Stochastic Solutions for Linear Inverse Problems using the Prior Implicit in a Denoiser

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep neural networks have provided state-of-the-art solutions for problems such as denoising, which implicitly rely on a prior probability model of natural images. Two recent lines of work – Denoising Score Matching and Plug-and-Play – propose methodologies for drawing samples from this implicit prior and using it to solve inverse problems, respectively. Here, we develop a parsimonious and robust generalization of these ideas for use in solving linear inverse problems. We rely on a classic statistical result that shows the least-squares solution for removing additive Gaussian noise can be written directly in terms of the gradient of the log of the noisy signal density. We use this to derive a stochastic coarse-to-fine gradient ascent procedure for drawing high-probability samples from the implicit prior embedded within a CNN trained to perform blind denoising. A generalization of this algorithm to constrained sampling provides a method for using the implicit prior to solve any deterministic linear inverse problem, with no additional training. The method relies on minimal assumptions and exhibits robust convergence over a wide range of parameter choices. To demonstrate the generality of our method we use the same algorithm to obtain state-of-the-art levels of unsupervised performance for deblurring, super-resolution, and compressive sensing.

1 Introduction

Many problems in image processing and computer vision rely, explicitly or implicitly, on prior probability models. Describing the full density of natural images is a daunting problem, given the high dimensionality of the signal space. Traditionally, models have been developed by combining assumed symmetry properties (e.g., translation-invariance, dilation-invariance), with simple parametric forms (e.g., Gaussian, exponential, Gaussian mixtures), often within pre-specified transformed coordinate systems (e.g., Fourier transform, multi-scale wavelets). While these models have led to steady advances in problems such as denoising (e.g., [1–6]), they are too simplistic to generate complex features that occur in our visual world, or to solve more demanding statistical inference problems.

In recent years, nearly all problems in image processing and computer vision have been revolutionalized by the use of deep Convolutional Neural Networks (CNNs). These networks are generally trained to perform specific tasks in a supervised fashion, without explicit use of an image prior. Their phenomenal performance suggests that they embed far more sophisticated priors than traditional methods. But these implicit priors arise from a combination of the distribution of the training data, the architecture of the network [7], regularization terms included in the optimization objective, and the optimization algorithm. Moreover, they are intertwined with the task for which they are optimized.

Nearly a decade ago, a strategy known as Plug-and-Play (P&P) was proposed for using a denoiser as a regularizer in solving other inverse problems [8], and a number of recent extensions have used this concept to develop MAP solutions for linear inverse problems [9–16]. Generally, the objective is decoupled into data fidelity and regularization terms, introducing a slack variable for use in a

proximal optimization algorithm (e.g., ADMM). The proximity operator of the regularization term can be seen as the MAP solution of a denoising problem, thus it can be substituted with a denoiser. A parallel line of research has focused on the use of generative models based on Score Matching [17–21]. The connection between Score Matching [22] and denoising autoencoders [23] was first shown in [24], by proving that the training criterion of a denoising autoencoder is equivalent to matching the score of the model and a Parzan density estimate of the data. Most recently, this idea has been used as the basis for drawing high-quality samples from the prior implicit in a CNN denoiser [20]. In Section 4 we elaborate on how these previous methods are related to our results.

Here, we derive a general algorithm for solving linear inverse problems using the prior implicit in a denoiser. We start with a result from classical statistics [25] that states that a denoiser that aims to minimize squared error of images corrupted by additive Gaussian noise may be interpreted as computing the gradient of the log of the density of noisy images. This result is related to Score Matching, but provides a more direct relationship between least-squares optimal denoising and the embedded prior [26, 16]. We develop a stochastic ascent algorithm that uses this denoiser-estimated gradient to draw high-probability samples from the embedded prior. Importantly, the gradient step sizes and the amplitude of injected noise are jointly and adaptively controlled by the denoiser, enabling robust and efficient convergence. We then combine this procedure with constraints arising from any deterministic linear measurement of an image to draw samples from the prior conditioned on this measurement, thus providing a general stochastic solution for any deterministic linear inverse problem. We demonstrate that our method produces visually high-quality results with state-of-the-art PSNR levels for unsupervised performance on superresolution, deblurring and compressive sensing.[1]

1.1 Image priors, manifolds, and noisy observations

Digital photographic images lie in a high-dimensional space ($\mathbb{R}^N$, where N is the number of pixels), and simple thought experiments suggest that they are concentrated on or near low-dimensional manifolds whose local coordinates represent continuous deformations and intensity variations. We can associate with this manifold a prior probability model, $p(x)$, by assuming that images within the manifold have constant or slowly-varying probability, while unnatural or distorted images (which lie off the manifold) have low or zero probability. Suppose we make a noisy observation of an image, $y = x + z$, where $x \in R^N$ is the original image drawn from $p(x)$, and $z \sim \mathcal{N}(0, \sigma^2 I_N)$ is a sample of Gaussian white noise. The observation density $p(y)$ is related to the prior $p(x)$ via marginalization:

$$p(y) = \int p(y|x)p(x)dx = \int g(y-x)p(x)dx, \tag{1}$$

where $g(z)$ is the Gaussian noise distribution. Equation (1) is in the form of a convolution, and thus $p(y)$ is a Gaussian-blurred version of the signal prior, $p(x)$. Moreover, the family of observation densities over different noise variances, $p_\sigma(y)$, forms a Gaussian scale-space representation of the prior [27, 28], analogous to the temporal evolution of a diffusion process.

1.2 Least squares denoising and CNNs

Given a noisy observation, y, the least squares estimate of the true signal is well known to be the conditional mean of the posterior:

$$\hat{x}(y) = \int x p(x|y)dx = \int x \frac{p(y|x)p(x)}{p(y)} dx \tag{2}$$

The structure of the equation mirrors a traditional approach to the problem: one chooses a prior probability model, $p(x)$, combines it with a likelihood function describing the noise, $p(y|x)$, and solves. Modern denoising solutions, on the other hand, are often based on discriminative training. One expresses the estimation function (as opposed to the prior) in parametric form, and sets the parameters by minimizing the denoising MSE over a large training set of example signals and their noise-corrupted counterparts [29-32]. Current state-of-the-art denoising results using CNNs obtained with this supervised approach are far superior to results of previous methods [33-35]. Recent work [36] demonstrates that these architectures can be simplified by removing all additive bias terms, with no loss of performance. The resulting *bias-free* networks may be analyzed as adaptive linear systems,

[1] URL of the code is available in appendix

which reveals that they perform an approximate projection onto a low-dimensional subspace. In our context, we interpret this subspace as a tangent hyperplane of the image manifold.

1.3 Exposing the implicit prior through Empirical Bayes estimation

Trained CNN denoisers contain detailed prior knowledge of image structure, but Eq. (2) implies that it is embedded within a high-dimensional integral. How can we obtain access to this implicit prior? Recent results have derived relationships between Score Matching density estimates and denoising, and have used these relationships to make use of implicit prior information [37, 38, 20, 39]. Here, we exploit a more direct but less-known result from the literature on Empirical Bayesian estimation. The idea was introduced in [40], extended to the case of Gaussian additive noise in [25] (see also [16]), and generalized to many other measurement models [26]. For the Gaussian noise case, the least-squares estimate of Eq. (2) may be rewritten as:

$$\hat{x}(y) = y + \sigma^2 \nabla_y \log p(y). \tag{3}$$

The proof is relatively straightforward. The gradient of the observation density of Eq. (1) is:

$$\nabla_y \, p(y) = \frac{1}{\sigma^2} \int (x - y) g(y - x) p(x) dx = \frac{1}{\sigma^2} \int (x - y) p(y, x) dx.$$

Multiplying both sides by $\sigma^2 / p(y)$ and separating the right side into two terms gives:

$$\sigma^2 \frac{\nabla_y \, p(y)}{p(y)} = \int x p(x|y) dx - \int y p(x|y) dx = \hat{x}(y) - y.$$

Rearranging terms and using the chain rule to compute the gradient of the log gives Miyasawa's result, as expressed in Eq. (3). This remarkable result, which re-expresses the integral over the prior and likelihood of Eq. (2) in terms of a *gradient*. Note that 1) the relevant density is not the prior, $p(x)$, but the noisy *observation density*, $p(y)$; 2) the gradient is computed on the *log* density (the associated "energy function"); and 3) the gradient adjustment is *not* iterative - the optimal solution is achieved in a single step, and holds for any noise level, σ.

2 Drawing high-probability samples from the implicit prior

Suppose we wish to draw a sample from the prior implicit in a denoiser. Equation (3) allows us to generate an image proportional to the gradient of $\log p(y)$ by computing the denoiser residual, $f(y) = \hat{x}(y) - y$. Song and Ermon [20] developed a Markov chain Monte Carlo (MCMC) scheme, combining gradient steps derived from Score Matching and injected noise in a Langevin sampling algorithm to draw samples from a sequence of densities $p_\sigma(y)$, while reducing σ in discrete steps, each associated with an appropriately trained denoiser. In contrast, starting from a random initialization, y_0, we aim to find a high-probability image (i.e., an image from the manifold) using a more direct and efficient stochastic gradient ascent procedure.

We compute gradients using the residual of a bias-free universal CNN denoiser, which automatically estimates and adapts to each noise level. On each iteration, the algorithm takes a small step in the direction specified by the denoiser, moving toward the image manifold and thereby reducing the amplitude of the effective noise. Under the interpretation that the denoiser performs a projection onto the current approximation of the image manifold, this noise reduction occurs in the subspace orthogonal to that manifold, and noise components parallel to the maniofld are retained. As the effective noise decreases, the observable dimensionality of the image manifold increases [36], enabling the synthesis of detailed image content. Since the family of observation densities, $p_\sigma(y)$ forms a scale-space representation of $p(x)$, the algorithm may be viewed as a form of coarse-to-fine optimization [41-44]. Assuming the step sizes are adequately controlled, the procedure will converge to a point on the manifold. Figure 7 provides a visualization of this process in two dimensions.

Each iteration operates by taking a deterministic step in the direction of the gradient (as obtained from the denoising function) and injecting some additional noise:

$$y_t = y_{t-1} + h_t f(y_{t-1}) + \gamma_t z_t, \tag{4}$$

where $f(y) = \hat{x}(y) - y$ is the residual of the denoising function, which is proportional to the gradient of $p(y)$, from Eq. (3). The parameter $h_t \in [0, 1]$ controls the fraction of the denoising correction

that is taken, and γ_t controls the amplitude of a sample of white Gaussian noise, $z_t \sim \mathcal{N}(0, I)$. The purpose of injecting noise is two-fold. First, from an optimization perspective, it allows the method to avoid getting stuck in local maxima. Second, it allows stochastic exploration of the manifold, yielding a more diverse (higher entropy) family of solutions. The effective noise variance of image y_t is:

$$\sigma_t^2 = (1 - h_t)^2 \sigma_{t-1}^2 + \gamma_t^2, \tag{5}$$

where the first term is the variance of the noise remaining after the denoiser correction (assuming the denoiser is perfect), and the second term is the variance arising from the injected noise. The assumption of perfect denoising is an idealization, but we show empirically (Fig. 8) that the algorithm converges reliably, with error levels falling as predicted by Eq. (5) or faster, across different settings of β and h_0.

To ensure convergence, we require the effective noise variance on each time step to be reduced, despite the injection of additional noise. For this purpose, we introduce a parameter $\beta \in [0, 1]$ to control the proportion of injected noise ($\beta = 1$ indicates no noise), and enforce the convergence by requiring that:

$$\sigma_t^2 = (1 - \beta h_t)^2 \sigma_{t-1}^2. \tag{6}$$

Combining this with Eq. (5) yields an expression for γ_t in terms of h_t:

$$\gamma_t^2 = \left[(1 - \beta h_t)^2 - (1 - h_t)^2\right] \sigma_{t-1}^2 = \left[(1 - \beta h_t)^2 - (1 - h_t)^2\right] \left\| f(y_{t-1}) \right\|^2 / N,$$

where the second equation assumes that the magnitude of the denoising residual provides a good estimate of the effective noise standard deviation, as was found in [36]. This allows the denoiser to adaptively control the gradient ascent step sizes, reducing them as the y_t approaches the manifold (see Fig. 7). This automatic adjustment results in efficient and reliable convergence, as demonstrated empirically in Fig. 1. Our initial implementation with a small constant fractional step size $h_t = h_0$ produced high quality results, but required many iterations. Intuitively, step sizes that are a fixed proportion of the distance to the manifold lead to exponential decay - a form of Zeno's paradox. To accelerate convergence, we introduced a schedule for increasing the step size proportion, starting from $h_0 \in [0, 1]$. The sampling process is summarized in Algorithm 1, and is provided as a block diagram in Fig. 6.

Algorithm 1: Coarse-to-fine stochastic ascent method for sampling from the implicit prior of a denoiser, using denoiser residual $f(y) = \hat{x}(y) - y$.

parameters: $\sigma_0, \sigma_L, h_0, \beta$
initialization: $t = 1$, draw $y_0 \sim \mathcal{N}(0.5, \sigma_0^2 I)$
while $\sigma_{t-1} \geq \sigma_L$ **do**
  $h_t = \frac{h_0 t}{1 + h_0 (t-1)}$;
  $d_t = f(y_{t-1})$;
  $\sigma_t^2 = \frac{||d_t||^2}{N}$;
  $\gamma_t^2 = \left((1 - \beta h_t)^2 - (1 - h_t)^2\right) \sigma_t^2$;
  Draw $z_t \sim \mathcal{N}(0, I)$;
  $y_t \leftarrow y_{t-1} + h_t d_t + \gamma_t z_t$;
  $t \leftarrow t + 1$
end

2.1 Image synthesis examples

For the denoiser, we used BF-CNN [36], a bias-free variant of DnCNN [33]. We obtained similar results (not shown) using other bias-free CNN architectures described in [36]. We trained this network on three different datasets: 40×40 patches cropped from Berkeley segmentation training set [45], in color and grayscale, and MNIST dataset [46] (see Appendix A for further details). We chose parameters $\sigma_0 = 1, \sigma_L = 0.01$, and $h_0 = 0.01$. Figure 2 provides visualization of two example trajectories, and diversity of samples (right). Additional visual examples, obtained with different levels of β, are shown in Appendix D.

3 Solving deterministic linear inverse problems using the implicit prior

Many applications in signal processing can be expressed as deterministic linear inverse problems - deblurring, super-resolution, estimating missing pixels (e.g., inpainting), and compressive sensing

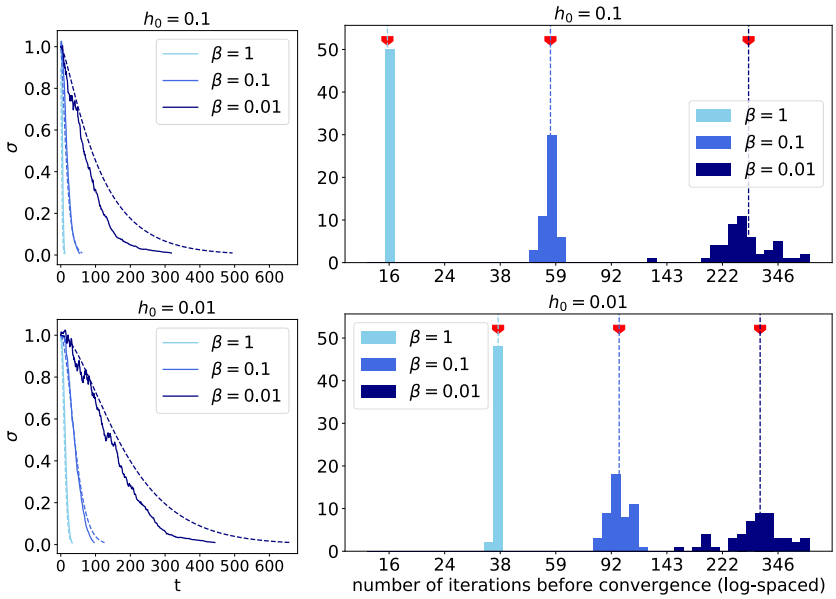


Figure 1: Convergence of sampling algorithm, quantified in terms of the effective noise standard deviation $\sigma = \frac{||d_t||}{\sqrt{N}}$ for three different levels of β and 2 levels of h_0. **Left.** Convergence of single examples (solid curves) is well-behaved and efficient in all cases. Dashed curves indicate the convergence predicted from the formulation of the algorithm: $\sigma_t = (1 - \beta h_t)\sigma_{t-1}$. For $\beta = 1$ (no injected noise), the empirical convergence closely approximates the prediction. For larger amounts of injected noise (smaller β), convergence is slower, but faster than predicted. A conceptual explanation of this observation is related to the dimensionality of the image subspace tangent to the observable manifold. The injected noise has non-zero amplitude in all directions, including directions parallel to the tangent subspace. These directions do not contribute to the effective noise variance because they contain image content. The number of these directions increases with reduction of noise [36] resulting in increasing overestimation of noise variance in the predicted curves. **Right.** Distribution of number of iterations before convergence to a criterion level of $\sigma = 0.01$ for 50 images. Red symbols indicate average values for each β and h_0.

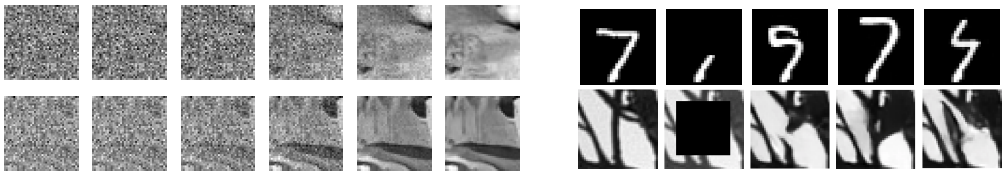


Figure 2: **Left.** Visualization of sampling algorithm trajectories. Each row shows a sequence of images, $y_t, t = 1, 9, 17, 25, \ldots$, from the iterative sampling procedure, with different initializations, y_0, and no added noise ($\beta = 1$), demonstrating the way that the algorithm amplifies and "hallucinates" structure found in the initial (noise) images. **Right.** Sampling diversity. Inpainting examples generated using two BF-CNN denoisers. First column: original images. Second column: partially measured images (with missing block of pixels). Right three columns: Restored examples. Each initialization results in a different restored image.

are all examples. Given a set of linear measurements of an image, $x^c = M^T x$, where M is a low-rank measurement matrix, one attempts to recover the original image. In Section 2, we developed a stochastic gradient-ascent algorithm for obtaining a high-probability sample from $p(x)$. Here, we modify this algorithm to solve for a high-probability sample from the conditional density $p(x|M^T x = x^c)$. Geometrically, constrained sampling corresponds to drawing points which sit at the intersection of the image manifold and the constrained hyperplane (see Fig. 11). Note that for these problems, the injection of noise ($\beta < 1$) is particularly important, because points on the intersection are not necessarily the closest points to the initial image.

3.1 Constrained sampling algorithm

Consider the distribution of a noisy image, y, conditioned on the linear measurements, $x^c = M^T x$. Without loss of generality, we assume the columns of the matrix M are orthogonal unit vectors (if not, we can re-parameterize to an equivalent constraint using the SVD. In this case, M is the pseudo-inverse of M^T, and that matrix MM^T can be used to project an image onto the measurement subspace). Using Bayes' rule, we write the conditional density of the noisy image conditioned on the linear measureement as

$$p(y|x^c) = p(y^c, y^u|x^c) = p(y^u|y^c, x^c)p(y^c|x^c) = p(y^u|x^c)p(y^c|x^c)$$

where $y^c = M^T y$, and $y^u = \bar{M}^T y$ (the projection of y onto the orthogonal complement of M). As with the algorithm of Section 2, we wish to obtain a local maximum of this function using stochastic coarse-to-fine gradient ascent. Applying the operator $\sigma^2 \nabla \log(\cdot)$ yields

$$\sigma^2 \nabla_y \log p(y|x^c) = \sigma^2 \nabla_y \log p(y^u|x^c) + \sigma^2 \nabla_y \log p(y^c|x^c).$$

The second term is the gradient of the observation noise distribution, projected into the measurement space. If this is Gaussian with variance σ^2, it reduces to $M(y^c - x^c)$. The first term is the gradient of a function defined only within the subspace orthogonal to the measurements, and thus can be computed by projecting the measurement subspace out of the full gradient. Combining these gives:

$$\begin{aligned} \sigma^2 \nabla_y \log p(y) &= (I - MM^T)\sigma^2 \nabla_y \log p(y) + M(x^c - y^c) \\ &= (I - MM^T) f(y) + M(x^c - M^T y). \end{aligned} \tag{7}$$

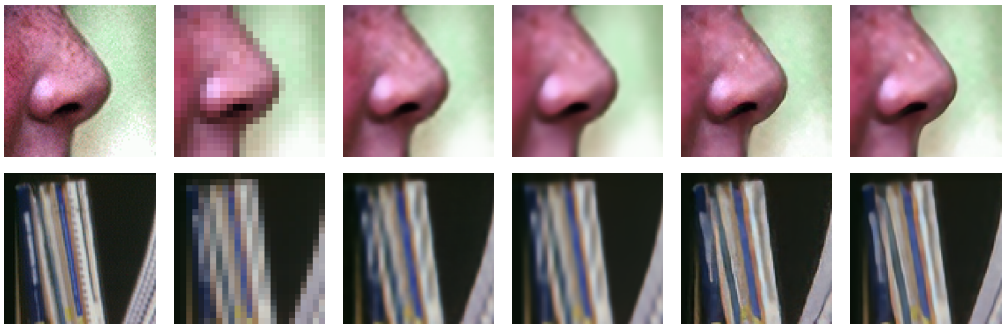
Thus, we see that the gradient of the conditional density is partitioned into two orthogonal components, capturing the gradient of the (log) noisy density, and the deviation from the constraints, respectively. To draw a high-probability sample from $p(x|x^c)$, we use the same algorithm described in Section 2, substituting Eq. (7) for the deterministic update vector, d_t (see Algorithm 2 in Appendix).

3.2 Linear inverse examples

We evaluate our method on three linear inverse problems (and provide two additional problems in Appendix E). The same algorithm and parameters are used on all problems - only the measurement matrix M and measured values $M^T x$ are altered. In particular, as in section 2.1, we used BF-CNN [36], and chose parameters $\sigma_0 = 1, \sigma_L = 0.01, h_0 = 0.01, \beta = 0.01$. For each example, we show original images (x), the direct least-squares reconstruction ($MM^T x$), and restored images. Subjective assessment of perceptual quality is particularly important in cases where the measurement matrix is of very low rank, and the distribution of solutions is diverse (the synthesis examples of the previous section correspond to the limiting case, with measurements of rank zero). We also provide numerical comparisons with other unsupervised methods, in terms of both PSNR and SSIM (an approximate measure of perceptual quality).

Spatial super-resolution. Here, one aims to reconstruct a high resolution image from a low resolution (i.e. downsampled) image. Downsampling is typically performed after lowpass filtering, and the downsampling factor and filter kernel determine the measurement model, M. Here, we use a 4×4 block-averaging filter, and 4×4 downsampling (i.e., measurements are averages over non-overlapping blocks). We compare to two recent unsupervised methods, Deep image Prior (DIP) [7] and DeepRED [13]. DIP chooses a random input vector, and adjusts the weights of a CNN to minimize the mean square error between the output and the corrupted image. Regularization by denoising (RED) is a recent successful example of Plug-and-Play [9]. DeepRED [13] combines DIP and RED, obtaining better performance than either method alone.

Examination of results on two example images demonstrates that our method produces results that are sharper with less noticeable artifacts (Fig. 3). Despite this, the PSNR and SSIM values are slightly worse (Table 1). These can be improved by averaging over realizations (last column of Table 1), at the expense of some blurring (last column of Fig. 3). This is expected: the algorithm produces samples of the prior subject to the measurement constraint, but the least-squares optimal solution is the mean of this distribution. If the the samples are drawn from a curved manifold, the average (a convex combination of those points) will lie off the manifold. Finally, our method is more than two orders of magnitude faster than either DIP or DeepRED (bottom row, Table 1).



cropped low res DIP [7] DeepRED [7] Ours Ours - avg

Figure 3: Spatial super-resolution. First column shows original images (cropped portion). Second column shows cropped portion with resolution reduced by averaging over 4x4 blocks (dimensionality reduction to 6.25%). Next three columns show reconstruction results obtained using DIP [7], DeepRED [13], and our method. In all cases, our method produces an image that is sharper with less noticeable artifacts (e.g., note blocking/aliasing artifacts along diagonal contours of lower image). The last column shows an average over 10 samples obtained by our method.

Table 1: Spatial super-resolution performance over Set5 (top 2 rows) and Set14 (second 2 rows). Values indicate YCbCr-PSNR (SSIM). Last row shows average Set14 runtime (sec) on a DGX GPU.

Factor	MM^Tx	DIP [7]	DeepRED [13]	Ours	Ours - avg
4:1	26.35 (0.826)	30.04 (0.902)	30.22 (0.904)	29.47 (0.894)	**31.20** (**0.913**)
8:1	23.02 (0.673)	24.98 (0.760)	24.95 (0.760)	25.07 (0.767)	**25.64** (**0.792**)
4:1	24.65 (0.765)	26.88 (0.815)	27.01 (0.817)	26.56 (0.808)	**27.14** (**0.826**)
8:1	22.06 (0.628)	23.33 (0.685)	23.34 (0.685)	23.32 (0.681)	**23.78** (**0.703**)
runtime		1,190	1,584	**9**	10×9

Deblurring (spectral super-resolution). The applications described above (and in Appendix E) are based on partial measurements in the pixel domain. Here, we consider a blurring operator that retains a set of low-frequency coefficient in the Fourier domain, discarding the rest. This is equivalent to convolving the image with a sinc kernel. In this case, M consists of the preserved low-frequency columns of the discrete Fourier transform, and MM^Tx is the blurred version of x. Example images are shown in Fig. 4 and numerical comparisons are shown in Table 2. Our method produces strong results, both perceptually and numerically.

Compressive sensing. Compressive sensing [50, 51] aims to recover signals from a small number of (typically, random) linear measurements. In brief, one acquires measurements with a sensing matrix containing a set of $n << N$ random orthogonal axes, and solves the inverse problem by

cropped low res DIP [7] DeepRED [7] Ours Ours - avg

Figure 4: Deblurring (spectral super-resolution). Measurements correspond to 5% of low frequencies.

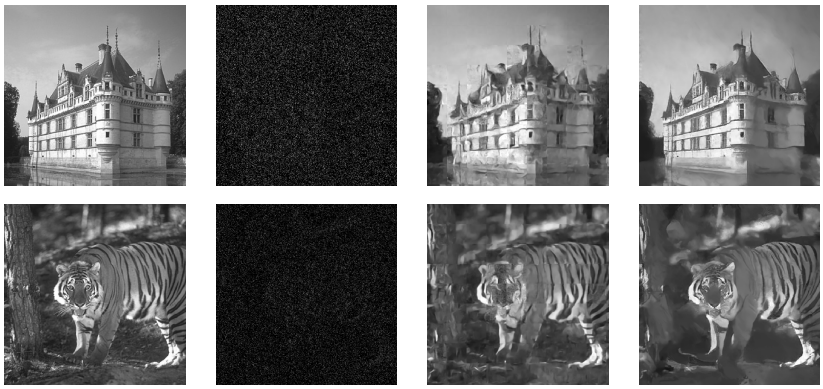
Table 2: Spectral super-resolution (deblurring) performance over Set5, in YCbCr-PSNR (SSIM).

Ratio	MM^Tx	DIP [7]	DeepRED [13]	Ours	Ours - avg
10%	30.2 (0.91)	32.54 (0.93)	32.63 (0.93)	31.82 (0.93)	**32.78** (**0.94**)
5%	27.77 (0.85)	29.88 (0.89)	29.91 (0.89)	29.22 (0.89)	**30.07** (**0.90**)

Table 3: Compressive sensing performance over Set68 [45]. Values indicate PSNR (SSIM).

Ratio	TVAL3 [47]	ISTA-Net [48]	DIP [7]	BNN [49]	Ours
25%	26.48 (0.77)	29.07 (0.84)	27.78 (0.80)	28.63 (0.84)	**29.16** (**0.88**)
10%	22.49 (0.58)	25.23 (0.69)	24.82 (0.69)	25.24 (0.71)	**25.47** (**0.78**)
4%	19.10 (0.42)	22.02 (0.54)	22.51 (0.58)	**22.52** (0.58)	22.07 (**0.68**)

assuming a sparse prior. Photographic images are not truly sparse in any fixed linear basis, but they can be reasonably approximated by low-dimensional subsets of Fourier or wavelet basis functions, and compressive sensing results are typically demonstrated using one of these. The manifold prior embedded within our CNN denoiser corresponds to a nonlinear form of sparsity, and our stochastic coarse-to-fine ascent algorithm can be used to recover an image from the measured linear projections onto the random basis. We compare to four other methods. TVAL3 [47] is an optimization-based algorithm with total variation regularization, ISTA-Net [48] is a block-based supervised CNN method trained to reconstruct images from measurements obtained from a single pre-specified measurement matrix. BNN [49] is an unsupervised Bayesian method for solving compressive sensing problems. DIP [7] was previously described. Fig. 5 shows results for two example images, and Table 3 summarizes numerical performance. All values are taken from [49] except for ISTA-Net which were obtained by running the open-source code. Our method generally outperforms all other methods, even those that are specialized for Compressive Sensing (ISTA-NET, BNN).



original measured ISTA-Net [48] Ours

Figure 5: Compressive sensing. Measurement matrix M contains random, orthogonal unit vectors, with dimensionality reduced to 10% of the number of image pixels. Third column: images recovered using ISTA-Net (trained, supervised, for a particular measurement matrix). Fourth column: Our method, which again exhibits fewer artifacts.

4 Related work

Our method is conceptually similar to Plug-and-Play[8], in using a denoiser to solve linear inverse problems. But it differs in a number of important ways: (1) *direct relationship between prior and denoiser mapping.* We derive our method from Miyasawa's relationship between the mapping and the prior, which is exact and explicit, and makes the algorithm interpretable. The rationale for using a denoiser as a regularizer in P&P frameworks arises from interpreting the proximal operator of the regularizer as a MAP solution of a denoising problem, providing an indirect connection to prior; (2)

sampling vs. MAP estimation. We sample a high probability image from the implicit prior that is consistent with a linear constraint. This stochastic solution does not minimize MSE, but has high perceptual quality. RED and other P&P methods are derived as MAP solutions, and although this is not equivalent to minimizing MSE (maximizing PSNR), the results generally have better PSNR than our sampling results, but are visually more blurred (see Figs. 3 and 4); (3) *automatic step-size selection and convergence.* P&P, which uses ADMM for optimization, relies on hyper-parameter selection that are critical for convergence (as discussed extensively in [9]). Our algorithm adjusts step-size automatically using the blind denoiser, with only two primary hyper-parameters (h_0 and β), and convergence is robust to choices of these (Fig. 1).

Our method is also closely related to recent work that uses Score Matching [22] to draw samples from an implicit prior. This line of work is rooted in the connection between Denoising Autoencoders and Score Matching, first described in [24]. The idea that denoising autoencoders learn the gradient of log density, and its connection to an underlying data manifold has been explored in [17, 37, 23]. More recently, Refs. [38, 18] trained a neural network to directly estimate energy (the negative log prior) and interpreted its gradient as the Score of the data. Finally, in a breakthrough paper that partially inspired our approach, Song and Ermon [20] trained a sequence of denoisers with decreasing levels of noise and used Langevin dynamics to sample from the underlying probability distributions in succession. Our work differs from these in several important ways: (1) *Direct derivation.* Our method is based on Miyasawa's explicit relationship between the denoiser mapping and implicit density [25], which we prove in a few lines of math (Section 1.3). The relationship of this result to Score Matching [26, 16], and the relationship between Score Matching and Denoising Autoencoders [24] are both significantly more nuanced and complex. (2) *Automatic stepsize adjustment.* Our method assumes and uses a single (universal) blind denoiser, rather than a family of denoisers trained for a pre-specified sequence of noise levels. In our method, the noise level schedule, hence the step size, is automatically adjusted by a single denoiser, based on its implicit estimate of distance to the manifold. (3) *Efficiency*. Our method is not based on Langevin dynamics (or any other MCMC method, all of which are computationally expensive). MCMC-based methods require convergence to and sampling from each of a discrete sequence of densities. Our algorithm uses gradient ascent to continuously maximize probability, resulting in substantial gains in efficiency. For comparison, Song&Ermon [20] report 1000 iteration to synthesize a 32×32 image, whereas we synthesize 40×40 images in roughly 35 iteration (for $\beta = 1$). (4) *Image synthesis vs. linear inverse problems.* Finally, we've focused the use of our method as a general solver for five different linear inverse problems, whereas Score Matching methods to date have been generally focused on unconditional image synthesis.

5 Discussion

We've described a framework for using the prior embedded in a denoiser to solve inverse problems. Specifically, we developed a stochastic coarse-to-fine gradient ascent algorithm that uses the denoiser to draw high-probability samples from its implicit prior, and a constrained variant that can be used to solve *any* deterministic linear inverse problem. It is worth noting that the assumed use of Gaussian additive noise in training the denoiser is necessary only to justify the use of Miyasawa's expression (Eq. 3), but does not impose any limitations on the subsequent use of the trained denoiser for sampling from the implicit prior or solving inverse problems. Denoisers can be trained supervised on unlimited amounts of unlabeled data, and as such, our method extends the power of supervised learning to a much broader set of problems, without further training.

Limitations. As discussed following Eq. (5), the algorithm is constructed based on an idealized assertion that the denoiser residual exactly cancels the noise. We showed empirically (Fig. 1) that this idealization does not disturb convergence for BF-CNN, but we cannot guarantee this would hold for all other denoisers. More generally, although the method can be used with any universal least-squares denoiser designed or trained to remove Gaussian noise, the complexity of the embedded prior, and thus the quality of our linear inverse results, relies heavily on the expressive power of the denoiser (BF-CNN in our case), as well as the diversity of the training set. Finally, the current algorithm is designed to solve *deterministic linear* inverse problems. We are currently exploring extensions to inverse problems involving noisy or nonlinear measurements.

References

[1] D Donoho. Denoising by soft-thresholding. *IEEE Trans Info Theory*, 43:613–627, 1995.

[2] P Moulin and J Liu. Analysis of multiresolution image denoising schemes using a generalized Gaussian and complexity priors. *IEEE Trans Info Theory*, 45:909–919, 1999.

[3] Hyvärinen. Sparse code shrinkage: Denoising of nonGaussian data by maximum likelihood estimation. *Neural Computation*, 11(7):1739–1768, 1999.

[4] J. Romberg, H. Choi, and R. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. *IEEE Trans Image Proc*, 10(7), July 2001.

[5] L Şendur and I W Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans Signal Proc*, 50(11):2744–2756, November 2002.

[6] J Portilla, V Strela, M J Wainwright, and E P Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans Image Proc*, 12(11):1338–1351, Nov 2003. Recipient, IEEE Signal Processing Society Best Paper Award, 2008.

[7] D Ulyanov, A Vedaldi, and V Lempitsky. Deep image prior. *Proc Int'l J Computer Vision*, pages 1867–1888, April 2020.

[8] 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*, pages 945–948. IEEE, 2013.

[9] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (RED). *CoRR*, abs/1611.02862, 2017.

[10] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[11] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1799–1808, 2017.

[12] S. H. Chan, X. Wang, and O. A. Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2017.

[13] 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*, pages 0–0, 2019.

[14] Afonso M Teodoro, José M Bioucas-Dias, and Mário AT Figueiredo. Image restoration and reconstruction using targeted plug-and-play priors. *IEEE Transactions on Computational Imaging*, 5(4):675–686, 2019.

[15] Y. Sun, B. Wohlberg, and U. S. Kamilov. An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3):395–408, 2019.

[16] E. T. Reehorst and P. Schniter. Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on Computational Imaging*, 5(1):52–67, 2019.

[17] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

[18] Saeed Saremi, Bernhard Schölkopf, Arash Mehrjou, and Aapo Hyvärinen. Deep energy estimator networks. *ArXiv e-prints (arXiv.org)*, 1805.08306, May 2018.

[19] Zengyi Li, Yubei Chen, and Friedrich T. Sommer. Learning energy-based models in high-dimensional spaces with multi-scale denoising score matching, 2019.

[20] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32*, pages 11918–11930. Curran Associates, Inc., 2019.

[21] Siavash A Bigdeli, Geng Lin, Tiziano Portenier, L Andrea Dunbar, and Matthias Zwicker. Learning generative models using denoising density estimators. *arXiv preprint arXiv:2001.02728*, 2020.

[22] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[23] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

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

[25] K Miyasawa. An empirical Bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist.*, 38:181–188, 1961.

[26] M Raphan and E P Simoncelli. Least squares estimation without priors or supervision. *Neural Computation*, 23(2):374–420, Feb 2011.

[27] J J Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[28] T Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[29] Y. Hel-Or and D. Shaked. A discriminative approach for wavelet shrinkage denoising. *IEEE Trans. Image Processing*, 17(4), April 2008.

[30] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. on Image processing*, 15(12):3736–3745, 2006.

[31] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in neural information processing systems*, pages 769–776, 2009.

[32] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399. IEEE, 2012.

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

[34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[35] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Patt. Analysis and Machine Intelligence*, 39(6):1256–1272, 2017.

[36] S Mohan*, Z Kadkhodaie*, E P Simoncelli, and C Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *Int'l. Conf. on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, April 2020.

[37] Y Bengio, L Yao, G Alain, and P Vincent. Generalized denoising auto-encoders as generative models. In *Adv. Neural Information Processing Systems (NIPS*13)*, pages 899–907. MIT Press, 2013.

[38] Saeed Saremi and Aapo Hyvarinen. Neural empirical Bayes. *Journal of Machine Learning Research*, 20:1–23, 2019.

[39] Zengyi Li, Yubei Chen, and Friedrich T Sommer. Annealed denoising score matching: Learning energy-based models in high-dimensional spaces. *arXiv preprint arXiv:1910.07762*, 2019.

[40] H Robbins. An empirical Bayes approach to statistics. *Proc. Third Berkley Symposium on Mathematcal Statistics*, 1:157–163, 1956.

[41] B D Lucas and T Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th Int'l Joint Conf. on Artificial Intelligence*, pages 674–679, Vancouver, 1981.

[42] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[43] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[44] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[45] 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 *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[46] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

[47] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang. An efficient augmented lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3):507–530, 2013.

[48] Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018.

[49] Tongyao Pang, Yuhui Quan, and Hui Ji. Self-supervised bayesian deep learning for image recovery with applications to compressive sensing. 2020.

[50] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[51] D. L. Donoho. Compressed sensing. *IEEE Trans Info Theory*, 52(4):1289–1306, 2006.

[52] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We claim the algorithm is general, simple, and efficient. General: we solve five inverse problems using the same algorithm and parameters, without any additional training. Simple: the derivation is based on a simple/direct expression relating denoising to priors, and a few basic empirical facts about universal CNN denoisers. Efficiency: empirically demonstrated in Fig. 1 and Table 1.
   (b) Did you describe the limitations of your work? [Yes] See second paragraph in section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We're not aware of any direct negative societal impacts of this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections 1.1 through 1.3

(b) Did you include complete proofs of all theoretical results? [Yes] See Sections 1.1 through 1.3

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Provided as a URL in the appendix.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We used publicly available pre-trained CNNs. Nonetheless we provide training details in Section 2.1, and Appendix A (in addition to citing the paper which proposed the CNN and the training scheme).

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] For convergence times, we show full distributions, rather than summarizing with error bars Fig. 1.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Run time, number of iterations, and type of GPU are reported in tables 1 and 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Provided citations for all datasets (MNIST, Berkeley segmentation dataset), pre-trained denoiser model (BF-CNN).

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] URL provided in Appendix

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]