Self-diffusion for Solving Inverse Problems

Guanxiong Luo
luoguan5@gmail.com

Shoujin Huang
solor.pikachu@gmail.com

Abstract

We propose *self-diffusion*, a novel framework for solving inverse problems without relying on pretrained generative models. Traditional diffusion-based approaches require training a model on a clean dataset to learn to reverse the forward noising process. This model is then used to sample clean solutions—corresponding to posterior sampling from a Bayesian perspective—that are consistent with the observed data under a specific task. In contrast, self-diffusion introduces a self-consistent iterative process that alternates between noising and denoising steps to progressively refine its estimate of the solution. At each step of self-diffusion, noise is added to the current estimate, and a self-denoiser, which is a single untrained convolutional network randomly initialized from scratch, is continuously trained for certain iterations via a data fidelity loss to predict the solution from the noisy estimate. Essentially, self-diffusion exploits the spectral bias of neural networks and modulates it through a scheduled noise process. Without relying on pretrained score functions or external denoisers, this approach still remains adaptive to arbitrary forward operators and noisy observations, making it highly flexible and broadly applicable. We demonstrate the effectiveness of our approach on a variety of linear inverse problems, showing that self-diffusion achieves competitive or superior performance compared to other methods.

1 Introduction

Inverse problems involve reconstructing a hidden cause—often called the *source* e.g., an image or signal—from observed data. The observed data is often incomplete and noisy, which leads to ill-posed problems. These problems occur in a wide range of domains and are common in fields like signal and image processing, including tasks like medical imaging reconstruction, image denoising, and low-level vision. Prior knowledge about the source is particularly valuable when addressing ill-posed problems, where observed data alone may not be sufficient for an accurate and faithful reconstruction. A Bayesian framework offers an approach in such scenarios by combining observed data with prior information. In this view, the goal is to infer the underlying source given the observations by estimating its posterior distribution. This posterior reflects a balance between the likelihood, which is determined by the forward model and noise, and the prior, which encodes assumptions or knowledge about the source. Because the likelihood is often fixed by the physical model, the choice of prior plays a critical role in improving the quality and stability of the reconstruction.

The conventional approach often relies on handcrafted priors to regularize the solution space and steer optimization toward more plausible outcomes. Classic techniques such as Tikhonov regularization [1], total variation [2], and sparsity-promoting methods [3, 4] have been widely used. While these methods can be effective in certain settings, they often struggle to faithfully recover complex signals when the observed data is insufficient. Thereafter, supervised learning approaches, since deep neural networks have gained popularity, are used to learn mappings from large datasets of input-output pairs [5, 6]. These learned models have demonstrated strong performance in specific tasks. However, their effectiveness typically depends on the availability of large amounts of labeled training data, which may not always be accessible. Furthermore, these models often lack robustness to distribution shifts,

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

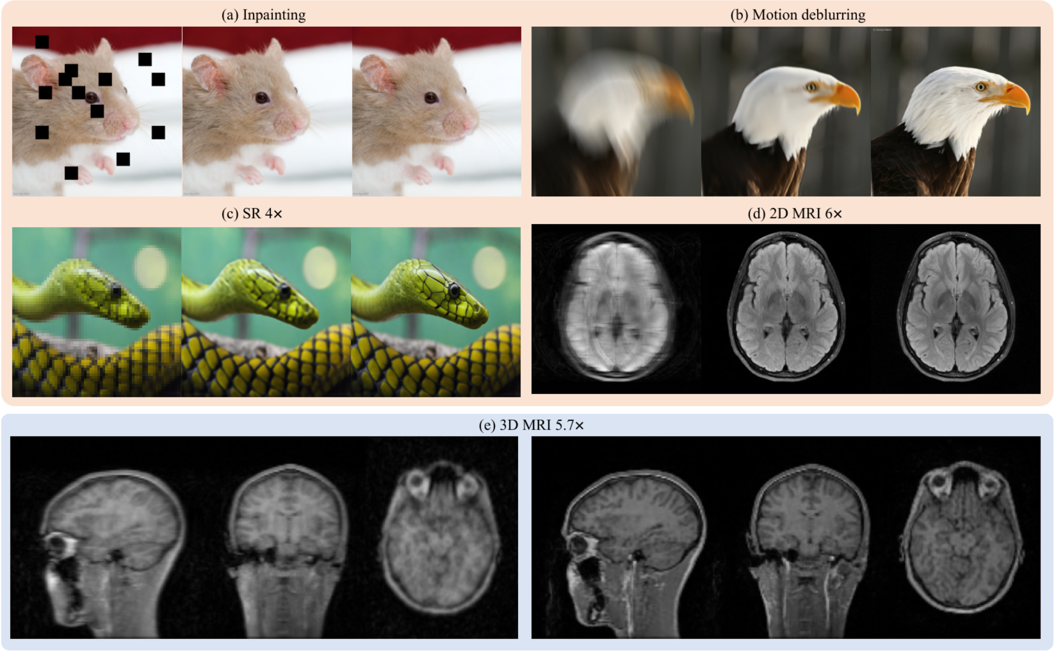

Figure 1: Demonstration of self-diffusion applied to various inverse problems across natural and medical imaging domains. (a) Image inpainting (b) Motion deblurring (c) 4× image super-resolution (SR) (d) 2D MRI reconstruction from 6× undersampled data (e) 3D MRI reconstruction from 5.7× undersampled data In each set, the leftmost image shows the degraded input, the middle shows the self-diffusion result, and the rightmost shows the ground-truth or reference.

such as varying noise levels or different acquisition settings, limiting their practical utility in some situations. Recent advances in generative modeling have introduced a powerful alternative: using generative models as learned priors. Models such as variational autoencoders (VAEs) [7], generative adversarial networks (GANs) [8], and conditional autoregressive models [9] have shown impressive capabilities in capturing complex data distributions. In particular, diffusion models [10, 11, 12] have emerged as a leading class of generative models with abilities to produce high-quality samples. Leveraging the denoising step in diffusion models, recent works have adapted them for inverse problems [13, 14, 15], using a pretrained model to guide the reconstruction process. These generative priors effectively encode structural knowledge of the data and yield robust reconstructions. However, their applicability depends heavily on access to large-scale curated datasets for training. When the data for training priors is limited or not available, these methods may not be suitable.

Interestingly, recent research has shown that deep neural networks can serve as strong implicit priors, even without pretraining. The Deep Image Prior (DIP) approach [16, 17, 18] exploits the tendency of overparameterized networks to capture natural image statistics. In DIP, a randomly initialized neural network is optimized directly on the observed measurements to minimize reconstruction error. Despite not being trained on external data, the network's inductive biases the solution toward natural-looking images. While promising, DIP's performance is sensitive to the optimization process and often suffers from premature convergence or limited expressiveness, which can lead to suboptimal results.

In this work, we propose self-diffusion, a novel approach for solving inverse problems without relying on pretrained generative models. Our method is motivated by the observation that diffusion models generate images progressively from low to high frequencies through the denoising process. Building on this insight, we design a self-diffusion process that iteratively reconstructs the source starting from noise and using a spectrum-regulated self-denoiser based on the DIP framework. During the process, a single neural network (serving as the denoiser) is continuously optimized to minimize the discrepancy between its estimated reconstruction and the observed data across different noise levels. This allows the model to refine the reconstruction in a progressive manner, guided entirely by the observed measurements, without any prior training or external datasets. Our contributions are as follows:

1. We introduce self-diffusion, a novel method for solving inverse problems that leverages the diffusion process without requiring pretrained generative models. To the best of our knowledge, this is the first such approach.
2. We provide a theoretical form of the self-diffusion process, offering a solid foundation for understanding its behavior and convergence properties from the perspective of spectral bias.
3. We evaluate self-diffusion on a range of inverse problems, including MRI reconstruction, low-level image restoration, and radar angle estimation. Experimental results demonstrate that our method achieves competitive performance compared to other methods.

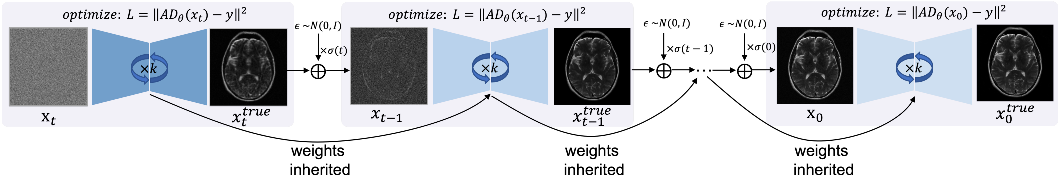

Figure 2: Overview of the self-diffusion framework. At each diffusion step t, Gaussian noise is added to the current estimate $\mathbf{x}_t^{\text{true}}$, producing a noisy input. A single self-denoiser (initialized with an untrained network) is then trained to produce $\mathbf{x}_{t-1}^{\text{true}}$ by minimizing the task-specific data fidelity loss. This iterative alternation of noising and denoising from $\mathbf{x}_T$ to $\mathbf{x}_1$ yields a coarse-to-fine reconstruction process without the need for pretrained models or external supervision.

2 Self-diffusion for inverse problem

Problem Setup. We consider the inverse problem defined by the equation

$$\mathcal{A}\mathbf{x}^{\text{true}} = \mathbf{y} \ ,$$

where $\mathcal{A} \in \mathbb{R}^{m \times n}$ is a known forward operator, $\mathbf{x}^{\text{true}} \in \mathbb{R}^n$ is the unknown deterministic solution, and $\mathbf{y} \in \mathbb{R}^m$ represents the observed data. In many practical scenarios, the problem is underdetermined, i.e., $m < n$, which results in more unknowns than equations and makes the it potentially non-unique and ill-posed. To resolve this, regularization techniques are typically applied to introduce prior knowledge and ensure the existence and uniqueness of a stable solution.

Bayesian framework is adopted in many works, where we treat $\mathbf{x}^{\text{true}}$ as a random variable with a prior distribution $p(\mathbf{x})$, reflecting our prior knowledge about the solution. The observed data $\mathbf{y}$ is modeled as an observation $\mathcal{A}\mathbf{x}^{\text{true}}$, with likelihood function $p(\mathbf{y}|\mathbf{x})$ that quantifies how probable the observed data is, given the solution $\mathbf{x}$. Bayes' theorem allows us to compute the posterior distribution of $\mathbf{x}^{\text{true}}$ given the observed data $\mathbf{y}$ and the prior $p(\mathbf{x})$

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \ ,$$

where $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\, d\mathbf{x}$ is the marginal likelihood. The solution $\mathbf{x}^{\text{true}}$ is then obtained by maximizing the posterior distribution. In this framework, the prior $p(\mathbf{x})$ can be interpreted as a form of regularization, as it enforces certain desirable properties on the solution (such as smoothness, sparsity, or structure).

Self-diffusion. To map the estimated noisy solution $\mathbf{x}_t = \mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t$ to the true solution $\mathbf{x}^{\text{true}}$, a self-diffusion process trains a self-denoiser $D_{\theta_{t,k}}$ at each noise step $t = T-1, \ldots, 0$ over $k = 0, \ldots, K-1$ iterations to minimize the loss

$$L_{t,k} = \|\mathcal{A}D_{\theta_{t,k}}(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t) - \mathbf{y}\|^2 \ , \tag{1}$$

where $\mathbf{x}_t^{\text{true}}$ is the estimate of $\mathbf{x}^{\text{true}}$ within noise step t, initialized as $\mathbf{x}_T^{\text{true}} = \epsilon_0$, with $\epsilon_0 \sim \mathcal{N}(0, I)$. The noise $\epsilon_t \sim \mathcal{N}(0, I)$ is sampled once at the start of noise timestep t and is fixed across all K iterations within that timestep. It is then resampled for the next noise step $t-1$. The noise schedule is $\sigma_t = \sqrt{1 - \bar{\alpha}_t}$, where $\bar{\alpha}_t = \prod_{i=0}^{t}(1 - \beta_i)$ and $\beta_t = \beta_{\text{end}} + \frac{t}{T-1}(\beta_{\text{start}} - \beta_{\text{end}})$. After K iterations at each noise step, the self-denoiser $D_{\theta_{t,k}}$ produce the estimated solution $\mathbf{x}_{t-1}^{\text{true}}$ for the next noise step $t-1$. As the noise level $\sigma(t)$ decreases to zero, the predicted solution converges to the true solution $\mathbf{x}^{\text{true}}$ under Theorem 1 in Heckel and Soltanolkotabi (2020). The detailed demonstration of this process is in appendix A. The pseudo-code for implementation is shown in Algorithm 1.

Since ϵ_t is fixed within each noise step but resampled at each subsequent noise step, the process $\mathbf{x}_t$ is piecewise stochastic. In the continuous limit, this process can be modelled with a stochastic differential equation (SDE) over time. Considering the noise resampling across steps, the SDE is

$$d\mathbf{x}_t = (\mathbf{x}_t - \mathbf{x}^{\text{true}})dt + \sigma(t)dW(t) \ , \tag{2}$$

where $W(t)$ is a Wiener process capturing the resampling of noise across time steps. The drift term $(\mathbf{x}_t - \mathbf{x}^{\text{true}})$ reflects the self-denoiser's push toward $\mathbf{x}^{\text{true}}$. The derivation of this SDE is provided in appendix B. This self-diffusion process is a combination of the forward and reverse processes because: 1) the forward process is self-referential ($\mathbf{x}_t = \mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t$), not starting from the curated noise-free training dataset; 2) the proposed self-denoiser directly learns to predict $\mathbf{x}^{\text{true}}$ at each noise step instead of relying on the pretrained score function or denoiser.

Algorithm 1 Self-diffusion (SDI) for solving the inverse problem

```
Input: $\mathbf{x}_T^{\text{True}}$; noise steps - $T$; iterations-$K$; learning rate - $\eta$; forward operator - $\mathcal{A}$; initialize
  $\theta_{T,0}$; default noise schedule $\beta_{\text{start}} = 0.0001$, $\beta_{\text{end}} = 0.01$
for $t = T - 1$ to 0 do
    Sample $\epsilon_t \sim \mathcal{N}(0, I)$, $\mathbf{x}_t = (\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t)$
    for $k = 0$ to $K - 1$ do
        Compute loss $L_{t,k}$ and $\nabla_{\theta_{t,k}} L_{t,k}$
        Update the weights using optimizer and learning rate $\eta$
    end for
    Set $\theta_t = \theta_{t,K}$
    Compute $\mathbf{x}_{t-1}^{\text{true}} = D_{\theta_t}(\mathbf{x}_t)$
end for
return $\mathbf{x}_0^{\text{true}}$
```

Noise-Modulated Spectral Bias. Neural networks, especially in overparameterized regimes, are known to exhibit spectral bias—a tendency to learn low-frequency components of a target function before high-frequency ones [19]. This behavior has been theoretically linked to the eigenspectrum of the Neural Tangent Kernel (NTK) [20], which governs the dynamics of gradient descent during training. In particular, the NTK often has larger eigenvalues associated with smooth, low-frequency modes and smaller ones for high-frequency oscillations. Consequently, gradient updates are naturally biased toward reconstructing coarse, low-frequency structures earlier in training.

The training of self-denoiser in self-diffusion exploits this spectral bias and further regulates it through the structured noise schedule. Considering the self-denoiser in Equation (1), we expand the expected loss function using a first-order Taylor expansion into

$$\mathbb{E}_{\epsilon_t}[||\mathcal{A}D_{\theta_{t,k}}(\mathbf{x}_t) - \mathbf{y}||^2] \approx \underbrace{||\mathcal{A}D_{\theta_{t,k}}(\mathbf{x}_t^{\text{true}}) - \mathbf{y}||^2}_{\text{Data Fidelity Term}} + \underbrace{\sigma(t)^2||\mathcal{A}J_D(\mathbf{x}_t^{\text{true}})||_F^2}_{\text{Regularization Term}}$$

where the second term acts as a regularizer, discouraging the network from fitting sharp, high-frequency fluctuations. The derivation of this expansion is provided in appendix C.1.

This regularization effect is modulated by the noise level σ_t. In early steps of diffusion, where noise is large, the regularization is strong, enforcing smooth outputs and prioritizing low-frequency components. As the process progresses and σ_t decreases, the regularization weakens, allowing the network to focus on finer, high-frequency details. This design leads to an implicit multi-scale learning regime, where the reconstruction transitions from global structure to local refinement. We reveal this process in Fourier space as detailed in appendix C.2.

3 Simulation

We apply the self-diffusion inference to recover a sparse 1D signal from compressed sensing measurements. The original signal is generated as a sum of sine wave with varying frequencies and amplitudes and has a sparse representation in the frequency domain. The signal $\mathbf{x}$ is constructed as

$$\mathbf{x} = \sum_{(A,f)\in\mathcal{S}} A \cdot \sin\left(2\pi f \frac{t}{N}\right) \ .$$

We have a predefined set of amplitude-frequency pairs $\mathcal{S}$ as

$$\mathcal{S} = \{(1.0, 1),\ (0.5, 15),\ (0.3, 20),\ (1.0, 6),\ (0.8, 3),\ (0.6, 4),\ (0.7, 5)\}\ ,$$

and N is the total signal length, with $t \in \{0, 1, ..., N-1\}$. This signal is deliberately designed to span both low and high frequencies with non-uniform amplitudes to challenge the network's ability to recover both coarse and fine details from compressed measurements. The signal is measured using a random Gaussian sampling matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, where $m \ll n$. The measurement matrix is normalized to have unit Frobenius norm. The measurements are then obtained as $\mathbf{y} = \mathbf{Ax}$. In this experiment, the signal length is 128, and the number of measurements is 35.

To recover the signal, we employ a 1D U-Net architecture with bottleneck layers and normalization layers as self-denoiser. All convolutional layers in the denoising network are initialized using a Gaussian distribution with zero mean and standard deviation of 0.02, while biases are initialized to zero. This initialization helps stabilize training and ensures the early network outputs maintain consistent scale and variance across layers. The parameters for Algorithm 1 is $\{T = 40; K = 200; \eta = 1e^{-5}; \beta_{\text{start}} = 4e^{-3}; \beta_{\text{end}} = 1e^{-6}\}$ and Adam optimizer is used. We use DIP to reconstruct the signal with the same learning rate and initialized network as SDI. The loss used in both methods are enhanced with L1 penalty in frequency domain. We also run ADMM-basis pursuit in [21], which is a classical compressed sensing method, as another baseline. Figure 3 shows the original signal and the recoverd signal in time domain and frequency domain. The progression of signal recovery over denoising steps is illustrated in the supplementary video *freq_prog.mp4*. In the early iterations, the model primarily recovers the low-frequency components of the signal, which dominate the coarse structure in the time domain. As the process continues, higher-frequency details gradually emerge. This progressive reconstruction—from low to high frequencies —demonstrates the self-denoising capability of refining the initial estimate iteratively through successive denoising steps while benefitting from the spectral bias of the self-denoiser.

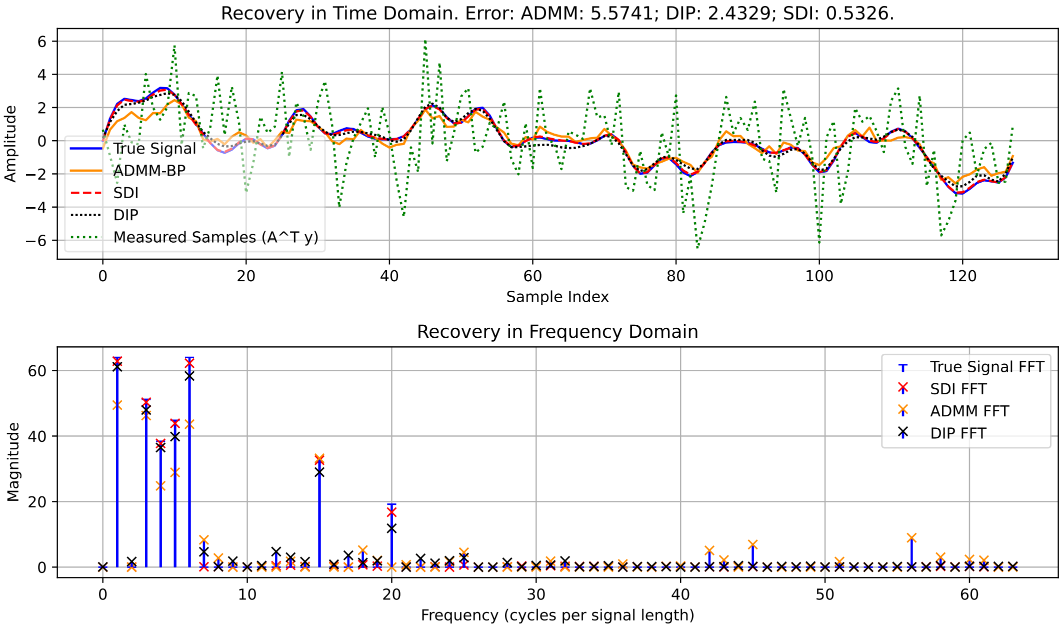

Figure 3: Recovery of a sparse signal composed of multiple sine waves from compressed measurements. The top plot illustrates time-domain reconstruction, comparing self-diffusion inference (SDI) to ADMM-basis pursuit (ADMM-BP) and Deep Image Prior (DIP). The bottom plot displays the frequency spectra, where SDI more accurately captures both low- and high-frequency components compared to ADMM-BP and DIP.

4 Application

To demonstrate the versatility and generality of self-diffusion across a range of inverse problems spanning both natural and medical imaging domains, we evaluate our method on several representative tasks, including 2D and 3D MRI reconstruction, low-level vision restoration (inpainting, deblurring, denoising, super-resolution), and radar angle estimation recovery. The implementation details for each task are provided in the following sections. The code is available at github:ggluo/self-diffusion.

4.1 MRI reconstruction

2D sampling. We evaluate the effectiveness of self-diffusion on 2D MRI reconstruction from undersampled k-space data. The raw k-space is subsampled using a equispaced Cartesian mask with an acceleration factor of 4x/6x along the phase-encoding direction. And 20 central auto-calibration signal (ACS) lines are obtained. From the fastMRI validation set [22], we randomly selected 20 samples for each contrast (T1, T1 post contrast, T2, and FLAIR) to form a test set of 80 subjects. We estimated coil sensitivity maps estimated using BART's `ecalib` command. Measurements are formed with $\mathcal{A}$, which consists of the undersampling mask, Fourier transform, and coil sensitivities.

For self-diffusion, we initialize the 2D U-Net with Gaussian initialization $\{\mu = 0, \sigma = 0.02\}$ for convolutional kernels and zero initialization for biases. To ensure stable optimization and consistent data fidelity weighting, we normalize the k-space measurements before input to the reconstruction process. Specifically, we compute a scale factor to match the norm of the initialized network's output to the norm of $\mathcal{A}^H\mathbf{y}$. The k-space measurements are scaled accordingly, preventing the network from unstable gradients due to mismatched signal energy. We use Adam optimizer in Algorithm 1 and set parameters $\{T = 40; K = 50; \eta = 0.001; \beta_{\text{start}} = 1e^{-3}; \beta_{\text{end}} = 1e^{-4}\}$. We compare SDI to the following baseline methods: Aseq [23], IMJENSE [24], and Deep Image Prior (DIP). For these Aseq and IMJENSE, we use authors's official implementations. The loss for DIP and SDI is enhanced with the total variation term, weight is set to 0.0001. As an ablation study, we also include SDI without resampling noise in Algorithm 1 and refer it as "w/o ϵ_t SDI". The PSNR, SSIM, and NRMSE metrics are computed against the fully sampled ground truth. Table 1 shows the reconstruction quality for all methods. SDI achieves the highest PSNR and SSIM, and the lowest NRMSE, outperforming Aseq and DIP that share the characteristic of using untrained networks but lack the iterative diffusion process. Figure 4 compares the reconstructed images for 2D MRI from 4× undersampled k-space data without ACS lines. Visually, SDI reconstructs sharper anatomical boundaries and finer textures with fewer aliasing artifacts compared to other baselines. In Appendix E, Figures 10 and 11 illustrate the visual results for 4× and 6× undersampled k-space data, respectively, and Figure 8 reveals the evolution of the estimate over the self-denoising process.

Table 1: Comparison for MRI reconstruction from 4× and 6× undersampled data.

	4×	6×	4× w/o ACS
Method	PSNR↑/SSIM↑/NRMSE↓	PSNR↑/SSIM↑/NRMSE↓	PSNR↑/SSIM↑/NRMSE↓
$\mathcal{A}^H\mathbf{y}$	26.78 / 0.7411 / 0.0472	25.84 / 0.6990 / 0.0526	15.01 / 0.4546 / 0.1815
IMJENSE	38.80 / 0.9604 / 0.0125	35.30 / 0.9398 / 0.0183	N/A
Aseq	32.18 / 0.8661 / 0.0272	29.74 / 0.8249 / 0.0353	30.87 / 0.8460 / 0.0318
DIP	38.35 / 0.9574 / 0.0136	34.92 / 0.9368 / 0.0192	36.34 / 0.9454 / 0.0178
w/o ϵ_t SDI	39.09 / 0.9639 / 0.0120	36.21 / 0.9474 / 0.0165	37.12 / 0.9497 / 0.0160
SDI	39.21 / 0.9640 / 0.0116	36.84 / 0.9492 / 0.0151	38.20 / 0.9541 / 0.0136

Comparison with methods using pretrained diffusion model. We compared SDI with CSGM [25] on reconstructing MR images of different contrast (T2 and FLAIR) as shown in Table 2. The diffusion model released in Ref. [25] was trained on T2 contrast images. CSGM performs better in the reconstruction of T2 contrast images except in the case without ACS lines. CSGM requires 1150 the network forward evaluations per image. When CSGM apply this pre-trained model to FLAIR images, its performance is not as good as T2 contrast images. However, this does not mean that SDI is superior to methods using pretrained diffusion model as Ref [13] showed better cross-contrast generalization performance. Further domain-specific comparisons are omitted as they are beyond the scope of this work.

Table 2: Compare SDI to CSGM on reconstructing different contrasted images. (PSNR↑/SSIM↑/NRMSE↓)

Acceleration	CSGM / T2	w/o ϵ_t SDI / T2	SDI / T2	CSGM / FLAIR	w/o ϵ_t SDI / FLAIR	SDI / FLAIR
4×	38.78 / 0.9693 / 0.0119	38.30 / 0.9683 / 0.0135	38.08 / 0.9681 / 0.0131	37.41 / 0.9435 / 0.0153	37.95 / 0.9572 / 0.0138	38.41 / 0.9564 / 0.0131
6×	37.55 / 0.9636 / 0.0136	35.11 / 0.9519 / 0.0186	35.83 / 0.9555 / 0.0166	35.46 / 0.9286 / 0.0190	35.21 / 0.9367 / 0.0187	35.81 / 0.9387 / 0.0175
4× w/o ACS	34.00 / 0.9376 / 0.0208	36.55 / 0.9543 / 0.0191	37.66 / 0.9620 / 0.0152	32.53 / 0.9015 / 0.0262	36.39 / 0.9454 / 0.0164	37.18 / 0.9408 / 0.0151

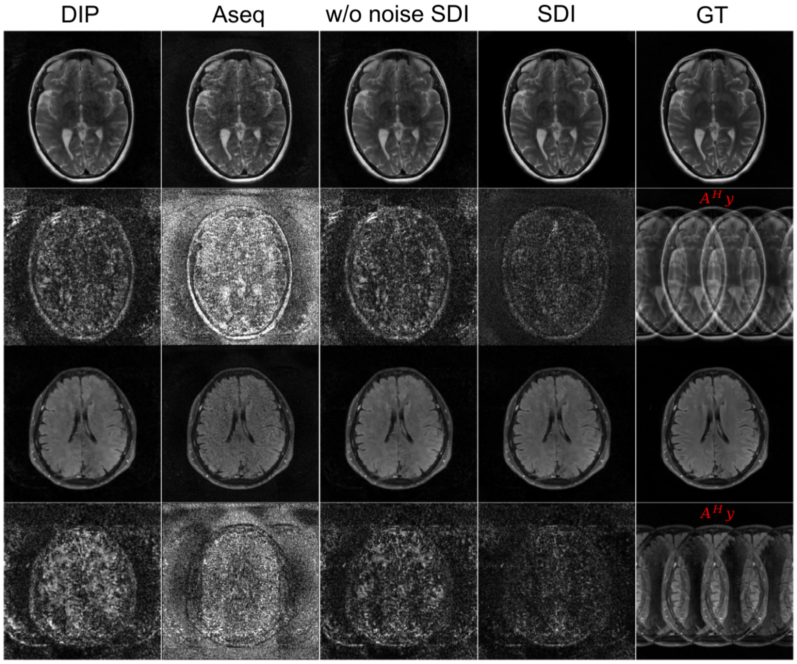

Figure 4: Reconstructions and error maps to GT from 4× undersampled k-space w/o ACS lines. $\mathcal{A}^H\mathbf{y}$ is the degraded.

4.2 Radar angle estimation recovery

Radar sensors are widely used in various applications, including automotive, aerospace, and etc. However, the resolution of radar angle estimation is often limited by the number of antennas (spatial sampling rate). This is a typical task that ground truth data is hard to obtain. Ref. [26] formulates it as an inverse problem, and we apply self-diffusion to recover unknown high-resolution angle estimations $\mathbf{x}$ from low-resolution radar range-azimuth heatmaps $\mathbf{y}$ which represent the spatial locations of objects through signal power. Following the setup in [26], we leverage a large-scale autonomous driving perception RADIal dataset [27]. The measurement of low-resolution radar range-azimuth heatmap is synthesized by applying the radar angle measurement process to the objects that captured by LiDAR sensors under 86 antennas (spatial sampling rate). The same as previous settings, the denoising network is randomly initialized. The parameters for Algorithm 1 is $\{T = 30; K = 100; \eta = 1e^{-3}; \beta_{\text{start}} = 2e^{-2}; \beta_{\text{end}} = 1e^{-4}\}$ and Adam optimizer is used. One thing deserves to be mentioned is that because the supposed output of our model is a 2D binary mask which can be easily thresholded to obtain the final point cloud, we use a sigmoid activation function as the last layer of the U-Net. Figure 5 illustrates the radar recovery using self-diffusion on the RADIal dataset and four different scene from the dataset are shown. The results demonstrate significant resolution improvement despite hardware limitations making labeled data for radar enhancement challenging to obtain.

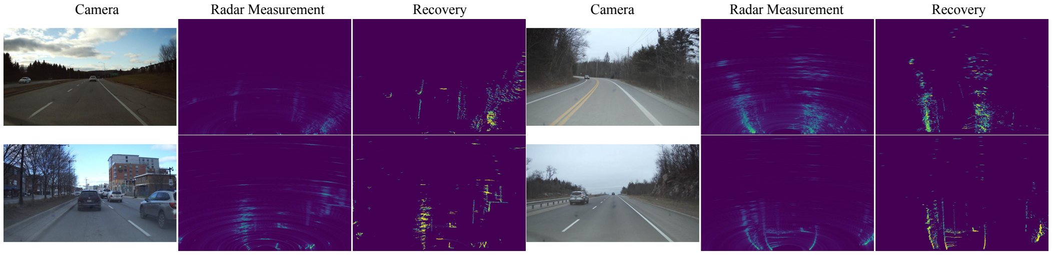

Figure 5: The camera column shows the visual context captured by the camera, depicting the road, vehicles, and surroundings. The radar measurement column displays the low-resolution radar range-azimuth heatmaps $\mathbf{y}$. The recovery column presents the recovered high-resolution angle estimations.

4.3 Low-level vision tasks

Evaluation. We evaluate self-diffusion on three representative low-level vision tasks: image inpainting, motion deblurring, and single-image super-resolution. Each task is formulated as $\mathbf{y} = \mathcal{A}\mathbf{x}$, where $\mathbf{x}$ is the unknown image, $\mathbf{y}$ is the observation and $\mathcal{A}$ is a task-specific degradation operator. The experiments are performed on 1000 images that are from the dataset ImageNet and resized to 256×256. For image inpainting, random rectangular masks remove regions of the image and the goal is to successfully fills missing regions with coherent structure and texture. For motion deblurring, the blur is generated using code from Git:LeviBorodenko/motionblur with kernel size 61 × 61 and intensity value 0.3, and goal is to reconstruct sharp images from blurry inputs. In super-resolution tasks, low-resolution images are generated via average pooling and the goal is to recover high-resolution images from low-resolution inputs. We use the Algorithm 1 for all the tasks above and set it with parameters $\{T = 40; K = 150; \eta = 0.001, \beta_{\text{start}} = 1e^{-4}; \beta_{\text{end}} = 1e^{-2}\}$. We also compare SDI to two baselines: DIP [16] and Aseq [23]. The loss for DIP is enhanced with the total variation term. Evaluation metrics like PSNR, SSIM, and LPIPS are computed against the ground truth and are shown in Table 3. The visual results are illustrated in Figures 12 to 15 listed in Appendix F.

Table 3: Reconstruction quality comparison for low-level vision tasks. (PSNR↑ / SSIM↑ / LPIPS↓)

Task	$\mathbf{y}$	Aseq	DIP	w/o ϵ_t SDI	SDI
2× SR	26.91 / 0.8924 / 0.0502	27.26 / 0.8359 / 0.0531	29.06 / 0.8530 / 0.0388	26.97 / 0.8852 / 0.0543	30.30 / 0.9186 / 0.0351
4× SR	22.45 / 0.7392 / 0.0833	24.14 / 0.7535 / 0.0719	24.55 / 0.6855 / 0.0650	24.09 / 0.7583 / 0.0704	25.82 / 0.8110 / 0.0585
Motion	18.84 / 0.5763 / 0.1272	22.65 / 0.6886 / 0.0840	25.94 / 0.7375 / 0.0589	26.92 / 0.8202 / 0.0513	27.76 / 0.8459 / 0.0481
Inpainting	18.06 / 0.8845 / 0.1339	28.06 / 0.8589 / 0.0516	29.41 / 0.9310 / 0.0384	25.42 / 0.8840 / 0.0804	30.78 / 0.9296 / 0.0422

Denoising. We included denoising results on CBSD68 dataset [28] with noise level $\sigma = 25$. SDI use the parameters $\{T = 30; K = 100; \eta = 0.002; \beta_{\text{start}} = 1e^{-2}; \beta_{\text{end}} = 8e^{-4}\}$. DIP has 3000 iterations enhanced with the total variation term whose weight is set to $5e^{-4}$. The other baselines use default parameters. As shown in Table 4, SDI performances worse than FFDNET and IR-SDE but better than DIP. The supervised method FFDNET provides the best performance in terms of PSNR and SSIM. IR-SDE is the best in terms of LPIPS. In addition to the numerical metrics, we provided visual comparisons in Figure 6.

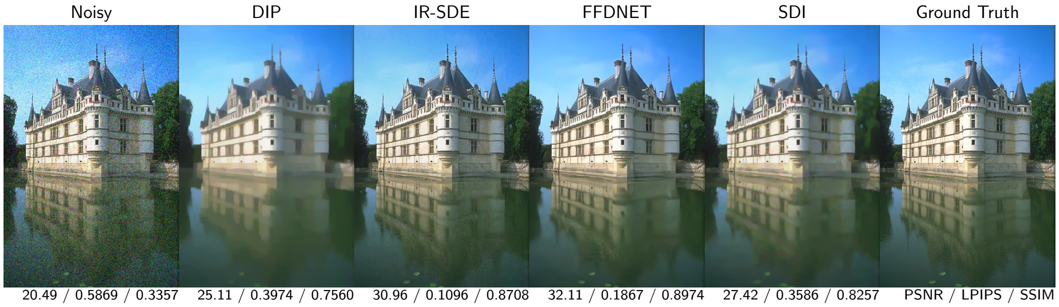

Figure 6: Visual comparison of denoising results across different methods against the ground truth. IR-SDE by Luo et al., 2023; FFDNET by Zhang et al., 2018.

Large image We further evaluated SDI on a large image of size (5000×3000), which is from an inexperienced shooter and corrupted by dust spots and minor noise. We masked the dust spots using a binary mask then restored the image using SDI via parameters $\{T = 40; K = 500; \eta = 0.001; \beta_{\text{start}} = 1e^{-2}; \beta_{\text{end}} = 1e^{-4}\}$. Results are shown in Figure 7. The noise on the cloudy sky and sea surface are removed, while the details on the architecture are preserved. The restoration takes around 2 hours on RTX A6000. As limited by the size of paper, we will post the boosted image at ggluo:self-diffusion for better display.

Table 4: Comparison for denoising

Method	PSNR↑ / LPIPS↓ / SSIM↑
DIP	25.44 / 0.307 / 0.6602
FFDNET	31.22 / 0.121 / 0.8821
IR-SDE	28.09 / 0.101 / 0.7866
SDI	28.10 / 0.207 / 0.7860

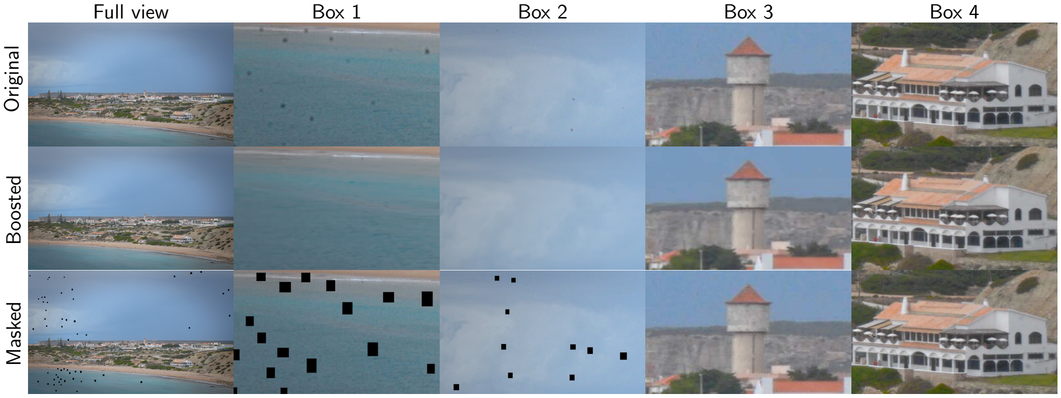

Figure 7: SDI boosts image quality. The original image is corrupted by dust spots and minor noise.

4.4 Hyperparameters sensitivity

Table 5 presents PSNR (dB) values for various combinations of iterations (K) and noise steps (T) when reconstructing an MRI image from a 6x-undersampled k-space. The following trends are observed: 1) For each fixed T, PSNR generally increases with K up to a certain point, then plateaus or slightly declines; 2) Higher T values tend to yield higher PSNR at lower K, but the advantage diminishes as K increases; 3) After reaching a peak, PSNR tends to stabilize or decrease slightly. Total iterations ($T \times K$) range from 250 ($T = 10, K = 25$) to 20000 ($T = 40, K = 500$). Generally, more iterations lead to higher PSNR before plateauing. Therefore, when choosing hyperparameters, first select T to balance performance and computational cost, then consider K for optimal PSNR.

Table 5: PSNR for different iterations K and steps T choices.

$T\backslash K$	25	50	75	100	150	200	300	400	500
10	24.65	28.89	31.16	33.12	34.41	35.11	35.95	35.96	36.06
20	29.35	33.27	34.04	35.30	35.69	36.15	36.33	36.02	36.24
40	32.13	34.54	35.60	35.96	36.19	36.17	36.07	36.04	36.11

Table 6 summarizes runtime and iteration comparison for different reconstruction methods.

Table 6: Runtime and iteration comparison for different reconstruction methods.

	IMJENSE	Aseq	CSGM	SDI	DPS	DDNM	SDI
Task, Size	MRI, 320×320	MRI, 320×320	MRI, 320×320	MRI, 320×320	low-level vision, 256×256	low-level vision, 256×256	inpainting, 5000×3000
Iter/sec	11.92	16.94	11.39	16.94	8.31	19.64	1.52
Total iters	1500	2000	1156	2000	1000	100	20000
GPU info	RTX 4090, 24GB	RTX 3090 Ti, 24GB	RTX 4090, 24GB	RTX 3090 Ti, 24GB	RTX 4090, 24GB	RTX 4090, 24GB	RTX A6000, 48GB

5 Discussions

In this work, we proposed self-diffusion as a novel approach for solving inverse problems, which combines elements of denoising, diffusion processes, and the spectral bias of neural networks. Self-diffusion leverages the spectral bias of overparameterized neural networks, which naturally prioritize low-frequency components during optimization. This bias is enhanced by a decreasing noise schedule, creating a hierarchical, coarse-to-fine reconstruction process. Early denoising steps focus on smooth, low-frequency structures, while later steps refine high-frequency details. This behavior mimics multiscale optimization—without requiring explicit architectural scale separation—and is well suited to many inverse problems as demonstrated by our experiments.

Compared to DIP, self-diffusion exhibits three key advantages: 1) Initialization robustness: DIP performance is sensitive to weight initialization, often requiring careful tuning. Self-diffusion, in contrast, is largely robust to random initializations due to its iterative, noise-modulated training scheme; 2) Frequency generalization: DIP tends to overfit high-frequency components early in training when data is noisy. Self-diffusion, through its structured noise schedule, maintains better control over frequency content during optimization; 3) Better zero-shot understanding: The noise in self-diffusion promotes understanding of image, which leads to more coherent structure and texture as shown in Figure 12. While this is the opposite case for the "w/o noise SDI" and DIP.

The self-diffusion framework introduces a form of implicit regularization driven by noise. The denoiser's Jacobian is captured by the differential $\nabla D_{\theta_{t,k}}$, and the term $\sigma(t)^2||\mathcal{A}J_D(\mathbf{x}_t^{\text{true}})||_F^2$ quantifies the frequencies bias. This mechanismis naturally similar to the NTK perspective, where low-frequency modes have larger eigenvalues and thus converge faster. The regularization term amplifies this bias by suppressing first-order derivatives, effectively damping high-frequency components as shown in Appendix C.2.

While self-diffusion shows strong generalization and performance across different tasks, several promising directions remain for further exploration: 1) Adaptive noise scheduling: learning task-specific or image-adaptive noise schedules may yield better convergence and fidelity; 2) Neural architecture search: automatically discovering architectures better suited for self-diffusion could improve both efficiency and reconstruction quality; 3) Hybriding with pretrained models: combining self-diffusion with pretrained features or priors may boost performance on more complex or semantic tasks.

6 Conclusions

Self-diffusion introduces a simple yet powerful framework for solving inverse problems without external data or pretrained models. By leveraging the intrinsic spectral bias of neural networks and regulating it through a structured noise schedule, it enables progressive, coarse-to-fine reconstruction. This approach achieves strong performance across diverse inverse problems like medical image reconstruction, signal recovery, and low-level vision tasks while remaining architecture- and data-agnostic. The results highlight the potential of self-supervised optimization guided by structured noise, opening new directions for learning-free and data-efficient image restoration.

References

[1] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977.

[2] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, 1992. ISSN 0167-2789. doi: 10.1016/0167-2789(92)90242-F. URL `https://www.sciencedirect.com/science/article/pii/016727899290242F`.

[3] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4): 1289–1306, 2006. doi: 10.1109/TIT.2006.871582.

[4] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. doi: 10.1109/TIT.2005.862083.

[5] Kyong Hoon Jin, Michael T. McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017. doi: 10.1109/TIP.2017.2713099.

[6] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017. doi: 10.1088/1361-6420/aa9581.

[7] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[9] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[10] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2256–2265. JMLR.org, 2015.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[12] 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. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

[13] Guanxiong Luo, Moritz Blumenthal, Martin Heide, and Martin Uecker. Bayesian mri reconstruction with joint uncertainty estimation using diffusion models. *Magnetic Resonance in Medicine*, 90(1):295–311, 2023.

[14] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=OnD9zGAGT0k`.

[15] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[17] Dave Van Veen, Ajil Jalal, Mahdi Soltanolkotabi, Eric Price, Sriram Vishwanath, and Alexandros G. Dimakis. Compressed sensing with deep image prior and learned regularization. *arXiv preprint arXiv:1806.06438*, 2018. URL `https://arxiv.org/abs/1806.06438`.

[18] Reinhard Heckel and Mahdi Soltanolkotabi. Compressive sensing with un-trained neural networks: Gradient descent finds a smooth approximation. In *International conference on machine learning*, pages 4149–4158. PMLR, 2020.

[19] Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJeqhA4YDS`.

[20] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[21] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[22] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.

[23] Ismail Alkhouri, Shijun Liang, Evan Bell, Qing Qu, Rongrong Wang, and Saiprasad Ravishankar. Image reconstruction via autoencoding sequential deep image prior. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=K1EG2ABzNE`.

[24] Ruimin Feng, Qing Wu, Jie Feng, Huajun She, Chunlei Liu, Yuyao Zhang, and Hongjiang Wei. Imjense: Scan-specific implicit representation for joint coil sensitivity and image estimation in parallel mri. *IEEE Transactions on Medical Imaging*, 43(4):1539–1553, 2024. doi: 10.1109/TMI.2023.3342156.

[25] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021.

[26] Yanlong Yang, Jianan Liu, Guanxiong Luo, Hao Li, Euijoon Ahn, Mostafa Rahimi Azghadi, and Tao Huang. Unsupervised radar point cloud enhancement via arbitrary lidar guided diffusion prior, 2025. URL `https://arxiv.org/abs/2505.09887`.

[27] Julien Rebut, Arthur Ouaknine, Waqas Malik, and Patrick Pérez. Raw high-definition radar for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17021–17030, 2022.

[28] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423 vol.2, 2001. doi: 10.1109/ICCV.2001.937655.

[29] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *International Conference on Machine Learning*, 2023.

[30] 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.

A Error Propagation under Ideal Denoiser Performance

We follow the setup for self-diffusion described in Section 2. All solutions to the ill-posed inverse problem can be parameterized as:

$$\mathbf{x} = \mathcal{A}^{\dagger}\mathbf{y} + \mathbf{z}_{\text{null}}$$

where $\mathcal{A}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathcal{A}$, and $\mathbf{z}_{\text{null}} \in \text{null}(\mathcal{A})$ is an arbitrary vector in the null space of $\mathcal{A}$. The challenge is to identify $\mathbf{x}^{\text{true}}$ from this affine solution space. The self-diffusion algorithm iteratively refines an estimate of $\mathbf{x}^{\text{true}}$. The process iterates backward in "noise time" steps $t = T-1, T-2, \ldots, 0$. At each step t, a neural network denoiser $D_{\theta_{t,k}}$ (e.g., a CNN) is trained by minimizing the loss:

$$L_{t,k} = \|\mathcal{A}D_{\theta_{t,k}}(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t) - \mathbf{y}\|^2$$

where $\mathbf{x}_t^{\text{true}} \in \mathbb{R}^n$ is the image estimate at step t. We initialize it with $\mathbf{x}_{T-1}^{\text{true}} = \epsilon_0,\ \epsilon_0 \sim \mathcal{N}(0, I_n)$ is a random noise vector. $\epsilon_t \sim \mathcal{N}(0, I_n)$ is a sample of Gaussian noise, fixed during the K inner training iterations within a given timestep t, and resampled for the next step $t-1$. $\sigma(t) > 0$ is a positive noise schedule, monotonically decreasing such that $\sigma(t) \to 0$ as $t \to 0$. $\theta_{t,k}$ are parameters of the denoiser at inner iteration k of step t. Then, the iterative update is given by:

1. Sample $\epsilon_t \sim \mathcal{N}(0, I_n)$, set $\mathbf{x}_t = \mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t$;
2. For $k = 0, \ldots, K-1$, compute the gradient of $L_{t,k}$ w.r.t. $\theta_{t,k}$ and update parameters;
3. Set the denoiser for step t as $D_{\theta_t} = D_{\theta_{t,K}}$;
4. Update the image estimate via $\mathbf{x}_{t-1}^{\text{true}} = D_{\theta_t}(\mathbf{x}_t)$.

We define the relative change error as $\mathbf{e}_t = \|\mathbf{x}_t^{\text{true}} - \mathbf{x}_{t-1}^{\text{true}}\|^2 / \|\mathbf{x}_t^{\text{true}}\|^2$. In this section, we analyze the behavior of the relative change error $\mathbf{e}_t$ under idealized denoiser performance i.e., the following holds as $t \to 0$:

$$\mathbb{E}_{\epsilon_t}[\mathbf{e}_t] \to 0 \quad \text{and} \quad \mathbb{E}_{\epsilon_t}[\|\mathbf{x}_t^{\text{true}} - \mathbf{x}^{\text{true}}\|^2] \to 0. \tag{3}$$

Heckel and Soltanolkotabi (2020) claimed that there exists a unique true image $\mathbf{x}^{\text{true}}$ that satisfies $\mathcal{A}\mathbf{x}^{\text{true}} = \mathbf{y}$ and untrained network possesses a structure (e.g., piecewise smoothness, low-frequency dominance) learnable by D_{θ_t}. Therefore, we view the trained D_{θ_t} act as a denoiser, implicitly favoring the structure of $\mathbf{x}^{\text{true}}$. For each step t, the K inner training iterations are sufficient for the denoiser D_{θ_t} to approximately satisfy the measurement constraint $\mathcal{A}D_{\theta_t}(\cdot) \approx \mathbf{y}$ when applied to inputs of the form $\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t$. The denoiser's error with respect to the true signal, $\delta_t(\mathbf{x}_{\text{noisy}}) := D_{\theta_t}(\mathbf{x}_{\text{noisy}}) - \mathbf{x}^{\text{true}}$, has an expected squared norm $\zeta_t := \mathbb{E}_{\epsilon_t}[\|\delta_t(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t)\|^2]$. We assume that $\zeta_t \to 0$ as $t \to 0$, when the measurements $\mathbf{y} = \mathcal{A}\mathbf{x}$ is enough to recover the true signal using the implict structure of denoiser D_{θ_t}. In the following, we show that the convergence of $\mathbf{e}_t$ in Equation (3) is determined by the denoiser's error with respect to the true signal, i.e., ζ_t.

A.1 Update Rule and Error Dynamics

Let $\mathbf{d}_t = \mathbf{x}_t^{\text{true}} - \mathbf{x}^{\text{true}}$ be the absolute error of the estimate at step t. The update rule for the image estimate is:

$$\mathbf{x}_{t-1}^{\text{true}} = D_{\theta_t}(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t)$$

Using the definition of δ_t, we have:

$$\mathbf{x}_{t-1}^{\text{true}} = \mathbf{x}^{\text{true}} + \delta_t(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t) .$$

Subtracting $\mathbf{x}^{\text{true}}$ from both sides gives the error recurrence:

$$\mathbf{d}_{t-1} = \delta_t(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t)$$

Taking the expected squared Euclidean norm:

$$\mathbb{E}_{\epsilon_t}[\|\mathbf{d}_{t-1}\|^2] = \mathbb{E}_{\epsilon_t}[\|\delta_t(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t)\|^2] = \zeta_t$$

Shifting the index, we get the expected squared error for the iterate $\mathbf{x}_t^{\text{true}}$:

$$\mathbb{E}_{\epsilon_t}[\|\mathbf{d}_t\|^2] = \zeta_{t+1}$$

Now, let's analyze the relative change error e_t

$$\mathbf{e}_t = \frac{\|\mathbf{x}_t^{\text{true}} - \mathbf{x}_{t-1}^{\text{true}}\|^2}{\|\mathbf{x}_t^{\text{true}}\|^2} = \frac{\|(\mathbf{x}_t^{\text{true}} - \mathbf{x}^{\text{true}}) - (\mathbf{x}_{t-1}^{\text{true}} - \mathbf{x}^{\text{true}})\|^2}{\|\mathbf{x}_t^{\text{true}}\|^2} = \frac{\|\mathbf{d}_t - \mathbf{d}_{t-1}\|^2}{\|\mathbf{x}_t^{\text{true}}\|^2} \ .$$

The norms of the iterates remain bounded below by a positive constant, i.e., $\|\mathbf{x}_t^{\text{true}}\|^2 \geq c^2 > 0$ for all relevant t, which is reasonable as $\mathbf{x}_t^{\text{true}} \to \mathbf{x}^{\text{true}}$ and $\|\mathbf{x}^{\text{true}}\| > 0$. Then, we get

$$\mathbb{E}_{\epsilon_t}[\mathbf{e}_t] \leq \frac{1}{c^2}\mathbb{E}_{\epsilon_t}[\|\mathbf{d}_t - \mathbf{d}_{t-1}\|^2] \ .$$

Using the inequality $\|\mathbf{a} - \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, we have

$$\mathbb{E}_{\epsilon_t}[\mathbf{e}_t] \leq \frac{2}{c^2}(\mathbb{E}_{\epsilon_t}[\|\mathbf{d}_t\|^2] + \mathbb{E}_{\epsilon_t}[\|\mathbf{d}_{t-1}\|^2]) \ .$$

Substitute the expressions in terms of ζ:

$$\mathbb{E}_{\epsilon_t}[\mathbf{e}_t] \leq \frac{2}{c^2}(\zeta_{t+1} + \zeta_t) \ .$$

Since $\zeta_t \to 0$ as $t \to 0$, then

$$\mathbb{E}_{\epsilon_t}[\mathbf{e}_t] \to 0 \quad \text{as} \quad t \to 0 \ .$$

A.2 Denoiser as a Proximal Operator

With sufficient training iterations (i.e., large K), the denoiser learns to produce outputs that are consistent with the measurements. This consistency enforced through the untrained network leads to implicitly favoring of smoothed images instead of the least-squares solution, which often suffers from artifacts and poor perceptual quality. The optimization dynamics of training the denoiser impart a spectral bias on the solution, favoring smoother, low-frequency components. This behavior can be interpreted through the lens of proximal optimization, whereby the denoiser acts approximately as a proximal operator:

$$D_{\theta_t}(\mathbf{x}_t) \approx \text{prox}_{\lambda_t R}(\mathbf{x}_t) = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{z} - \mathbf{x}_t\|^2 + \lambda_t R(\mathbf{z}),$$

where $R(\mathbf{z})$ is an implicit regularization functional that discourages undesirable features such as high-frequency noise or implausible structures. We will analyze in more detail, in Appendix C, how the spectral bias shaped by the noise schedule.

B Continuous Time Approximation

The discrete update rule in previous section is written as

$$\mathbf{x}_{t-1}^{\text{true}} - \mathbf{x}_t^{\text{true}} = D_{\theta_t}(\mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t) - \mathbf{x}_t^{\text{true}} \ .$$

Since t decreases from T to 0 and let $x(t) = \mathbf{x}_t^{\text{true}}$, we have

$$x(t + \Delta t) - x(t) = x(t-1) - x(t) = D_{\theta_t}(x(t) + \sigma(t)\epsilon_t) - x(t) \ ,$$

where the change in t is negative ($\Delta t = -1$). In the continuous limit $\Delta t \to 0$, we have

$$\frac{dx(t)}{dt} = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} \ .$$

Since $\Delta t = -1$, the right hand side becomes

$$x(t-1) - x(t) = D_{\theta_t}(x(t) + \sigma(t)\epsilon_t) - x(t) \ ,$$

which leads to

$$\frac{dx(t)}{dt} \approx \frac{x(t) - x(t-1)}{1} = -\left(D_{\theta_t}(x(t) + \sigma(t)\epsilon_t) - x(t)\right) \ .$$

With $D_{\theta_t}(x(t) + \sigma(t)\epsilon_t) = \mathbf{x}^{\text{true}} + \delta_t(x(t) + \sigma(t)\epsilon_t)$, we have

$$\begin{aligned} \frac{dx(t)}{dt} &= -\big((\mathbf{x}^{\text{true}} + \delta_t(x(t) + \sigma(t)\epsilon_t)) - x(t)\big) \\ &= x(t) - \mathbf{x}^{\text{true}} - \delta_t(x(t) + \sigma(t)\epsilon_t) \end{aligned}$$

As $t \to 0$, $\sigma(t) \to 0$, and $\delta_t \to 0$, we have the ordinary differential equation

$$\frac{dx(t)}{dt} \approx x(t) - \mathbf{x}^{\text{true}} \ .$$

Since ϵ_t is fixed within each noise step but resampled at each new noise step, the process is piecewise stochastic. In the continuous limit, we model the resampling as a stochastic process over time. The SDE form, considering the noise resampling across steps, is

$$dx(t) = (x(t) - \mathbf{x}^{\text{true}})dt + \sigma(t)dW(t)$$

where $W(t)$ is a Wiener process capturing the resampling of noise across noise steps. The drift term $x(t) - \mathbf{x}^{\text{true}}$ reflects the denoiser's push towards $\mathbf{x}^{\text{true}}$, and the diffusion term $\sigma(t)dW(t)$ models the noise resampling.

C Derivation of Noise-Modulated Spectral Bias

The regularization term,as described in the previous section, arises from the noise added to the input of the self-denoiser and plays a critical role in enhancing the spectral bias. The term is derived from the expected loss over the noise distribution and is approximated as a regularizer. Below, we elaborate on the regularization term, its mathematical form, its effect in the Fourier domain, and its role in the self-diffusion process.

C.1 Regularization Perspective

To find a second-order approximation for the expected loss, $\mathbb{E}_{\epsilon_t}[||\mathcal{A}D(x_t) - y||^2]$, where $\mathbf{x}_t = \mathbf{x}_t^{\text{true}} + \sigma(t)\epsilon_t$ and $\epsilon_t \sim \mathcal{N}(0, I)$. For clarity, let's simplify the notation for a single step t: $D(\cdot) = D_{\theta_{t,k}}(\cdot)$, $x_0 = \mathbf{x}_t^{\text{true}}$, $\sigma = \sigma(t)$, $\epsilon = \epsilon_t$, and let the full transformation be $f(x) = \mathcal{A}D(x)$. The loss is $L = ||f(x_0 + \sigma\epsilon) - y||^2$. We want to compute $\mathbb{E}_\epsilon[L]$.

We start by approximating the output of the function $f(x)$ for a small perturbation $\sigma\epsilon$ around the point x_0. Using a first-order multivariate Taylor expansion, we get:

$$f(x_0 + \sigma\epsilon) \approx f(x_0) + J_f(x_0)(\sigma\epsilon)$$

where $J_f(x_0)$ is the Jacobian matrix of the function f evaluated at x_0. The (i, j)-th element of J_f is $\frac{\partial f_i}{\partial x_j}$. Now, we substitute this approximation back into the loss expression:

$$L \approx ||(f(x_0) - y) + \sigma J_f(x_0)\epsilon||^2 \ .$$

To simplify, let's define the residual vector $r = f(x_0) - y$. The expression becomes:

$$L \approx ||r + \sigma J_f(x_0)\epsilon||^2 \ .$$

We can expand the squared Euclidean norm (dot product) as $(v)^T(v)$:

$$L \approx (r + \sigma J_f(x_0)\epsilon)^T(r + \sigma J_f(x_0)\epsilon) \ ,$$
$$L \approx r^T r + 2\sigma r^T J_f(x_0)\epsilon + \sigma^2(J_f(x_0)\epsilon)^T(J_f(x_0)\epsilon) \ ,$$
$$L \approx ||r||^2 + 2\sigma r^T J_f(x_0)\epsilon + \sigma^2\epsilon^T J_f(x_0)^T J_f(x_0)\epsilon \ .$$

This gives us three terms to analyze. We now take the expectation of the loss L with respect to the noise distribution $\epsilon \sim \mathcal{N}(0, I)$.

1. The residual $r = f(x_0) - y$ does not depend on ϵ.

$$\mathbb{E}_\epsilon[||r||^2] = ||r||^2 = ||f(x_0) - y||^2$$

2. This term is linear in ϵ. Since the expectation of the noise is zero, $\mathbb{E}_\epsilon[\epsilon] = 0$, this term vanishes.

$$\mathbb{E}_\epsilon[2\sigma r^T J_f(x_0)\epsilon] = 2\sigma r^T J_f(x_0)\mathbb{E}_\epsilon[\epsilon] = 0$$

3. This term is a quadratic form in ϵ. For a random vector $\epsilon \sim \mathcal{N}(0, I)$ and a constant matrix Q, we have the property $\mathbb{E}[\epsilon^T Q\epsilon] = \text{Tr}(Q)$.

$$\mathbb{E}_\epsilon[\sigma^2\epsilon^T J_f(x_0)^T J_f(x_0)\epsilon] = \sigma^2\text{Tr}(J_f(x_0)^T J_f(x_0))$$

The trace of $J^T J$ is equal to the squared Frobenius norm of the matrix J, denoted $||J||_F^2$ by convention.

$$\sigma^2 \text{Tr}(J_f(x_0)^T J_f(x_0)) = \sigma^2 ||J_f(x_0)||_F^2$$

Combining the expectations of the three terms, we arrive at the final approximation for the expected loss:

$$\mathbb{E}_\epsilon[L] \approx ||f(x_0) - y||^2 + \sigma^2 ||J_f(x_0)||_F^2$$

Substituting back our original notation ($f(x) = \mathcal{A}D(x)$, $\mathbf{x}_t = \mathbf{x}_t^{true} + \epsilon_t$ and $J_f = J_{\mathcal{A}D} = \mathcal{A}J_D$):

$$\mathbb{E}_{\epsilon_t}[||\mathcal{A}D(\mathbf{x}_t) - y||^2] \approx \underbrace{||\mathcal{A}D(\mathbf{x}_t^{\text{true}}) - y||^2}_{\text{Data Fidelity Term}} + \underbrace{\sigma(t)^2 ||\mathcal{A}J_D(\mathbf{x}_t^{\text{true}})||_F^2}_{\text{Regularization Term}}$$

This term penalizes the squared Frobenius norm of the Jacobian of the entire transformation from the denoiser's input to its final output. The Jacobian measures how much the output changes in response to small changes in the input. Penalizing its norm forces the learned function D to be smoother and less sensitive to high-frequency noise. This directly achieves the goal of modulating the spectral bias. By penalizing large derivatives, the optimization is biased toward learning low-frequency components first, especially when the noise level $\sigma(t)$ is high. This provides a rigorous foundation for the coarse-to-fine reconstruction behavior described in the previous section.

C.2 Fourier Domain Analysis

To understand the effect of the regularization term, we decompose the denoiser output in the orthonormal Fourier basis $\{\mathbf{e}_k\}_{k=1}^n$, where each $\mathbf{e}_k$ corresponds to a Fourier mode with frequency index k. The denoiser's output can then be expressed as

$$D(\cdot) = \sum_{k=1}^{n} d_{t,k} \, \mathbf{e}_k,$$

where $d_{t,k}$ are the Fourier coefficients at time t. The derived regularizer is

$$R(D) = \sigma(t)^2 \, \|\mathcal{A}J_D(\mathbf{x}_t^{\text{true}})\|_F^2,$$

where J_D denotes the Jacobian of the denoiser D, $\mathcal{A}$ is a known linear operator, and $\| \cdot \|_F$ is the Frobenius norm. A first-order differential operator acts on a Fourier mode $\mathbf{e}_k \sim e^{ik \cdot x}$ by scaling its amplitude by its frequency magnitude $|k|$. Hence, by Parseval's theorem, the squared L_2-norm of the gradient of D can be expressed in terms of its Fourier coefficients as

$$\|\nabla D\|^2 = \int \|\nabla D(x)\|^2 \, dx \propto \sum_k |k|^2 |d_{t,k}|^2.$$

Since the Frobenius norm of the Jacobian, $\|J_D\|_F^2$, equals the sum of the squared L_2-norms of the gradients of each output component, it follows that

$$\|J_D\|_F^2 \propto \sum_k |k|^2 |d_{t,k}|^2.$$

Thus, the Jacobian term introduces a frequency-dependent quadratic penalty on the Fourier coefficients, scaling as k^2. Substituting this into the regularizer gives

$$R(D) = \sigma(t)^2 \, \|\mathcal{A}J_D\|_F^2 \quad \Rightarrow \quad R(D) \propto \sigma(t)^2 \sum_k |k|^2 \, \|\mathcal{A}\mathbf{e}_k\|^2 |d_{t,k}|^2,$$

showing that higher-frequency modes are penalized more strongly. Moreover, the penalty strength decreases over time through the factor $\sigma(t)^2$. We can therefore write an effective loss in the Fourier domain as:

$$\mathbb{E}_{\epsilon_t}[L_{t,k}] \approx \underbrace{\sum_{k=1}^{n} \|\mathcal{A}\mathbf{e}_k\|^2 (d_{t,k} - c_k)^2}_{\text{Data Fidelity}} + \underbrace{\sigma(t)^2 \sum_{k=1}^{n} \kappa |k|^2 \|\mathcal{A}\mathbf{e}_k\|^2 |d_{t,k}|^2}_{\text{Jacobian Regularizer}},$$

where c_k are the Fourier coefficients of the true signal and κ is a proportionality constant. This formulation confirms that the noise-modulated regularizer naturally suppresses high-frequency components, with a time-dependent weight controlled by $\sigma(t)^2$.

D Evolution of the reconstruction process

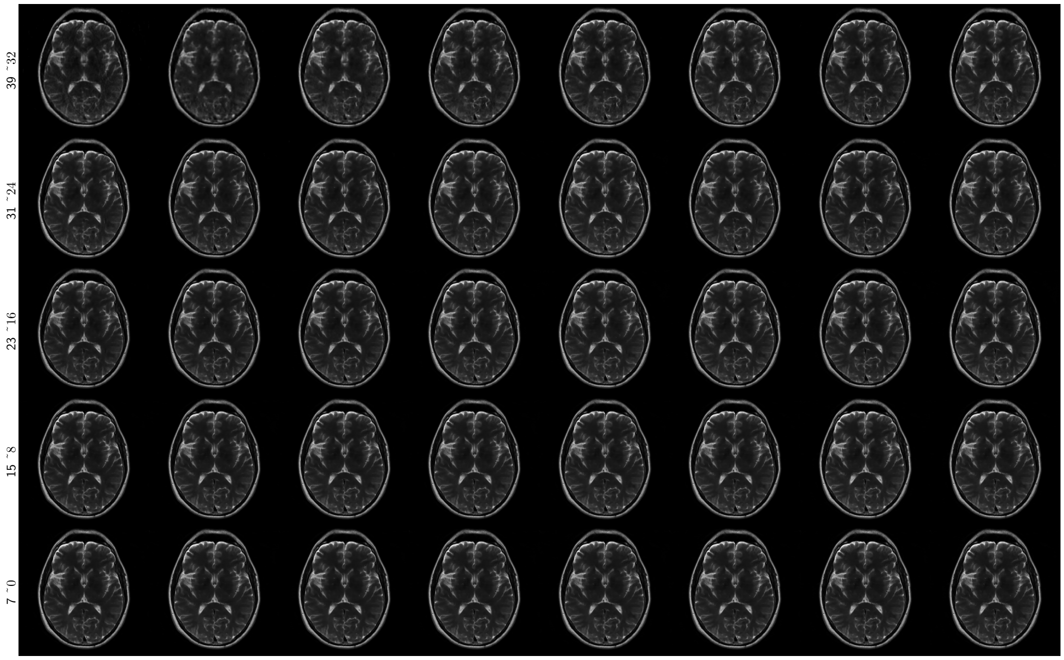

Figure 8: The evolution of reconstruction $\mathbf{x}_t^{\text{true}}$ over noise timestep from 39 to 0.

E MRI Reconstruction

3D sampling. To further demonstrate the flexibility of self-diffusion, we apply it to the reconstruction of volumetric 3D MRI data from undersampled k-space measurements. This task poses additional challenges due to the increased dimensionality and the need for spatial coherence across slices. We consider a 12-coil 3D T1-weighted brain scan and apply a Cartesian undersampling mask on the k-space that has a dimension of $160 \times 160 \times 128$. The undersampling mask is generated using BART's command `poisson` with an acceleration factor of 2.5 along two phase-encoding directions. This leads to a total acceleration factor of 8. The coil sensitivities are estimated using BART's command `ecalib`. The measurements $\mathbf{y}$ are formed with $\mathcal{A}$, which consists of the undersampling mask, 3D Fourier transform, and coil sensitivities. To accommodate 3D dimensionality, we adapt the denoising network to a 3D U-Net architecture with 3D convolutional layers and which is initialized in the same way as the 2D U-Net. The parameters for Algorithm 1 are $\{T = 40; K = 50; \eta = 0.001\}$ and Adam optimizer is used. Figure 9 shows the reconstructed sagittal, coronal, and axial slices.

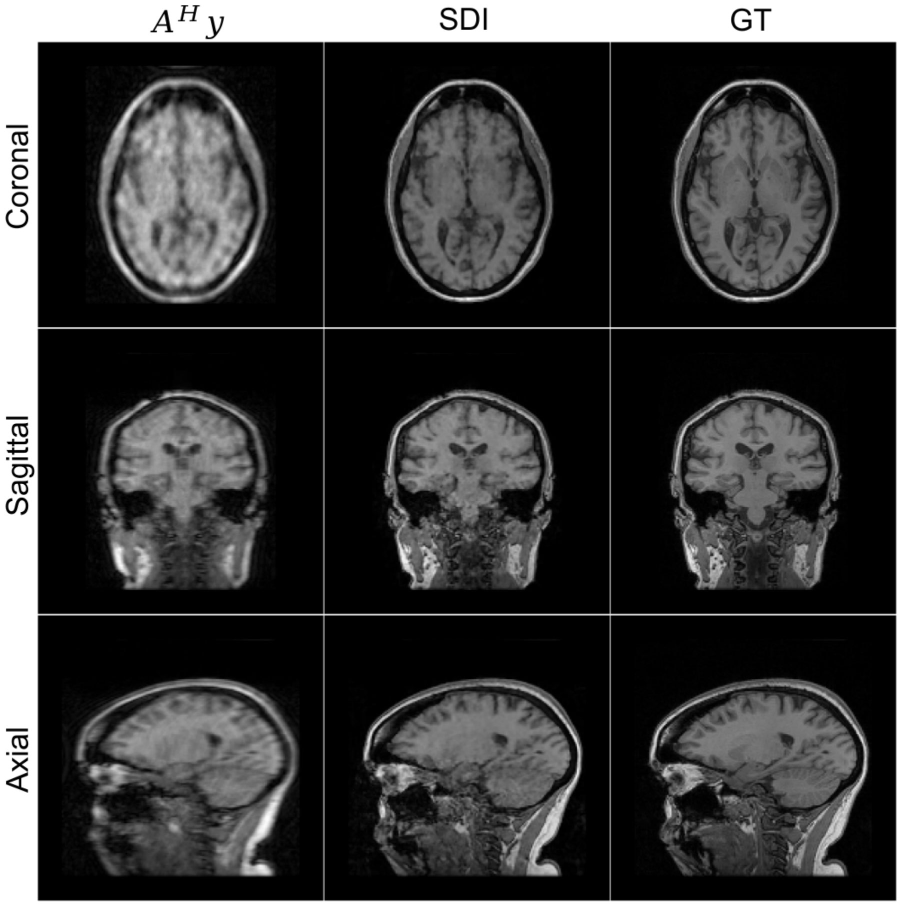

Figure 9: Reconstruction of a 3D T1-weighted brain MRI using self-diffusion from 8× undersampled measurements. Shown are sagittal, coronal, and axial slices, demonstrating sharp anatomical structures and minimal artifacts. The initial measurement ($\mathcal{A}^H \mathbf{y}$), self-diffusion inference (SDI), and ground truth (GT).

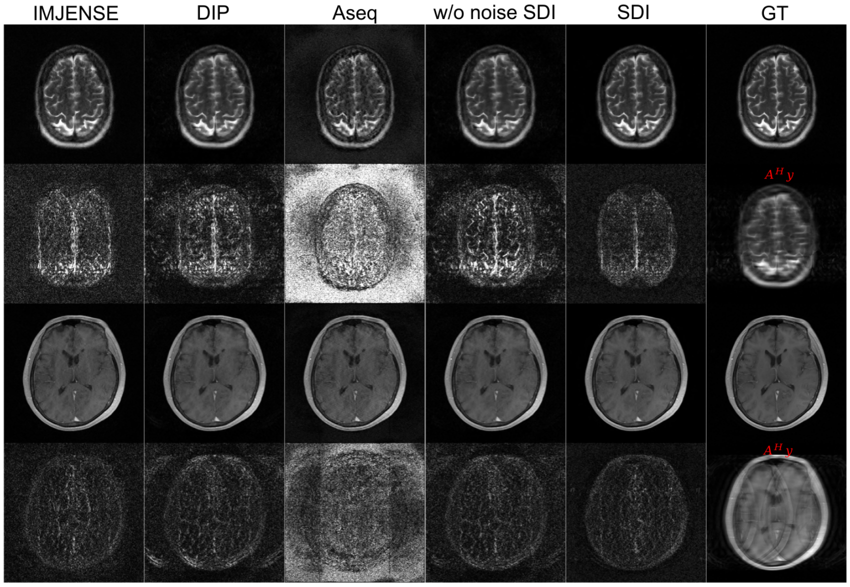

Figure 10: Reconstruction from 4× undersampled k-space with 20 ACS lines using different methods ($\mathcal{A}^H \mathbf{y}$,IMJENSE, DIP, Aseq, w/o noise SDI, SDI) and corresponding error maps are shown.

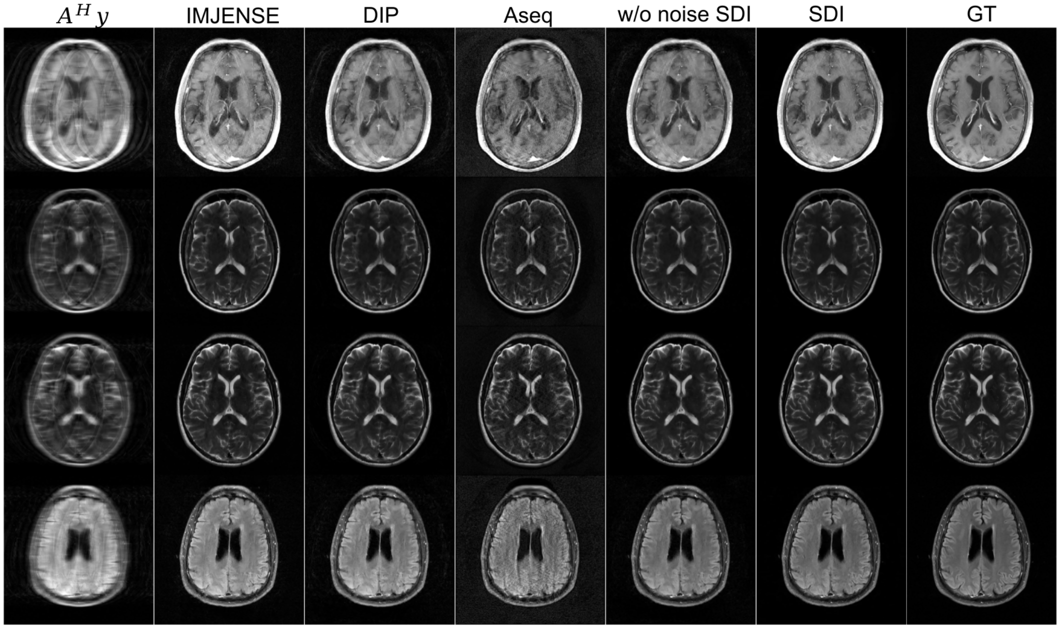

Figure 11: Reconstruction from 6× undersampled k-space with 20 ACS lines using different methods (IMJENSE, DIP, Aseq, w/o noise SDI, SDI).

F Low-level Vision Tasks

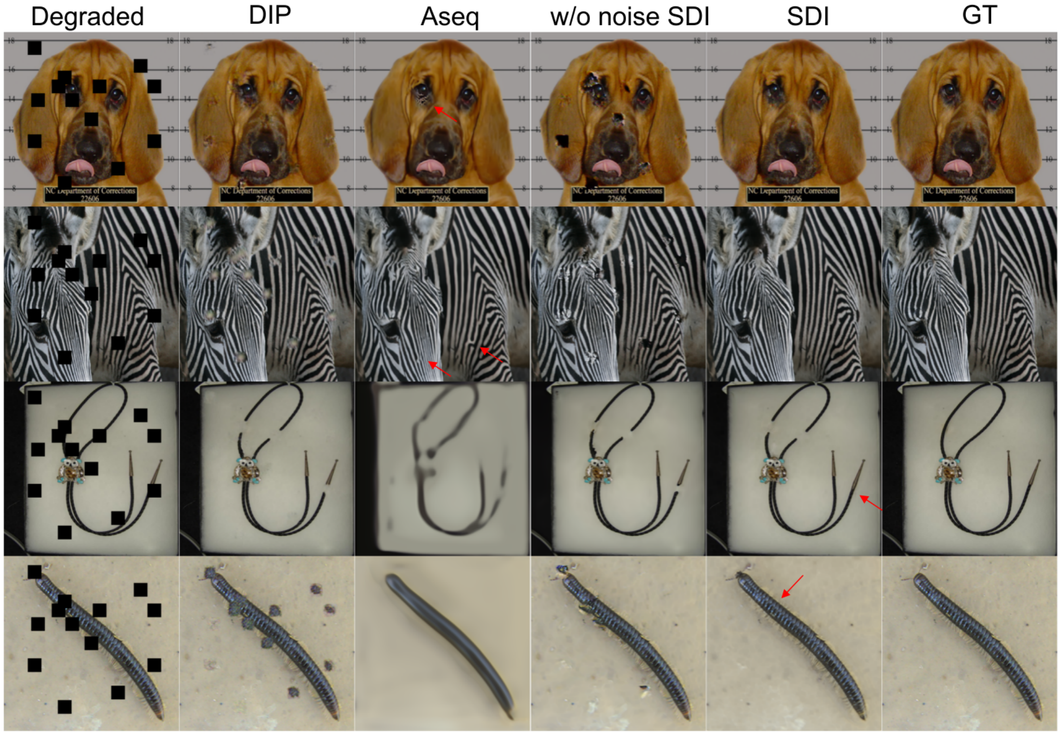

Figure 12: Comparison of inpainting results across different methods (DIP, Aseq, w/o noise SDI, SDI) against the ground truth (GT).

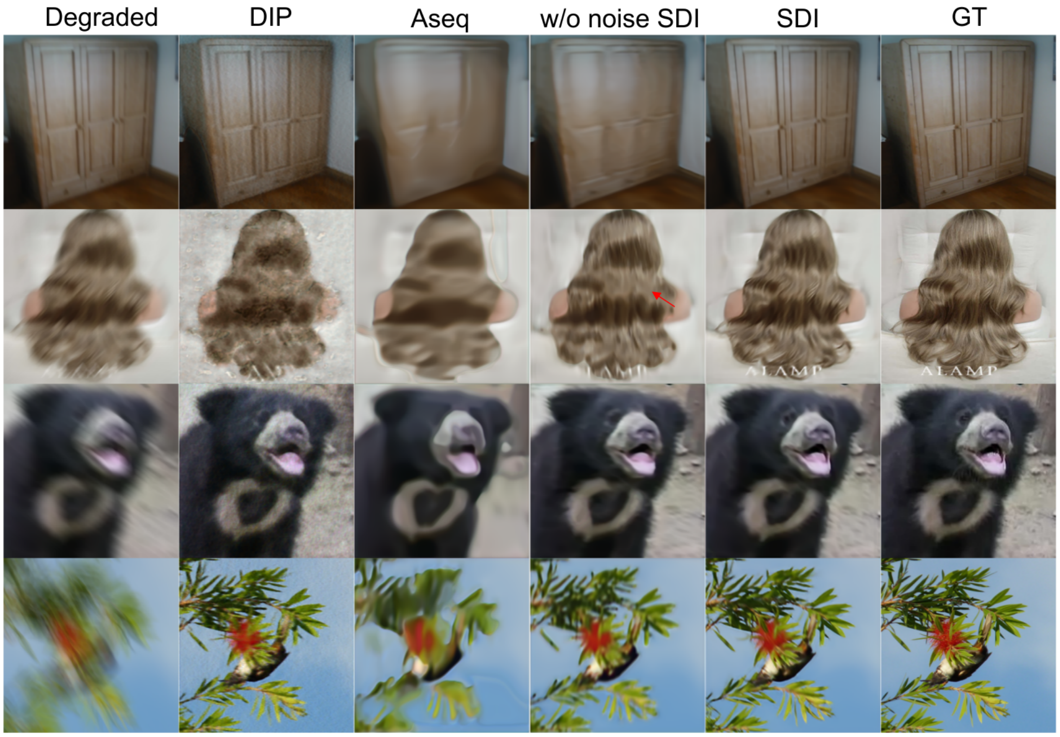

Figure 13: Comparison of motion deblurring results across different methods (DIP, Aseq, w/o noise SDI, SDI) against the ground truth (GT).

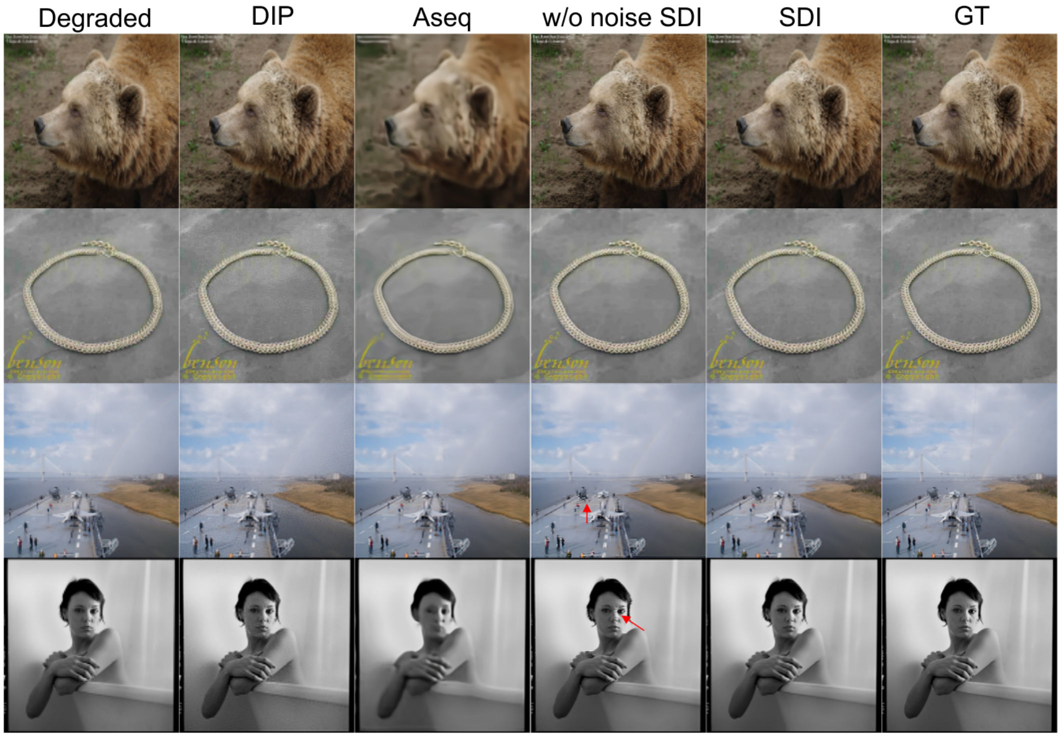

Figure 14: Comparison of 2× super-resolution results across different methods (DIP, Aseq, w/o noise SDI, SDI) against the ground truth (GT), illustrating the enhancement of image details in various scenes including a bear, a necklace, an aircraft carrier, and a person in a bathtub.

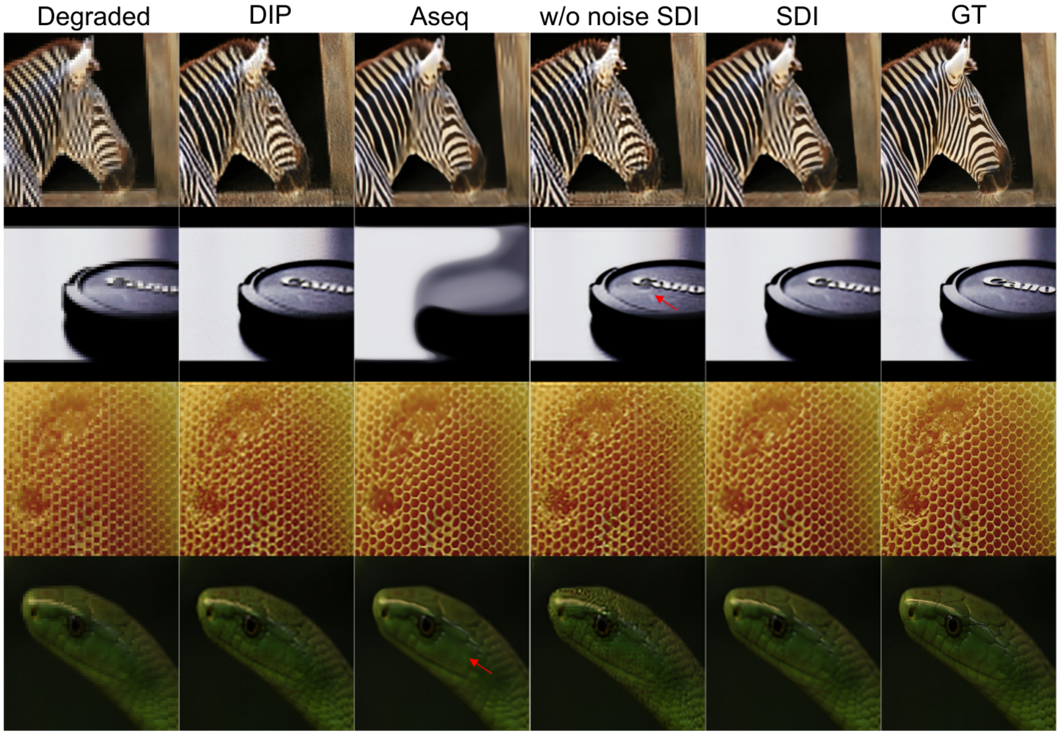

Figure 15: Comparison of 4× super-resolution results across different methods (DIP, Aseq, w/o noise SDI, SDI) against the ground truth (GT), illustrating the enhancement of fine details in images, including a zebra, a lens lid, a honeycomb, and a snake.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See abstract and introduction in Section 1

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See the discussion in Section 5

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: See Appendices A to C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: See Sections 3 and 4

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: We will make sure that the data and code are open access after review.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See Sections 3 and 4

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [NA]

   Justification: N/A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [NA]

   Justification: This work is not computation intensive.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This work has neutral societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This work does not pose a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We will license our code and data under CC-BY 4.0.

    Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: We will prepare documentation for our code and data.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [No]

    Justification: We did not involve crowdsourcing.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

Justification: We use chatgpt for writing, editing, and formatting purposes.

Guidelines:

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