Backward Diffusion iterates Noising-Relaxed Denoising

Arthur Leclaire, Eliot Guez, Bruno Galerne

Backward Diffusion iterates Noising-Relaxed Denoising

Arthur LECLAIRE[1] Eliot GUEZ[1] Bruno GALERNE[2,3]

[1]LTCI, Télécom Paris, IP Paris

[2]Université d'Orléans, Université de Tours, CNRS, IDP, UMR 7013, Orléans, France

[3]Institut Universitaire de France (IUF)

Résumé – L'objectif de cette communication est d'offrir une perspective synthétique sur les liens entre les modèles de génération d'images par diffusion et le problème classique de débruitage additif gaussien. Nous montrons comment formuler les schémas de diffusion comme un algorithme alternant des étapes de bruitage et débruitage relaxé. Ceci permet de mieux comprendre les schémas de bruit utilisés, d'accélérer l'échantillonnage du modèle, et aussi d'utiliser ces schémas avec d'autres débruiteurs existants. Enfin, nous questionnons l'utilisation des débruiteurs par diffusion pour régulariser des problèmes inverses dans un cadre *plug-and-play*, et nous exhibons de potentiels problèmes de stabilité que ces régularisations très profondes induisent.

Abstract – The goal of this paper is to offer a synthetic view of connections between the celebrated diffusion models and classical additive Gaussian denoising. It allows to formulate standard diffusion schemes as a simple iterative noising-relaxed denoising process. By bringing new understanding on the noise schedules, this allows to accelerate the model sampling, and also to use diffusion schemes with off-the-shelf denoisers. Finally, we question the use of diffusion-based denoisers to regularize inverse problems in a plug-and-play fashion, and highlight potential stability problems induced by such very deep regularizations.

1 Introduction

Imaging inverse problems are often formulated in a Bayesian setting, where one has to choose a prior that encodes some kind of image regularity [10, 16]. Following major progress on denoising neural networks [9], plug-and-play (PnP) algorithms [12, 13, 11, 5] allow to use a pre-learned denoiser (acting as an implicit regularization) in order to address more general inverse problems. More recently, diffusion models [8, 14] solved very difficult generative modeling problems, triggering their subsequent use as generative prior for image restoration tasks [7, 6]. The main goal of this paper is to offer a condensed view on the connection between diffusion models and classical additive Gaussian denoising.

Some connections have already been discovered between (score-based) denoising and prior sampling [15, 4]. Our goal here is show how a diffusion scheme (e.g. DDPM [8]) can be simply expressed through the use of a denoiser for additive Gaussian noise. We believe that our formulation could be more accessible to the image processing community. In addition to considerably simplifying the presentation of generative diffusion algorithms, our study also allows to explore simpler and lighter variants of DDPM sampling, and also to bring new understanding on the noise schedule. For example, it allows for simple direct pruning of a noise schedule underlying a pre-learned diffusion model, which considerably accelerates sampling. Also, our framework allows to use diffusion schemes with other (possibly non-deep) denoisers.

Notice that the results shown in this paper required no training of neural networks and are based only on pre-learned neural networks: the **DRUNet** denoiser of [9] (which was trained on a combination of databases of "natural" images) and the score-based denoiser of [2] referred to as **DiffUNet** (which was trained on the FFHQ dataset [7] of face images).

2 Noising-Denoising Schemes

Let q_0 be a probability distribution on $\mathbf{R}^d$, e.g. the empirical distribution of a database of clean images (with color values in $[0,1]^3$).

2.1 Denoising Diffusion Model

First, we recall the construction of the denoising diffusion probabilistic model (DDPM) proposed in [8] and its relation to score-based denoisers. The DDPM model consists in two discrete-time stochastic processes defined on $[0:T] = \{0,\dots,T\}$: a forward model $(X_t)_{t\in[0:T]}$ (which is an iterative noising process) and a backward model $(\tilde{X}_t)_{t\in[0:T]}$ (which is an iterative denoising process that depends on a trained neural network). These processes depend on sequences of parameters $(\alpha_t)_{t\in[1:T]}, (\beta_t)_{t\in[1:T]}$.

The DDPM forward model is defined by

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{\beta_t} Z_t \quad \text{for } t = 1,\dots,T \tag{1}$$

where (Z_t) are i.i.d. $\mathcal{N}(0, \mathsf{Id})$. By induction, one can show that for any t there exists $E_t \sim \mathcal{N}(0,\mathsf{Id})$ such that

$$X_t = \sqrt{\overline{\alpha}_t} X_0 + \sqrt{\overline{\beta}_t} E_t, \tag{2}$$

where $(\overline{\alpha}_t), (\overline{\beta}_t)$ are defined by $\overline{\alpha}_0 = 1, \overline{\beta}_0 = 0$ and

$$\overline{\alpha}_t = \overline{\alpha}_{t-1}\alpha_t, \quad \text{and} \quad \overline{\beta}_t = \alpha_t \overline{\beta}_{t-1} + \beta_t. \tag{3}$$

The noise schedule is called "variance-preserving" if $\alpha_t + \beta_t = 1$ (which implies $\overline{\alpha}_t + \overline{\beta}_t = 1$ for all t), but we do

not restrict to this choice here. Gaussian conditioning gives

$$\mathbb{E}[X_{t-1}|X_t, X_0] = \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t X_0 + \sqrt{\alpha_t}\overline{\beta}_{t-1}X_t}{\overline{\beta}_t}, \quad (4)$$

$$\mathsf{Var}[X_{t-1}|X_0, X_t] = \frac{\overline{\beta}_{t-1}}{\overline{\beta}_t}\beta_t =: \tilde{\beta}_t, \quad (5)$$

which, after manipulation, also gives

$$\mathbb{E}[X_{t-1}|X_t, X_0] := \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{\beta_t}{\overline{\beta}_t}(X_t - \sqrt{\overline{\alpha}_t}X_0)\right). \quad (6)$$

Without knowing X_0, Tweedie formula [3] gives the MMSE prediction of X_{t-1}, or X_0 knowing only X_t for $t \in [1:T]$:

$$\mathbb{E}[X_{t-1}|X_t] = \frac{1}{\sqrt{\alpha_t}}(X_t + \beta_t \nabla \log q_t(X_t)), \quad (7)$$

$$\mathbb{E}[X_0|X_t] = \frac{1}{\sqrt{\overline{\alpha}_t}}(X_t + \overline{\beta}_t \nabla \log q_t(X_t)), \quad (8)$$

where q_t is the probability density of X_t.

The DDPM backward process starts from a random image $\mathbf{X}_T$ (often Gaussian) and applies backward transitions

$$\mathbf{X}_{t-1} = \mu_\theta^t(\mathbf{X}_t) + \sqrt{\tilde{\beta}_t}\mathbf{Z}_{t-1}, \text{ with } \tilde{\beta}_t = \frac{\overline{\beta}_{t-1}}{\overline{\beta}_t}\beta_t, \quad (9)$$

and where $(\mathbf{Z}_t)$ are i.i.d. $\mathcal{N}(0, \mathsf{Id})$, and

$$\mu_\theta^t(x) := \frac{1}{\sqrt{\alpha_t}}\left(x - \frac{\beta_t}{\sqrt{\overline{\beta}_t}}\varepsilon_\theta^t(x)\right) \approx \mathbb{E}[X_{t-1}|X_t = x] \quad (10)$$

is parameterized by a neural network ε_θ^t, and is trained to approximate the MMSE predictor (7). One can show [17] that the approximation $\varepsilon_\theta^t \approx -\sqrt{\overline{\beta}_t}\nabla \log q_t$ (for $L^2(q_t)$ distance) of the Stein score $\nabla \log q_t$ is equivalent to

$$\operatorname*{Argmin}_\theta \mathbb{E}[\|X_{t-1} - \mu_\theta^t(X_t)\|_2^2]. \quad (11)$$

Similarly, it is equivalent to ask that

$$\frac{1}{\sqrt{\overline{\alpha}_t}}\left(x - \sqrt{\overline{\beta}_t}\varepsilon_\theta^t(x)\right) \approx \mathbb{E}[X_0|X_t = x] \quad (12)$$

approximates well the best predictor (8) of X_0 given $X_t = x$.

2.2 DDPM as "Noising-Relaxed Denoising"

In this section, we show how to interpret the DDPM scheme only in terms of additive Gaussian noise. This allows to derive a simple noise preservation rule, which guides the choice of backward noise schedule. This connects with the "variance-exploding" formulation of [14].

For $\sigma > 0$, let $U_0 + \sigma Z$ where $U_0 \sim q_0$ and $Z \sim \mathcal{N}(0, \mathsf{Id})$ are independent. We consider a denoiser $D_\sigma : \mathbf{R}^d \to \mathbf{R}^d$ (which may be parameterized by a neural network) that approximates the minimum mean-square error (MMSE) denoiser $D_\sigma^*(u) = \mathbb{E}[U_0|U_0 + \sigma Z = u]$. For $\rho \in [0, 1]$ we introduce the relaxed denoiser

$$D_\sigma^\rho = (1-\rho)\mathsf{Id} + \rho D_\sigma. \quad (13)$$

By setting $U_t = \frac{X_t}{\sqrt{\overline{\alpha}_t}}$, (2) writes as

$$\boxed{U_t = U_0 + \sigma_t E_t, \quad \text{with} \quad \sigma_t = \sqrt{\frac{\overline{\beta}_t}{\overline{\alpha}_t}}.} \quad (14)$$

The prediction (12) expresses as a Gaussian denoiser

$$\boxed{D_{\sigma_t}(u) = u - \sigma_t \epsilon_\theta^t(\sqrt{\overline{\alpha}_t}\, u),} \quad (15)$$

which is trained to approximate the MMSE denoiser

$$D_{\sigma_t}^*(u) = \mathbb{E}[U_0|U_t = u]. \quad (16)$$

The backward DDPM scheme then translates on $\mathbf{U}_t := \frac{\mathbf{X}_t}{\sqrt{\overline{\alpha}_t}}$ as

$$\mathbf{U}_{t-1} = \frac{\mu_\theta^t(\sqrt{\overline{\alpha}_t}\mathbf{U}_t)}{\sqrt{\overline{\alpha}_{t-1}}} + \sqrt{\frac{\tilde{\beta}_t}{\overline{\alpha}_{t-1}}}\,\mathbf{Z}_{t-1}, \quad (17)$$

which, after simplification, gives

$$\mathbf{U}_{t-1} = \left(1 - \frac{\beta_t}{\overline{\beta}_t}\right)\mathbf{U}_t + \frac{\beta_t}{\overline{\beta}_t}D_{\sigma_t}(\mathbf{U}_t) + \sqrt{\frac{\tilde{\beta}_t}{\overline{\alpha}_{t-1}}}\,\mathbf{Z}_{t-1}. \quad (18)$$

Setting $\rho_t = \frac{\beta_t}{\overline{\beta}_t}$ and $\gamma_{t-1} = \sqrt{\frac{\tilde{\beta}_t}{\overline{\alpha}_{t-1}}}$, this backward scheme can be interpreted as iterative "relaxed denoising-noising":

$$\forall t \in [1:T], \quad \boxed{\mathbf{U}_{t-1} = D_{\sigma_t}^{\rho_t}(\mathbf{U}_t) + \gamma_{t-1}\mathbf{Z}_{t-1}.} \quad (19)$$

The parameters $\alpha_t, \beta_t, \tilde{\beta}_t$ can be related to $\sigma_t, \rho_t, \gamma_t$, and their tuning should ensure that the noise level of $\mathbf{U}_t$ stays close to σ_t all along the backward process. This reflects in a variance rule, which was already given in [15]. Assuming that $\mathbf{U}_t$ has noise level σ_t and that the remaining portion of noise in $D_{\sigma_t}^{\rho_t}(\mathbf{U}_t)$ has level $(1-\rho_t)\sigma_t$, then considering the noise variance on both sides of (19) gives

$$\boxed{\sigma_{t-1}^2 = (1-\rho_t)^2\sigma_t^2 + \gamma_{t-1}^2.} \quad (20)$$

One can check that the $\tilde{\beta}_t$ choice of (9) (as advised in [8]) indeed makes this equality true. Indeed, the relation (3) can be written in terms of σ_t, ρ_t as

$$\sigma_t^2 = \sigma_{t-1}^2 + \rho_t\sigma_t^2 \quad \text{i.e.} \quad \rho_t = \frac{\sigma_t^2 - \sigma_{t-1}^2}{\sigma_t^2}. \quad (21)$$

2.3 Iterative Noising-Relaxed Denoising

Based on the previous derivations, we here propose to consider diffusion schemes as iterative noising-denoising. Let $(\sigma_t)_{t\in[0:T+1]} \in \mathbf{R}_+^{T+1}$, $(\rho_t)_{t\in[1:T+1]} \in [0,1]^{T+1}$ be such that $\forall t \in [1:T]$, $\gamma_t^2 := \sigma_t^2 - (1-\rho_{t+1})^2\sigma_{t+1}^2 \geq 0$.

We define iterative noising-relaxed denoising (INRD) as follows. Starting from a random $\mathbf{V}_T$, for $t = T, \dots, 1$, we let

$$\boxed{\mathbf{V}_{t-1} = D_{\sigma_t}^{\rho_t}(\mathbf{V}_t + \gamma_t\mathbf{Z}_t).} \quad \text{(INRD)}$$

By convention, $\mathbf{V}_T$ has noise level $(1-\rho_{T+1})\sigma_{T+1}$ so that $\mathbf{V}_T + \gamma_T\mathbf{Z}_T$ can be denoised at level σ_T. The case $\rho_t = 1$ gives iterative noising-denoising (IND):

$$\boxed{\mathbf{V}_{t-1} = D_{\sigma_t}(\mathbf{V}_t + \sigma_t\mathbf{Z}_t).} \quad \text{(IND)}$$

Generated images are then obtained as samples of $\mathbf{V}_0$. INRD connects to the schemes of Section 2.2 with $\mathbf{V}_{t-1} = D^{\rho_t}_{\sigma_t}(\mathbf{U}_t)$.

These schemes depend in an obvious way on the noise schedule (σ_t) that controls the level of additive Gaussian noise at each step. Since most of the literature on image denoising focuses on additive Gaussian noise, we claim that this formulation is much easier to understand for image processing practicioners. Current iterates $\mathbf{V}_t$ have noise level 0 and can be displayed as images. INRD generalizes the DDPM scheme, by allowing for example $\rho_t = 1$ which leads to IND. If (σ_t, ρ_t) satisfies (21) with $\rho_t \in (0, 1)$ (in particular σ_t increases), then INRD induces a DDPM-like scheme. For any non-decreasing noise schedule (σ_t), (21) gives a canonical choice of relaxation (ρ_t). But INRD is defined more generally with the only constraint that we denoise sufficiently at each iteration: $\rho_t \geq 1 - \frac{\sigma_{t-1}}{\sigma_t}$. Notice that this condition is met with the canonical relaxation $\rho_t = 1 - \frac{\sigma^2_{t-1}}{\sigma^2_t}$ as soon as $\sigma_{t-1} \leq \sigma_t$.

3 Experiments

3.1 Faster Diffusion Schemes

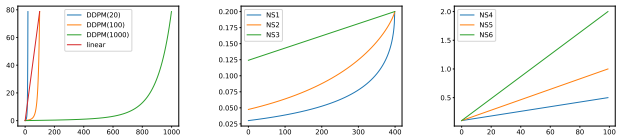
The generative quality of a diffusion scheme depends crucially on the chosen schedules (σ_t, ρ_t). There is obviously a trade-off between sufficient denoising (which produces geometric structures as in the images of the dataset) and sufficient noising (which authorizes random innovation at each step). On Fig. 1 we illustrate several possible noise schedules: the DDPM noise schedule of [8], and also super-geometric and affine schedules. For DDPM we initialize at $\mathbf{V}_T = 0.5$. It is interesting to notice that the DDPM noise schedule starts around $80\times$ the image range.

For all experiments shown in this paper, we used the canonical relaxation $\rho_t = \frac{\sigma^2_t - \sigma^2_{t-1}}{\sigma^2_t}$ (as in standard DDPM). With this choice, we only need to fix a noise schedule (σ_t) adapted to the denoiser. In particular, this allows for very simple pruning of pre-learned diffusion models, for example using DDPM on a time subgrid (using $\sigma'_t = \sigma_{at}$ for $a \in \mathbb{N}^*$).

On Fig. 2, we show samples of IND and INRD based on DiffUNet. Surprisingly convincing samples can be obtained with the simpler algorithm IND. While samples of IND seem smoother, the created geometric stuctures do not appear less relevant compared to INRD(1000) (which is the standard DDPM). Also, both IND and INRD produce convincing samples, even for 20 steps ($50\times$ less than DDPM).

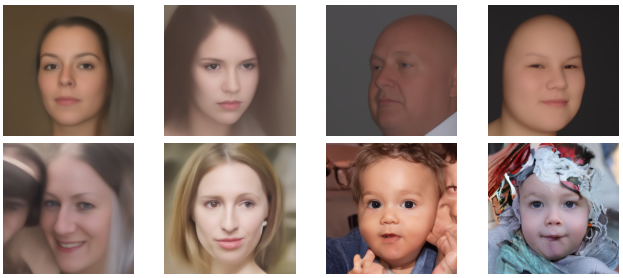
3.2 Diffusion with pre-learned denoisers

Since (INRD) expresses simply through a Gaussian denoiser, it can be directly tested with off-the-shelf denoisers. On Fig. 3, we show samples obtained with INRD with DRUNet and total variation denoiser $\mathsf{TVD}_\lambda(u) = \mathsf{Argmin}_v \frac{1}{2}\|u-v\|_2^2 + \lambda \mathsf{TV}(v)$. The scaling parameter was here (abusively) set to $\lambda = \sigma$. We rely on implementations of the DeepInverse library [1]. For DRUNet, we used noise schedules adapted to the noise range $[0, 0.2]$ that was used for the denoiser training. Also, for DRUNet, we initialize $\mathbf{V}_T \sim \mathcal{N}(0, s_T^2 \mathsf{Id})$ with $s_T \gg \sigma_T$ in an ad-hoc manner; otherwise the process degenerates immediately to the mean value, and no geometric structures appear.



DiffUNet DRUNet TV

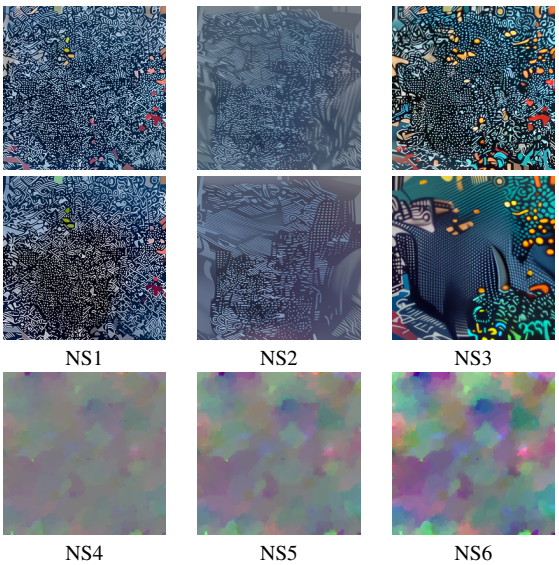
Figure 1 – Noise schedules



Linear(100) DDPM(20) DDPM(100) DDPM(1000)

Figure 2 – Samples of IND (1st row) and INRD (2nd row) with DiffUNet. Noise schedules are shown on Fig. 1. For each column we indicate the number of time steps.

Our experiments indicate that, while some widely adopted noise schedules produce good samples with the pre-learned DiffUNet, they may not be adapted to other denoisers. Our experiments highlighted stability problems when using INRD with DRUNet. For this reason, in Fig. 3 we only show intermediate samples of this INRD-DRUNet. In contrast, we did not observe such stability problems for INRD based on DiffUNet or TVD. These experiments therefore confirm that finding the right noise schedule (σ_t) requires some careful ad-hoc tuning, in a way that depends on the chosen denoiser. However, contrary to what is apparent in several papers on the subject, the success of a noise schedule appears to us as completely independent of any rationale used to inspire the sampling scheme, beause the learning phase of standard diffusion models relies only on the approximation of the MMSE denoiser for all noise levels (σ_t).



NS1 NS2 NS3

NS4 NS5 NS6

Figure 3 – Samples of IND-DRUNet (Rows 1,2) and INRD-TV (Row 3) with noise schedules of Fig. 1. For DRUNet, we show samples after 60 and 120 iterations, out of 400 iterations.

3.3 Plug-and-Play Imaging

This last section shows that, in the other way around, diffusion denoisers like DiffUNet can be used in PnP schemes in order to adress linear imaging inverse problems. Here we consider a deblurring problem $y = Av_c + w$ where $y \in \mathbf{R}^m$ is the degraded observation, $v_c \in \mathbf{R}^d$ is a clean image, $A \in \mathbf{R}^{m\times d}$ is a periodic convolution and $w \sim \mathcal{N}(0, \nu^2\mathsf{Id})$ with $\nu = 0.01$. The corresponding data-fidelity writes as $f(v) = \frac{1}{2\nu^2}\|Av - y\|_2^2$. We will focus only on the PnP-PGD algorithm, which consists in the iterations $v_{n+1} = D_\sigma \circ (\mathsf{Id} - \tau\nabla f)(v_n)$. As already noticed in [4], relation (15) allows to use the pre-learned DiffUNet denoiser in this PnP scheme. Contrary to diffusion schemes, most PnP schemes require to use the denoiser only for low noise levels, and here we will show comparisons with the smallest noise level $\sigma_0 = \sqrt{\frac{\bar\beta_0}{\bar\alpha_0}} \approx 0.005$ accessible for DiffUNet.

On Fig. 4 we display several results obtained with this PnP-PGD algorithm based on DRUNet and DiffUNet on images taken from the FFHQ validation set. Our main observations are as follows. First, with this raw (non-convergent) PnP-PGD scheme, DiffUNet does not offer sufficient implicit regularization to solve this deblurring problem in a stable way: numerical artifacts may appear, and not only on the image border. This often reflects in a clear PSNR drop and residual $v_{n+1} - v_n$ not going to 0. Second, while DRUNet tends to oversmooth most regions, other oscillatory (non-numerical) artifacts may be left by DiffUNet, especially subtle textural patterns in homogeneous regions (visible only with close look, and not accounted for by PSNR). Therefore, if one looks for a good compromise between performance, stability and computational cost when using this precise PnP-PGD scheme, one should definitely favor the lighter network DRUNet. Indeed, light networks appear sufficient to regularize inverse problems that are not severely ill-posed (especially in contexts where the restoration algorithm should not hallucinate content), and some of their stability issues can be settled by turning to convergent variants of PnP algorithms [11] instead of raw PnP-PGD with uncontrolled denoiser.

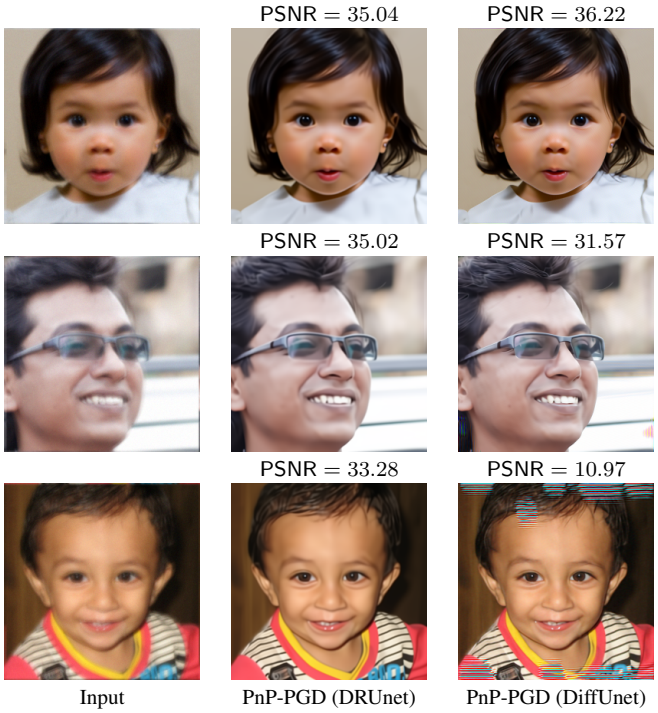


Figure 4 – Results of PnP-PGD deblurring based on DRUNet and DiffUnet (please zoom in for close inspection).

Acknowledgments. The authors acknowledge support of the French National Research Agency through Projects ANR-19-CE40-005 MISTIC and ANR-23-CE40-0017 SOCOT.

References

[1] DeepInverse: A deep learning framework for inverse problems in imaging, 2023. https://deepinv.github.io/deepinv/.

[2] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, volume 34, pages 8780–8794, 2021.

[3] B. Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

[4] C.Y. Park et al. Plug-and-play priors as a score-based method. *arXiv preprint arXiv:2412.11108*, 2024.

[5] E.C. Faye et al. Regularization by denoising: Bayesian model and Langevin-within-split Gibbs sampling. *IEEE Transactions on Image Processing*, 2024.

[6] F. Coeurdoux et al. Plug-and-play split Gibbs sampler: embedding deep generative priors in Bayesian inference. *IEEE Transactions on Image Processing*, 2024.

[7] H. Chung et al. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023.

[8] J. Ho et al. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pages 6840–6851, 2020.

[9] K. Zhang et al. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[10] L.I. Rudin et al. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[11] S. Hurault et al. Gradient step denoiser for convergent plug-and-play. In *ICLR*, 2021.

[12] S.V. Venkatakrishnan et al. Plug-and-play priors for model based reconstruction. In *GCSIP*, pages 945–948. IEEE, 2013.

[13] Y. Romano et al. The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[14] Y. Song et al. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[15] Z. Kadkhodaie and E.P. Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. In *NeurIPS*, 2020.

[16] S. Mallat. *A Wavelet Tour of Signal Processing, The Sparse Way*. Elsevier, 2009.

[17] P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.