A PLUG-AND-PLAY DEEP IMAGE PRIOR

Zhaodong Sun, Fabian Latorre, Thomas Sanchez, and Volkan Cevher

École Polytechnique Fédérale de Lausanne (EPFL)

ABSTRACT

Deep image priors (DIP) offer a novel approach for the regularization that leverages the inductive bias of a deep convolutional architecture in inverse problems. However, the quality of DIP approaches often degrades when the number of iterations exceeds a certain threshold due to overfitting. To mitigate this effect, this work incorporates a plug-and-play prior scheme which can accommodate additional regularization steps within a DIP framework. Our modification is achieved using an augmented Lagrangian formulation of the problem, and is solved using an Alternating Direction Method of Multipliers (ADMM) variant, which can capture existing DIP approaches as a special case. We show experimentally that our ADMM-based DIP pairing outperforms competitive baselines in PSNR while exhibiting less overfitting.

Index Terms— Deep Image Prior, Plug-and-Play Prior, Alternating Direction Method of Multipliers (ADMM), Inverse Problem, Overfitting

1. INTRODUCTION

Inverse problems revolve around retrieving an unknown signal or image $\boldsymbol{x}^\star$ from undersampled or corrupted measurements. As such problems are ill-posed, a key ingredient for a successful recovery is the choice of regularization, or, in Bayesian terms, the choice of the prior. Classical compressive sensing methods [1] traditionally resorted to statistical priors such as the ℓ_1-norm and TV-norm (total variation) [2] and they enjoy recovery guarantees under sparsity assumptions. However, these simple regularizers cannot fully capture the true distribution of natural images, and perform poorly when only a few measurements are available.

Recently, deep learning-based methods have recently achieved state-of-the-art performance for a host of inverse problems [3, 4]. Their success has been attributed to their ability to incorporate data-driven priors, at the cost of losing formal recovery guarantees. One of the driving forces behind this paradigm change has been the development of Generative Adversarial Networks (GANs) [5], which have been used to successfully perform Compressive Sensing, known as Compressed Sensing using Generative Models (CSGM) [6] even when few measurements are available. However, training such a model usually requires a large dataset, which hinders their applicability in domains where data is scarce.

Deep Image Prior (DIP) [7] is a different approach, conceptually close to CSGM [6] but different in a crucial way: it leverages the inductive bias of an untrained convolutional neural network architecture as the prior, removing the dependency on large datasets. The reconstruction quality of DIP [7] is comparable with some state-of-the-art data-driven methods. However, early stopping is necessary as a large number of iterations degrades the performance as the DIP approach can also overfit the noisy measurements.

Several papers attempt to resolve this issue [8, 9, 10]. Stochastic Gradient Langevin Dynamics (SGLD) is used to obtain posterior samples[8]; an averaging scheme can be used to reduce overfitting. The *Deep Decoder* architecture [9] mitigates overfitting by limiting the number of trainable parameters. Finally, [10] proposes to use Stein's unbiased risk estimator (SURE) as the loss function, which can be regarded as a surrogate to the generalization error. Since the loss function can be seen as the generalization loss, their experiments confirm that this method can mitigate overfitting.

In this work, we propose a new method to incorporate plug-and-play (PnP) priors [11] with deep image prior [7]. PnP has shown strong empirical results and is not susceptible to overfitting like DIP. Our modification is achieved using an augmented Lagrangian formulation of the problem, and is solved using an Alternating Direction Method of Multipliers (ADMM) variant, which can capture existing DIP approaches as a special case. From numerical experiments, we show that this hybrid prior method can avoid overfitting of DIP, thus improving the performance. Our code is available at `https://github.com/zhaodongsun/pnp_dip`.

2. BACKGROUND

In linear inverse problems, we are given a reference signal $\boldsymbol{x}^\star \in \mathbb{R}^n$ and noisy measurements $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{\epsilon}$ with Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{R}^m$. $\boldsymbol{A}$ is the measurement matrix with shape $m \times n$ ($m \ll n$), and can correspond to a downsampling operator (super-resolution task), random pixel removal (inpainting), Gaussian linear measurements (compressive sensing), among many others. In this setting, we focus on the following optimization problem:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + R(\boldsymbol{x}), \tag{1}$$

where $R(\boldsymbol{x})$ is the prior or *regularization* term.

2.1. Deep Image Prior (DIP)

DIP [7] follows the generic template of Problem 1. However, the prior $R(\boldsymbol{x})$ is replaced with the constraint for $\boldsymbol{x}$ to lie in the range of a convolutional neural network (CNN), $G(\boldsymbol{z};\theta)$:

$$\begin{aligned} &\min_{\boldsymbol{x},\theta} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \\ &\text{s.t. } \boldsymbol{x} = G(\boldsymbol{z};\theta) \end{aligned} \tag{2}$$

This optimization formulation can be rewritten as follows:

$$\min_{\theta} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}G(\boldsymbol{z};\theta)\|_2^2. \tag{3}$$

We then see that the problem can be solved by optimizing over the weights of the generator $G(\boldsymbol{z};\theta)$. Note that while we can also optimize over the latent $\boldsymbol{z}$ variable, it does not improve practical results significantly. As a result, like in most works, the variable $\boldsymbol{z}$ is initialized randomly and then is kept fixed in the sequel.

2.2. Plug-and-play Prior (PnP)

Plug-and-play prior [11] is a flexible framework that incorporates state-of-the-art denoising algorithms as priors in more challenging inverse problems. The idea is to reformulate Problem 1 to enable the usage of denoising algorithms that can not be expressed in the form of an optimization problem. The first step is to split the optimization variables to $\boldsymbol{x}$ and $\boldsymbol{y}$

$$\begin{aligned} &\min_{\boldsymbol{y},\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - A\boldsymbol{y}\|_2^2 + R(\boldsymbol{x}) \\ &\text{s.t. } \boldsymbol{x} = \boldsymbol{y} \end{aligned} \tag{4}$$

Since both two splitted variables are in two separated terms, they use alternating direction method-of-multipliers (ADMM) [12] to solve Problem 4. If we define $H(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{b} - A\boldsymbol{y}\|_2^2$, the augmented Lagrangian with scaled dual variable u and penalty parameter ρ is

$$\mathcal{L}_\rho(\boldsymbol{x},\boldsymbol{y},\boldsymbol{u}) = H(\boldsymbol{y}) + R(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{u}\|_2^2 - \frac{\rho}{2}\|\boldsymbol{u}\|_2^2 \tag{5}$$

The ADMM steps for each variable of the augmented Lagrangian then read

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \operatorname{prox}_{\frac{R}{\rho}}(\boldsymbol{y}_t - \boldsymbol{u}_t) \\ \boldsymbol{y}_{t+1} &= \operatorname{prox}_{\frac{H}{\rho}}(\boldsymbol{x}_{t+1} + \boldsymbol{u}_t) \\ \boldsymbol{u}_{t+1} &= \boldsymbol{u}_t + (\boldsymbol{x}_{t+1} - \boldsymbol{y}_{t+1}) \end{aligned} \tag{6}$$

The main contribution of PnP is replacing the proximal step for $\boldsymbol{x}$ with a state-of-the-art denoising algorithm like BM3D [13], improving the overall performance.

3. PROPOSED METHOD

We propose to combine deep image prior [7] with plug-and-play priors [11] in a principled fashion. Our goal is to limit the overfitting issues observed when using DIP, improving the quality of the result. We first reformulate (2) as follows

$$\begin{aligned} &\min_{\theta,\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}G(\boldsymbol{z};\theta)\|_2^2 + R(\boldsymbol{x}) \\ &\text{s.t. } \boldsymbol{x} = G(\boldsymbol{z};\theta) \end{aligned} \tag{7}$$

We simply substitute $\boldsymbol{x}$ with $G(\boldsymbol{z};\theta)$ in the data fidelity term at equation 2 and further incorporate the regularizer $R(\boldsymbol{x})$. From another perspective, the variable $\boldsymbol{y}$ in the plug-and-play prior problem at Equation 4 is replaced with a network $G(\boldsymbol{z};\theta)$. The formulation obtained it then a hybrid version of both DIP [7] and plug-and-play priors [11]. To solve this new problem, we first write the augmented Lagrangian with scaled dual variable $\boldsymbol{u}$ and penalty parameter ρ.

$$\begin{aligned} \mathcal{L}_\rho(\boldsymbol{x},\theta,\boldsymbol{u}) = &\frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}G(\boldsymbol{z};\theta)\|_2^2 + R(\boldsymbol{x}) \\ &+ \frac{\rho}{2}\|\boldsymbol{x} - G(\boldsymbol{z};\theta) + \boldsymbol{u}\|_2^2 - \frac{\rho}{2}\|\boldsymbol{u}\|_2^2 \end{aligned} \tag{8}$$

The corresponding ADMM steps for this augmented Lagrangian are

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \operatorname{prox}_{\frac{R}{\rho}}(G(\boldsymbol{z};\theta_t) - \boldsymbol{u}_t) \\ \theta_{t+1} &= \operatorname*{argmin}_{\theta} \mathcal{L}_\rho(\boldsymbol{x}_{t+1},\theta,\boldsymbol{u}_t) \\ \boldsymbol{u}_{t+1} &= \boldsymbol{u}_t + \big(\boldsymbol{x}_{t+1} - G(\boldsymbol{z};\theta_{t+1})\big) \end{aligned} \tag{9}$$

This augmented Lagrangian with respect to θ is non-convex, which means the minimization over θ is often intractable. Similar to [14], we take one gradient descent step for θ. Indeed, in some convex problems where this exact minimization is intractable, convergence to a solution is still guaranteed after modifying ADMM in this way [15]. In our experiments, this simple first-order update is enough to obtain good solutions. Our method is summarized as Algorithm 1.

Note that the only way in which the prior changes the behaviour of the algorithm is through its proximal operator (line 2 in Algorithm 1). For some priors, such as total variation or ℓ_1-norm, the proximal operator can understood as a denoising algorithm. This motivates us to replace the proximal operator step with other denoising algorithms like BM3D [13], in a similar fashion to PnP [11].

3.1. Relation to Prior Works

Our method (PnP-DIP) is a combination of DIP [7] and plug-and-play prior [11]. It can be seen as a variation of [14], which combines a pre-trained GAN with a prior $R(\boldsymbol{x})$ and solves the problem by ADMM. The main difference is that our method does not require any data, providing an advantage in practical scenarios. Our proposed method generalizes the method of van Veen et al. [16], where they propose a total variation plus a learned regularization for Problem 1 and use gradient descent to solve it.

Proc. ICASSP 2021, June 06-11, 2021, Toronto, Canada. DOI: 10.1109/ICASSP39728.2021.9414879

DeepRED [17] is perhaps the closest in essence to our work. They also leverage ADMM to fuse DIP together with an additional prior. However we differ in two important ways. First, the objective function they consider has a specific form of regularization with a differentiability assumption and hence, it cannot work with some common non-smooth priors such as the ℓ_∞-norm, ℓ_1-norm or total variation norm. For example, their method cannot handle the case where the prior is a constraint like in the denoising task presented in Section 4.1.2). In contrast, our method can incorporate these non-smooth priors via the proximal operator in $\boldsymbol{x}$ step.

A second crucial difference is that their ADMM variant has a double-loop structure: The minimization steps over $\boldsymbol{x}$ and $\boldsymbol{y}$ are performed approximately with iterative gradient steps or fixed point iterations. In sharp contrast, our proposed Algorithm 1 requires only single applications of the proximal operator and a single gradient descent update, eliminating the extra complexity of the double-loop of DeepRED [17]. The time complexity of Algorithm 1 will nonetheless be tied to the cost of applying the denoising algorithm, evaluating the neural network G and optimizing the weights of G through backpropagation.

Algorithm 1 Plug-and-play DIP

Input: Augmented Lagrangian: $\mathcal{L}_\rho$, Number of iterations: T, Regularizer: R, Convolutional Neural Network: G, Primal initialization: $\theta_0, \boldsymbol{x}_0$, Dual initialization: $\boldsymbol{u}_0$, Step size for θ: β, Optimizer for θ: Opt

for $t = 0, 1, ..., T-1$ **do**
  $\boldsymbol{x}_{t+1} \leftarrow \operatorname{prox}_{\frac{R}{\rho}}(G(\boldsymbol{z};\theta_t) - \boldsymbol{u}_t)$ ▷ Update $\boldsymbol{x}$
  $g_\theta \leftarrow \nabla_\theta \mathcal{L}_\rho(\boldsymbol{x}_{t+1}, \theta, \boldsymbol{u}_t)$ ▷ Gradient w.r.t. θ
  $\theta_{t+1} \leftarrow \theta_t - \beta \cdot \text{Opt}(\theta_t, g_\theta)$ ▷ Update θ
  $\boldsymbol{u}_{t+1} \leftarrow \boldsymbol{u}_t + \big(\boldsymbol{x}_{t+1} - G(\boldsymbol{z};\theta_{t+1})\big)$ ▷ Dual update
end for

4. EXPERIMENTS AND DISCUSSION

4.1. Implementation Details

4.1.1. Inpainting and super-resolution

We use the CelebA dataset [18] to test our algorithm. All the images are resized to $128 \times 128 \times 3$ (49152 pixels). For inpainting, The measurement $\boldsymbol{b}$ is obtained by randomly choosing half of the pixels, and adding Gaussian noise with zero mean and standard deviation $\sigma = 10/255$ (the image range is from 0 to 1). For $2\times$ super-resolution, the measurement $\boldsymbol{b}$ is obtained by sampling every other pixel along height and width of the image, with added Gaussian noise.

We use BM3D[13], NLM[19], and TV denoising algorithms as proximal operators. 10 images from CelebA [18] are randomly chosen to tune the parameters in the proximal operator (or denoising algorithm). After tuning, we choose the shrinkage parameter 0.01 for the TV proximal operator. For the BM3D and NLM denoising algorithms, we choose the standard deviation of the noise as 0.05 and 0.01, respectively. Other parameters of NLM are set as follows: Patch distance 2, cut-off distance 0.05. For detailed description of these parameters, please refer to the implementations of these denoising algorithms[1].

We test our algorithm on another 25 images from the CelebA dataset [18]. In our algorithm, when updating θ, we use the ADAM optimizer [20] with learning rate 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The penalty parameter ρ in the augmented Lagrangian is 1. For the choice of the convolutional neural network, we use a similar encoder-decoder architecture to the one used in [7]. The parameters of the network are $n_u = n_d = [16, 32, 64, 128, 128], n_s = [0, 0, 0, 0, 0], k_d = k_u = 3$. For the explanation of these parameters, please refer to the supplementary material in [7]. We also use a moving average to smooth the reconstructed images for better reconstruction and smooth PSNR curves, similar to [8].

We also implemented the original DIP [7], plug-and-play priors [11] and DIP+TV [16, 21] as baselines. For DIP [7], we use the same network architecture and ADAM optimizer [20] with the same parameters as our proposed method. For the implementation of plug-and-play priors [11], the parameters that we need to determine are the penalty parameters ρ and the parameters inside the denoising algorithms (see equation 6 for the parameters). All these parameters are the same as our proposed method above. For DIP+TV [16, 21], we use the same ADAM optimizer [20] and the shrinkage parameter of TV norm is 0.01, which is the same as our previous setting in our method. Note that the learned prior mentioned in [16] is not used.

4.1.2. Denoising

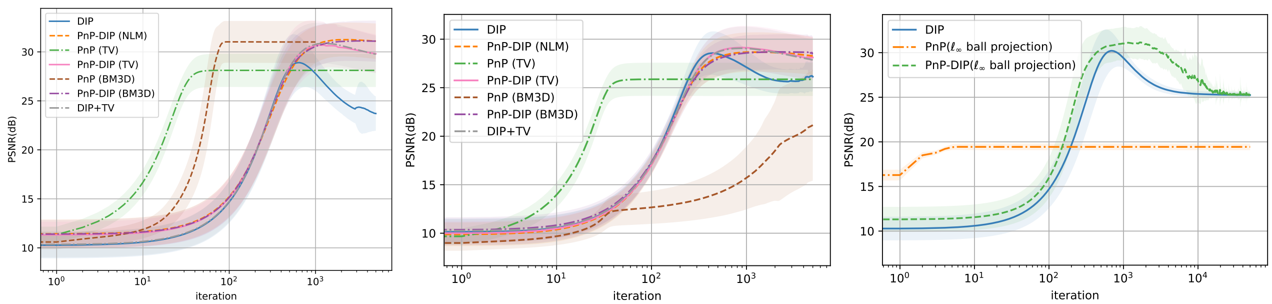
We choose the measurement matrix $\boldsymbol{A}$ as the identity matrix, and we add uniform noise $\epsilon \sim \mathcal{U}[-a, a]$. Since uniform noise is bounded, we have the prior knowledge that the solution $\boldsymbol{x}$ should satisfy $\{\boldsymbol{x} | \|\boldsymbol{x} - \boldsymbol{b}\|_\infty \leqslant a\}$, which is also illustrated at Chapter 7.1.1 in [22]. The proximal operator in $\boldsymbol{x}$ step of our method corresponds to the projection onto this ℓ_∞ ball $\{\boldsymbol{x} | \|\boldsymbol{x} - \boldsymbol{b}\|_\infty \leqslant a\}$. We add the uniform noise $\epsilon \sim \mathcal{U}[-\frac{50}{255}, \frac{50}{255}]$. Other settings are the same as in the inpainting task.

4.2. Discussion on the results

4.2.1. Inpainting and Super-resolution

For these two tasks, we compare four methods, which are our proposed plug-and-play deep image prior (PnP-DIP), plug-and-play prior (PnP) [11], deep image prior (DIP) [7] and

[1] **BM3D:** `http://www.cs.tut.fi/~foi/GCF-BM3D/`, **NLM:** skimage.restoration.denoise_nl_means at `https://scikit-image.org/docs/dev/api/skimage.restoration.html`, **TV:** `https://github.com/albarji/proxTV`



(a) inpainting (b) super-resolution (c) denoising

Fig. 1. PSNR with respect to iterations

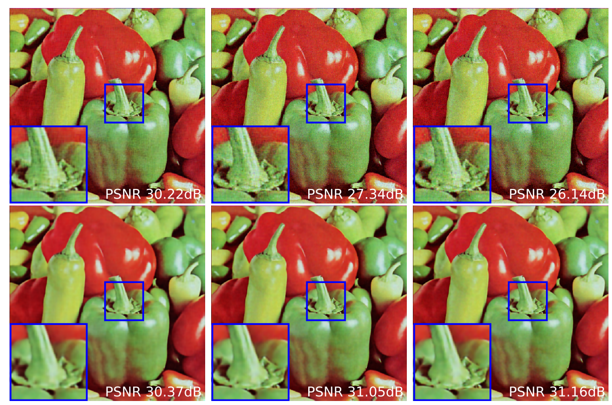
Method	PSNR (dB) for inpainting	PSNR (dB) for super-resolution
DIP [7]	23.70 ± 1.75	26.14 ± 1.51
DIP+TV [16, 21]	29.76 ± 1.90	27.87 ± 2.10
PnP (TV) [11]	28.11 ± 1.69	25.86 ± 1.71
PnP (BM3D) [11]	31.01 ± 2.14	21.15 ± 5.73
PnP-DIP (TV)	29.81 ± 1.97	28.12 ± 2.18
PnP-DIP (BM3D)	$\mathbf{31.10 \pm 1.81}$	$\mathbf{28.46 \pm 1.81}$
PnP-DIP (NLM)	31.09 ± 2.09	28.26 ± 1.93

Table 1. Final PSNR at Figure 1 for inpainting and super-resolution (The largest PSNR are in bold.)

DIP+TV [16, 21]. For PnP-DIP and PnP, they can combine denoising algorithms in their framework, and we try different options of denoising algorithms for comparison. Since we test these algorithms on 25 images, we get 25 PSNR curves with respect to iterations. In Figure 1, we show the mean PSNR curve with respect to iterations for each method for the 25 images. The shade represents the standard deviation. Table 1 shows the PSNR in the last iteration at Figure 1 (a) and (b).

From the results, we observe that deep image prior has the overfitting problem since it will first reach the peak of PSNR and then gradually drop. After using our hybrid prior method called PnP-DIP, the overfitting phenomenon is mitigated or almost disappears. Our proposed PnP-DIP also has the best PSNR compared with DIP or PnP alone. However, we also observe that the denoising algorithm will increase the running time for each iteration. For our PnP-DIP with TV and NLM, the running time is twice the one of DIP. However, results in the method being BM3D 200 times slower than DIP alone. There is an inherent trade-off between speed and performance depending on the denoiser.

We test our method on the *pepper* image to visually illustrate the effect of our method in the inpainting. For DIP in Figure 2 (top row), we can observe that the PSNR decreases and the image becomes noisy, which is the indication of DIP overfitting. For our PnP-DIP (NLM) in Figure 2 (bottom row), the PSNR increases and the image becomes clear.



Iteration 1000 Iteration 3000 Iteration 5000

Fig. 2. Visual Results of DIP (top row) and PnP-DIP (bottom row) at different iterations in the inpainting task

The visual results demonstrate that our PnP-DIP method can improve the performance of DIP and limit overfitting.

4.2.2. *Denoising*

We show the PSNR curves for the denoising task in Figure 1 (c). Plug-and-play prior (PnP) with ℓ_∞ ball projection fails to do denoising task. Perhaps the ℓ_∞ ball projection alone cannot provide enough prior knowledge for the denoising task. However, Our proposed PnP-DIP with ℓ_∞ ball projection still works well. With this weak prior knowledge, PnP-DIP with ℓ_∞ ball projection surpasses DIP in almost all iterations and the overfitting is alleviated. This demonstrates that our PnP-DIP method can achieve better performance by combining DIP with prior knowledge that DIP does not have.

5. ACKNOWLEDGEMENTS

This work was funded (in part) by Hasler Foundation Program: Cyber Human Systems (project number 16066) and (in part) through a PhD fellowship of the Swiss Data Science Center, a joint venture between EPFL and ETH Zurich. It also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 725594 - time-data)

6. REFERENCES

[1] David L Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[3] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[4] Jonas Adler and Ozan Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Problems*, vol. 33, no. 12, pp. 124007, 2017.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[6] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis, "Compressed sensing using generative models," in *International Conference on Machine Learning*, 2017, pp. 537–546.

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

[8] Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon, "A bayesian perspective on the deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5443–5451.

[9] Reinhard Heckel and Paul Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," in *International Conference on Learning Representations*, 2019.

[10] Christopher A Metzler, Ali Mousavi, Reinhard Heckel, and Richard G Baraniuk, "Unsupervised learning with stein's unbiased risk estimator," *arXiv preprint arXiv:1805.10531*, 2018.

[11] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 945–948.

[12] Stephen Boyd, Neal Parikh, and Eric Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Now Publishers Inc, 2011.

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[14] Fabian Latorre, Armin Eftekhari, and Volkan Cevher, "Fast and provable admm for learning with generative priors," in *Advances in Neural Information Processing Systems*, 2019, pp. 12027–12039.

[15] Junfeng Yang and Xiaoming Yuan, "Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization," *Mathematics of computation*, vol. 82, no. 281, pp. 301–329, 2013.

[16] 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.

[17] Gary Mataev, Peyman Milanfar, and Michael Elad, "Deepred: Deep image prior powered by red," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[19] Antoni Buades, Bartomeu Coll, and J-M Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005, vol. 2, pp. 60–65.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S Kamilov, "Image restoration using total variation regularized deep image prior," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7715–7719.

[22] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.