by the blur operator, and iterative reconstruction methods can easily introduce ringing or texture artifacts.

Additional Result: Super-resolution $\times 2$



Figure 13: Additional qualitative result for image super-resolution under $\times 2$ downsampling. The numbers below each image denote PSNR↑ / SSIM↑ / LPIPS↓; the best and second-best values are shown in **bold** and underlined, respectively. TRACE achieves the best PSNR and LPIPS and matches the second-best SSIM on this example.

Additional Result: Super-resolution $\times 4$



Figure 14: Additional qualitative result for image super-resolution under $\times 4$ downsampling. The numbers below each image denote PSNR↑ / SSIM↑ / LPIPS↓; the best and second-best values are shown in **bold** and underlined, respectively. TRACE achieves the highest PSNR and SSIM on this example.

For both deblurring settings, we synthesize the observation as $y = A(x_{\text{true}})$ without adding extra noise, where x_{true} is the resized RGB image (default 256×256). The blur operator is implemented as channel-wise (depthwise) convolution with shared kernels across three channels and symmetric padding ($p = 10$, kernel size 21×21), outputs are clipped to $[0, 1]$.

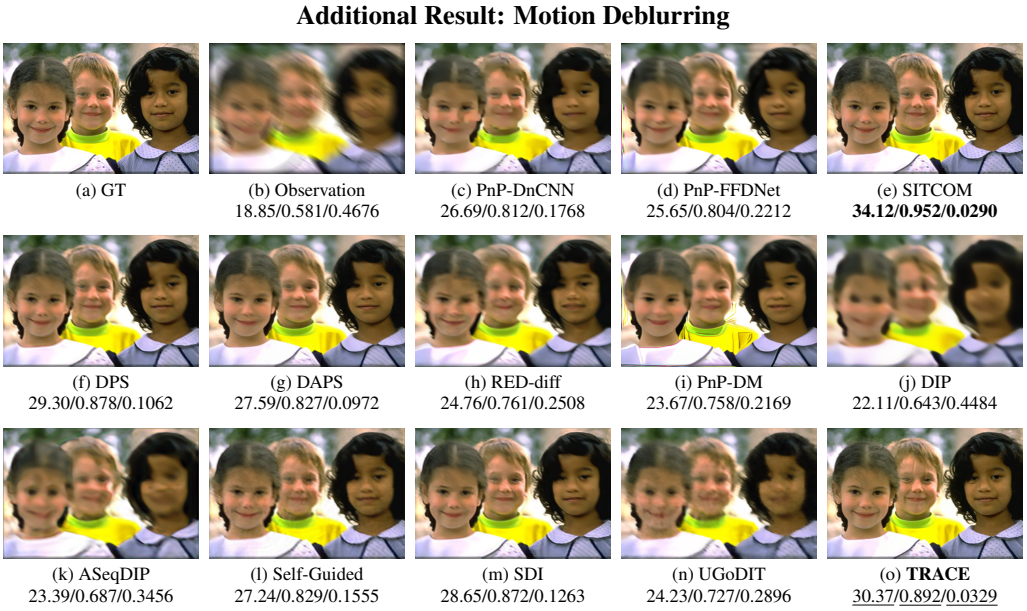


Figure 15: Additional qualitative result for motion deblurring. The numbers below each image denote PSNR↑ / SSIM↑ / LPIPS↓; the best and second-best values are shown in **bold** and underlined, respectively. TRACE achieves the second-best PSNR, SSIM, and LPIPS on this example.

Motion blur. We use a spatially invariant linear motion kernel of length 21 and direction $45°$. The kernel is constructed by drawing a line through the kernel center along the motion direction and normalizing it to unit sum.

Nonlinear blur. For nonlinear deblurring, we follow DPS (Chung et al., 2023a) and use the BKSE forward model (Tran et al., 2021): the measurement is generated as $y = \mathcal{F}_\phi(x, k) + n$, where $\mathcal{F}_\phi$ is a blur network pretrained on the GoPro dataset and k is a randomly sampled latent blur kernel, fixed across all test images, with additive Gaussian noise of standard deviation 0.05.

Figure 15 shows that TRACE achieves the second-best PSNR, SSIM, and LPIPS on the displayed example. Compared with DIP and ASeqDIP, TRACE recovers sharper structures and avoids severe blur-related artifacts. Compared with Self-Guided DIP, UGoDIT, and SDI, TRACE further improves both distortion-based metrics and perceptual quality. These additional results are consistent with the main-paper observation that trajectory regularization improves stability and reconstruction reliability in ill-conditioned inverse problems.

G Additional Ablations and Computational Analysis

This section provides additional analyses of practical design choices and computational cost in TRACE. We first study how a fixed optimization budget is allocated between the number of outer trajectory states and the number of inner network updates. We then examine the effect of using a pretrained network initialization instead of the random initialization adopted in the main experiments. Finally, we report runtime, memory usage, and network-evaluation cost for TRACE and the compared methods.

G.1 Outer and Inner Optimization Budget

TRACE contains two optimization scales: the number of outer trajectory states T and the number of inner network updates K performed at each state. To study their relative allocation independently of the total number of network updates, we fix

$$N = TK = 6000$$

and vary T and K accordingly.

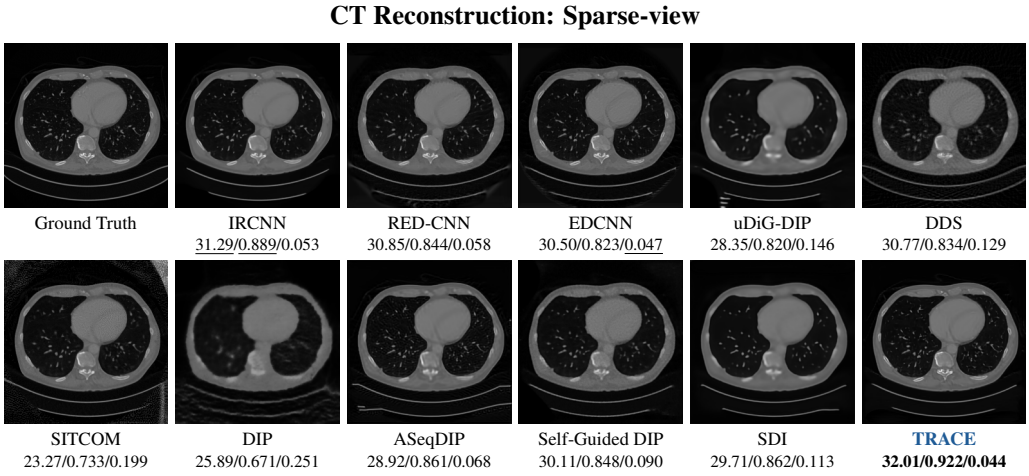


Figure 16: Representative sparse-view CT reconstructions (PSNR↑ / SSIM↑ / LPIPS↓); the best and second-best values are shown in **bold** and underlined, respectively. TRACE preserves anatomical structures more clearly and achieves the best PSNR, SSIM, and LPIPS in this example.

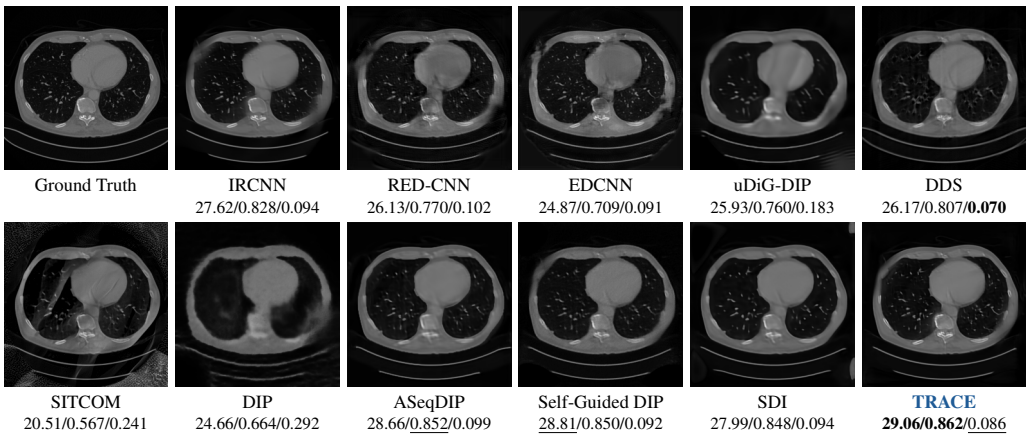


Figure 17: Additional qualitative results for limited-angle CT reconstruction (PSNR↑ / SSIM↑ / LPIPS↓); the best and second-best values are shown in **bold** and underlined, respectively. TRACE achieves the highest PSNR and SSIM and the second-best LPIPS in this example, showing improved robustness to missing angular measurements and reduced limited-angle artifacts.

We evaluate three Set14 images under 50% missing-pixel inpainting. Table 6 shows that the reconstruction quality depends on how the fixed optimization budget is distributed between the outer and inner loops. Increasing T from 10 to 40 generally improves PSNR and SSIM, with the best PSNR and SSIM obtained at $T = 40$ and $K = 150$. Further increasing T to 50 or 60, and consequently reducing the number of inner updates per state, does not provide additional gains. These results motivate the default choice $T = 40$ and $K = 150$ used in the main experiments.

G.2 Effect of Pretrained Initialization

The main TRACE experiments use random initialization and therefore do not require a pretrained network. As an additional experiment, we examine whether the same trajectory formulation can also be used when a pretrained initialization is available.

Specifically, we initialize the network parameters using the publicly released unconditional 256×256 Guided Diffusion checkpoint (`256x256_diffusion_uncond.pt`) from OpenAI (OpenAI, 2021), and then optimize the network using the same TRACE procedure. The pretrained model is

Table 6: Sensitivity to the allocation of a fixed optimization budget between the number of outer trajectory states T and inner updates K. The total number of network updates is fixed at $N = TK = 6000$. Best and second-best values are shown in **bold** and underlined, respectively.

T	K	PSNR ↑	SSIM ↑	LPIPS ↓
10	600	31.65	0.946	0.0286
20	300	32.26	<u>0.951</u>	**0.0252**
30	200	31.92	0.946	0.0315
40	150	**32.65**	**0.952**	<u>0.0259</u>
50	120	<u>32.43</u>	0.948	0.0273
60	100	32.21	0.948	0.0323

Table 7: Effect of network initialization on 50% missing-pixel inpainting. The pretrained model is used only for initialization before TRACE optimization and is not used as a fixed reconstruction prior.

Initialization	PSNR ↑	SSIM ↑	LPIPS ↓
Random (default)	32.59	0.949	**0.0279**
Pretrained	**32.75**	**0.952**	0.0301

used only to initialize the parameters; it is not kept fixed as a reconstruction prior during optimization.

Table 7 reports the average results on three Set14 images under 50% missing-pixel inpainting. Pretrained initialization increases PSNR from 32.59 to 32.75 dB and SSIM from 0.949 to 0.952, while LPIPS changes from 0.0279 to 0.0301. The differences are modest, indicating that pretrained initialization can be incorporated into TRACE when available, while the main method does not depend on it.

G.3 Computational Cost

We additionally compare the computational cost of TRACE with the methods used in the natural-image experiments. Table 8 reports network size, peak GPU memory, forward/backward evaluations, reconstruction time, and the corresponding reconstruction quality for $4\times$ super-resolution.

TRACE requires substantially more computation than feed-forward plug-and-play CNN methods and several pretrained diffusion solvers, because the network is optimized directly for each measurement instance. Among the untrained neural methods with comparable optimization scale, TRACE has nearly the same measured runtime and memory usage as SDI (76.6 vs. 76.3 seconds per image and 1.66 GB for both), while achieving higher PSNR and SSIM in this benchmark. These results highlight the computational trade-off of the training-free setting: TRACE avoids external model training but performs instance-specific optimization at reconstruction time.

H Experimental details for SPC on diffusion solvers

Protocol. For each solver, Base is the released implementation with the SPC weight set to zero, which reproduces the original algorithm exactly; +SPC is the same implementation with the SPC term switched on. Apart from the SPC term, the two runs are identical: they use the same pretrained prior, hyperparameters, random seed, initial noise, and measurements, and all images of a task are processed in one paired batch. One sample is drawn per image or slice.

We index the solver's T outer steps in execution order by $s = 0, \dots, T-1$. For $s \geq 1$, the SPC term is the proximal penalty

$$\frac{\beta_s}{2} \|z_s - z_{s-1}\|_2^2$$

Table 8: Computational cost and reconstruction quality on SR $4\times$ (BSDS68, mean over 5 images $\times$ 3 random seeds). All methods run one image at a time on a single, otherwise idle NVIDIA RTX PRO 6000 Blackwell GPU with the configuration used for the main results. *Time* is the per-image reconstruction time (mean $\pm$ std over the 15 runs), CUDA-synchronised, excluding model loading, I/O and metric logging; *GPU-h* is the corresponding GPU time for 100 images. *Fwd / Bwd* counts forward / backward passes through the network; *Params* is the size of the network being optimised or evaluated. Diffusion-prior methods additionally rely on a natural-image pretrained model, whose training cost is not included. Best and second-best quality values are in bold and underlined, respectively.

Method	Params (M)	Mem. (GB)	Fwd / Bwd	Time (s)	GPU-h	PSNR↑	SSIM↑	LPIPS↓
Untrained networks (no training data)								
UGoDIT	2.2	0.48	600 / 600	3.3 ± 0.1	0.09	28.33	0.736	0.421
ASeqDIP	1.0	0.37	1200 / 1200	4.3 ± 0.0	0.12	22.93	0.596	0.570
DIP	2.2	0.55	3000 / 3000	16.9 ± 0.2	0.47	30.00	0.830	0.219
Self-Guided DIP	69.2	1.89	3600 / 3600	33.8 ± 0.1	0.94	23.30	0.619	0.614
SDI	88.1	1.66	6000 / 6000	76.3 ± 0.1	2.12	<u>30.16</u>	0.831	0.262
TRACE (ours)	88.1	1.66	6000 / 6000	76.3 ± 0.1	2.13	**30.46**	**0.852**	0.261
Pretrained CNN denoiser + plug-and-play								
PnP-FFDNet	0.9	0.04	200 / 0	0.3 ± 0.0	0.01	29.41	<u>0.851</u>	0.295
PnP-DnCNN	0.7	0.09	200 / 0	0.5 ± 0.0	0.01	30.10	0.848	0.268
Pretrained diffusion prior								
RED-diff	93.6	0.65	1000 / 0	13.8 ± 0.6	0.38	27.70	0.736	0.391
DAPS	93.6	0.93	1000 / 0	18.8 ± 0.6	0.52	29.33	0.780	0.267
SITCOM	93.6	4.42	600 / 600	19.7 ± 0.2	0.55	30.13	0.829	<u>0.186</u>
DPS	93.6	2.32	1000 / 1000	26.0 ± 0.7	0.72	26.98	0.738	**0.165**
PnP-DM	93.6	0.94	4121 / 0	58.1 ± 0.9	1.61	26.85	0.738	0.364

added to the solver's update, where z_s is the solver's current clean-image estimate and z_{s-1} is the corresponding estimate from the previous outer step. For PnP-DM and RED-diff, z_s denotes the current iterate. The SPC term is inactive at $s = 0$ because no previous outer-step estimate is available.

For DAPS and PnP-DM, the penalty is additionally divided by σ_s^2, where σ_s is the solver's current annealing noise level, matching the scaling of their existing prior term. For RED-diff, z_{s-1} is replaced by an anchor that is refreshed every 20 iterations because consecutive iterates are nearly identical.

The weight follows a linear schedule in execution order:

$$\beta_s = w \left[1 + (r-1)\frac{s}{T}\right], \qquad s = 0, \ldots, T-1.$$

The schedule assigns $\beta_0 = w$ at the beginning and

$$\beta_{T-1} = w \left[1 + (r-1)\frac{T-1}{T}\right] \approx wr$$

at the final step.

FFHQ deblurring. The task is to recover a 256×256 FFHQ face image from a measurement blurred by a 61×61 Gaussian kernel (standard deviation 3.0) and corrupted by additive Gaussian noise with standard deviation 0.05 (images scaled to $[-1, 1]$). All solvers use the same unconditional FFHQ-256 diffusion model, and results are averaged over the first 20 images of the FFHQ-256 set (indices 0–19). (w, r) is $(0.06, 2.0)$ for DPS, $(60, 0.5)$ for DAPS, $(0.3, 0.8)$ for RED-diff, $(0.3, 0.3)$ for PnP-DM and $(0.3, 0.3)$ for SITCOM. For PnP-DM we set the likelihood parameter $\tau = 0.05$ instead of 0.01 (for both Base and +SPC), since 0.01 produces hatching artifacts in flat background regions on this task. PSNR is computed by the InverseBench Zheng et al. (2025) image evaluator (prediction and target clipped to $[0, 1]$).

Multi-coil MRI. The task is to reconstruct a complex-valued 320×320 image from retrospectively undersampled, noiseless 15-coil k-space data (fastMRI knee; sensitivity maps provided with the data), using a random Cartesian phase-encoding mask with a fully sampled central band of 8% of the lines at $4\times$ (80 of 320 lines) and 4% at $8\times$ (40 of 320 lines); the same mask is used for all slices. All solvers share an unconditional complex-valued diffusion model pretrained on multi-coil fastMRI knee images. The test set consists of 10 consecutive slices (slices 005–014) of a single fastMRI knee volume (file1000033). (w, r) is $(30,\ 0.5)$ for DAPS, $(0.5,\ 0.6)$ for RED-diff, $(1.0,\ 0.8)$ for PnP-DM and $(0.1,\ 0.1)$ for SITCOM at both accelerations, and $(10^{-3},\ 1)$ at $4\times$ and $(10^{-4},\ 10)$ at $8\times$ for DPS. At $8\times$, DPS uses guidance scale 1 instead of its default 10 (for both Base and +SPC). PSNR is computed on the magnitude image, with the dynamic range set to the maximum of the ground-truth magnitude (InverseBench MRI evaluator).