One Network to Solve Them All — Solving Linear Inverse Problems using Deep Projection Models

J. H. Rick Chang*, Chun-Liang Li, Barnabás Póczos, B. V. K. Vijaya Kumar, and Aswin C. Sankaranarayanan
Carnegie Mellon University, Pittsburgh, PA

Abstract

While deep learning methods have achieved state-of-the-art performance in many challenging inverse problems like image inpainting and super-resolution, they invariably involve problem-specific training of the networks. Under this approach, each inverse problem requires its own dedicated network. In scenarios where we need to solve a wide variety of problems, e.g.*, on a mobile camera, it is inefficient and expensive to use these problem-specific networks. On the other hand, traditional methods using analytic signal priors can be used to solve any linear inverse problem; this often comes with a performance that is worse than learning-based methods. In this work, we provide a middle ground between the two kinds of methods — we propose a general framework to train a single deep neural network that solves arbitrary linear inverse problems. We achieve this by training a network that acts as a quasi-projection operator for the set of natural images and show that any linear inverse problem involving natural images can be solved using iterative methods. We empirically show that the proposed framework demonstrates superior performance over traditional methods using wavelet sparsity prior while achieving performance comparable to specially-trained networks on tasks including compressive sensing and pixel-wise inpainting.*

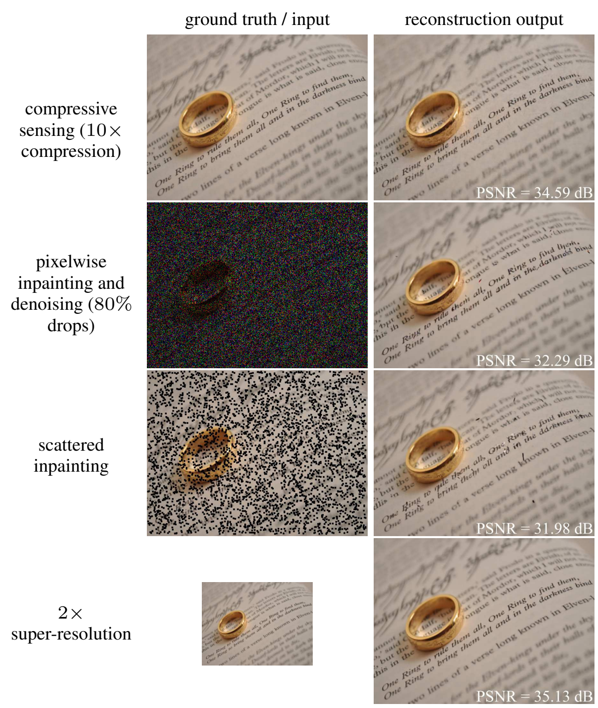

Figure 1: The same network is used to solve the following tasks: compressive sensing problem with $10\times$ compression, pixelwise random inpainting with 80% dropping rate, scattered inpainting, and $2\times$-super-resolution. Note that even though the nature and input dimensions of the problems are very different, the proposed framework is able to use a single network to solve them all without retraining.

1. Introduction

At the heart of many image processing tasks is a linear inverse problem, where the goal is to reconstruct an image $\mathbf{x} \in \mathbb{R}^d$ from a set of measurements $\mathbf{y} \in \mathbb{R}^m$ of the form $\mathbf{y} = A\mathbf{x} + \mathbf{n}$, where $A \in \mathbb{R}^{m \times d}$ is the measurement operator and $\mathbf{n} \in \mathbb{R}^m$ is the noise. For example, in image inpainting, A is the linear operation of applying a pixelwise mask to the image $\mathbf{x}$. In super-resolution, A downsamples high-resolution images. In compressive sensing, A is a short-fat matrix with fewer rows than columns and is typically a random sub-Gaussian or a sub-sampled orthonormal matrix. Linear inverse problems are often underdetermined, *i.e.*, they involve fewer measurements than unknowns. Such underdetermined systems are extremely difficult to solve since the operator A has a non-trivial null space and there are an infinite number of feasible solutions; however, only a few of the feasible solutions are valid natural images.

*Chang, Bhagavatula and Sankaranarayanan were supported, in part, by the ARO Grant W911NF-15-1-0126. Chang was also partially supported by the CIT Bertucci Fellowship. Sankaranarayanan was also supported, in part, by the INTEL ISRA on Compressive Sensing.

Solving linear inverse problems. There are two broad approaches for solving linear underdetermined problems. The first approach regularizes the inverse problem with signal priors that identify the true solution from the infinite set of feasible solutions [9, 18, 19, 31, 39]. However, most hand-designed signal priors provide limited identification ability, i.e., many non-image signals can satisfy the constraints and be falsely identified as natural images. The second approach learns a direct mapping from the linear measurement $\mathbf{y}$ to the solution $\mathbf{x}$, with the help of large training datasets and deep neural nets. Such methods have achieved state-of-the-art performance in many challenging image inverse problems like super-resolution [17, 29], inpainting [38], compressive sensing [28, 35, 36], and image debluring [49]. Despite their superior performance, these methods are designed for specific problems and usually cannot solve other problems without retraining the mapping function — even when the problems are similar. For example, a $4\times$-super-resolution network cannot be easily readapted to solve $2\times$ super-resolution problems; a compressive sensing network for Gaussian random measurements is not applicable to sub-sampled Hadamard measurements. Training a new network for every single measurement operator is a wasteful proposition. In comparison, traditional methods using hand-designed signal priors can solve any linear inverse problems but they often have poorer performance on an individual problem. Clearly, a middle ground between these two classes of methods is needed.

One network to solve them all. We ask the following question: *if we have a large dataset of natural images, can we learn from the dataset a signal prior that can deal with* ***any*** *linear inverse problem involving images?* Such a signal prior can significantly lower the cost to incorporate inverse algorithms into consumer products, for example, via the form of specialized hardware design. To answer this question, we observe that in optimization algorithms for solving linear inverse problems, signal priors usually appears in the form of proximal operators. Geometrically, the proximal operator *projects* the current estimate closer to the feasible sets (natural images) constrained by the signal prior. Thus, we propose to learn the proximal operator with a *deep projection model*. Once learned, the same network can be integrated into many standard optimization frameworks for solving arbitrary linear inverse problems of natural images.

Contributions. We make the following contributions.

- We propose a general framework that, for large image datasets, implicitly learns a signal prior in the form of a projection operator. When integrated into an alternating direction method of multipliers (ADMM) algorithm, the same proposed projection operator can solve challenging linear inverse problems.
- We identify the convergence conditions of the nonconvex ADMM with the proposed projection operator, and we use these conditions as the guidelines to design the proposed projection network.
- We empirically show that specially-trained networks are indeed sensitive to changes in the linear operators and noise in the linear measurements, and require retraining for effective usage. In contrast, the proposed method can be easily repurposed to small and big changes in the measurement operator without any retraining.

Limitations. A limitation of our method is its reliance on iterative methods; this is often computationally expensive when compared to specially-trained networks that are often non-iterative. Using a learned projection network also limits our ability to fine-tune the weight of the signal prior on-the-fly. Our convergence analysis is based on a perfectly learned projection network, which may not occur in practice. For very challenging problems like image inpainting with large missing regions, our current projection network may fail to produce satisfying results (see Figure 7).

2. Related Work

Given noisy linear measurements $\mathbf{y}$ and the corresponding linear operator A, which is usually underdetermined, the goal of linear inverse problems is to find a solution $\mathbf{x}$, such that $\mathbf{y} \approx A\mathbf{x}$ and $\mathbf{x}$ be a signal of interest, in our case, an image. Based on their strategies to deal with the underdetermined nature of the problem, algorithms for linear inverse problems can be roughly categorized into those using hand-designed signal priors and those learning from datasets. In this section, we briefly review some of these methods.

Hand-designed signal priors. Linear inverse problems are usually regularized by signal priors in a penalty form:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda\phi(\mathbf{x}), \tag{1}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}$ is the signal prior and λ is the non-negative weighting term. Signal priors constraining the sparsity of $\mathbf{x}$ in some transformation domain have been widely used in literatures. For example, since images are usually sparse after wavelet transformation or after taking gradient operations, a signal prior ϕ can be formulated as $\phi(\mathbf{x}) = \|W\mathbf{x}\|_1$, where W is a operator representing either wavelet transform, taking image gradient, or other hand-designed linear operation that produces sparse features from images [20]. Using signal priors of ℓ_1-norms provides two advantages. First, it forms a convex optimization problem and provides global optimality. The optimization problem can be solved efficiently with a variety of algorithms for convex optimization. Second, ℓ_1 priors enjoy many theoretical guarantees, thanks to results in compressive sensing [8]. For example, if the linear operator A satisfies conditions like the restricted isometry property and $W\mathbf{x}$ is sufficiently sparse, the optimization problem (1) provides the sparsest solution.

Despite their algorithmic and theoretical benefits, hand-designed priors are often too generic to constrain the solution set of the inverse problem (1) to be natural images — we can easily generate noise-like signals that have sparse wavelet coefficients or gradients.

Learning-based methods. The ever-growing number of images on the Internet enables state-of-the-art algorithms to deal with challenging problems that traditional methods are incapable of solving. For example, image inpainting and restoration can be performed by pasting image patches or transforming statistics of pixel values of similar images in a large dataset [15, 24]. Image denoising and super-resolution can be performed with dictionary learning methods that reconstruct image patches with sparse linear combinations of dictionary entries learned from datasets [4, 50]. Large datasets can also help learn end-to-end mappings from the linear measurement domain to the image domain. Given a linear operator A and a dataset $\mathcal{M} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the pairs $\{(\mathbf{x}_i, A\mathbf{x}_i)\}_{i=1}^n$ can be used to learn an inverse mapping $f \approx A^{-1}$ by minimizing the distance between $\mathbf{x}_i$ and $f(A\mathbf{x}_i)$, even when A is underdetermined. State-of-the-art methods usually parametrize the mapping functions with deep neural nets. For example, stacked auto-encoders and convolutional neural nets have been used to solve compressive sensing and image deblurring problems [28, 35, 36, 49, 51]. Recently, adversarial learning [21] has been demonstrated for its ability to solve many challenging image problems, such as image inpainting [38] and super-resolution [14, 29].

Despite its ability to solve challenging problems, learning end-to-end mappings has a major disadvantage — the number of mapping functions scales linearly with the number of problems. Since the datasets are generated based on specific operators As, these end-to-end mappings can only solve the given problems. Even if the problems change slightly, the mapping functions (neural nets) need to be retrained. For example, a mapping to solve $2\times$-super-resolution cannot be used directly to solve $3\times$- or $4\times$-super-resolution with satisfactory performance; it is even more difficult to re-purpose a mapping for image inpainting to solve super-resolution problems. This specificity of end-to-end mappings makes it costly to incorporate them into consumer products that need to deal with a variety of image processing applications.

Deep generative models. Another thread of research learns generative models from image datasets. Suppose we have a dataset containing samples of a distribution $P(\mathbf{x})$. We can estimate $P(\mathbf{x})$ and sample from the model [27, 43, 44], or directly generate new samples from $P(\mathbf{x})$ without explicitly estimating the distribution [21, 40]. Dave *et al*. [16] use a spatial long-short-term memory network to learn the distribution $P(\mathbf{x})$; to solve linear inverse problems, they solve a maximum a posteriori estimation — maximizing $P(\mathbf{x})$ over $\mathbf{x}$ subject to $\mathbf{y} = A\mathbf{x}$. Nguyen *et al*. [37] use a discriminative network and denoising autoencoders to implicitly learn the joint distribution between the image and its label $P(\mathbf{x}, y)$, and they generate new samples by sampling the joint distribution $P(\mathbf{x}, y)$, *i.e*., the network, with an approximated Metropolis-adjusted Langevin algorithm. To solve image inpainting, they replace the values of known pixels in sampled images and repeat the sampling process. As the proposed framework, these methods can be used to solve a wide variety of inverse problems. They use a probability framework and thereby can be considered orthogonal to the proposed framework, which is motivated by a geometric perspective.

3. One Network to Solve Them All

Signal priors play an important role in regularizing underdetermined inverse problems. As mentioned earlier, traditional priors constraining the sparsity of signals in gradient or wavelet bases are often too generic, in that we can easily create non-image signals satisfying these priors. Instead of using traditional signal priors, we propose to learn a prior from a large image dataset. Since the prior is learned directly from the dataset, it is tailored to the statistics of images in the dataset and, in principle, provide stronger regularization to the inverse problem. In addition, similar to traditional signal priors, the learned signal prior can be used to solve any linear inverse problems pertaining to images.

3.1. Problem formulation

The proposed framework is motivated by the optimization technique, alternating direction method of multipliers (ADMM) [7], that is widely used to solve linear inverse problems as defined in (1). A typical first step in ADMM is to separate a complicated objective into several simpler ones by variable splitting, *i.e*., introducing an additional variable $\mathbf{z}$ that is constrained to be equal to $\mathbf{x}$. This gives us the following optimization problem:

$$\min_{\mathbf{x},\mathbf{z}} \frac{1}{2}\|\mathbf{y} - A\mathbf{z}\|_2^2 + \lambda\,\phi(\mathbf{x}) \qquad \text{s.t.} \qquad \mathbf{x} = \mathbf{z}, \tag{2}$$

that is equivalent to the original problem (1). The scaled form of the augmented Lagrangian of (2) can be written as

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \frac{1}{2}\|\mathbf{y} - A\mathbf{z}\|_2^2 + \lambda\,\phi(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2,$$

where $\rho > 0$ is the penalty parameter of the constraint $\mathbf{x} = \mathbf{z}$, and $\mathbf{u}$ represents the dual variables divided by ρ. By alternately optimizing $\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{u})$ over $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{u}$, ADMM is composed of the following steps:

$$\mathbf{x}^{(k+1)} \leftarrow \arg\min_{\mathbf{x}} \frac{\rho}{2}\left\|\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}\right\|_2^2 + \lambda\,\phi(\mathbf{x}) \tag{3}$$

$$\mathbf{z}^{(k+1)} \leftarrow \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{y} - A\mathbf{z}\|_2^2 + \frac{\rho}{2}\left\|\mathbf{x}^{(k+1)} - \mathbf{z} + \mathbf{u}^{(k)}\right\|_2^2 \tag{4}$$

$$\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}.$$

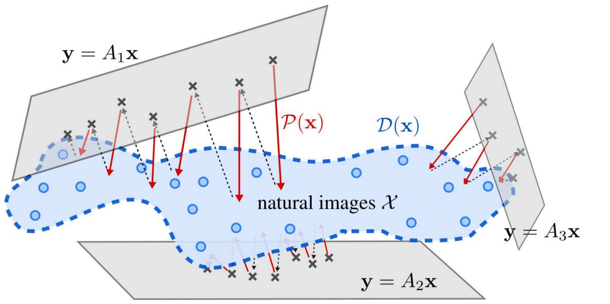

Figure 2: Given a large image dataset, the proposed framework learns a classifier $\mathcal{D}$ that fits a decision boundary of the natural image set. Based on $\mathcal{D}$, a projection network $\mathcal{P}(\mathbf{x})$:$\mathbb{R}^d \rightarrow \mathbb{R}^d$ is trained to fit the proximal operator of $\mathcal{D}$, which enables one to solve a variety of linear inverse problems using ADMM.

The update of $\mathbf{z}$ in (4) is a least squares problem and can be solved efficiently via conjugate gradient descent. The update of $\mathbf{x}$ in (3) is the proximal operator of the signal prior ϕ with penalty $\frac{\rho}{\lambda}$, denoted as $\mathbf{prox}_{\phi,\frac{\rho}{\lambda}}(\mathbf{v})$, where $\mathbf{v}$=$\mathbf{z}^{(k)}$$-$$\mathbf{u}^{(k)}$. When the signal prior uses ℓ_1-norm, the proximal operator is simply a soft-thresholding on $\mathbf{v}$. Notice that the ADMM algorithm separates the signal prior ϕ from the linear operator A. This enables us to learn a signal prior that can be used with any linear operator.

3.2. Learning a proximal operator

Since signal priors only appears in the form of proximal operators in ADMM, instead of explicitly learning a signal prior ϕ and solving the proximal operator in each step of ADMM, we propose to directly learn the proximal operator.

Let $\mathcal{X}$ represent the set of all natural images. The best signal prior is the indicator function of $\mathcal{X}$, denoted as $\mathcal{I}_{\mathcal{X}}(\cdot)$, and its corresponding proximal operator $\mathbf{prox}_{\mathcal{I}_{\mathcal{X}},\rho}(\mathbf{v})$ is a projection operator that projects $\mathbf{v}$ onto $\mathcal{X}$ from the geometric perspective— or equivalently, finding a $\mathbf{x} \in \mathcal{X}$ such that $\|\mathbf{x} - \mathbf{v}\|$ is minimized. However, we do not have the oracle indicator function $\mathcal{I}_{\mathcal{X}}(\cdot)$ in practice, so we cannot evaluate $\mathbf{prox}_{\mathcal{I}_{\mathcal{X}},\rho}(\mathbf{v})$ to solve the projection operation. Instead, we propose to train a classifier $\mathcal{D}$ with a large dataset whose decision function approximates $\mathcal{I}_{\mathcal{X}}$. Based on the learned classifier $\mathcal{D}$, we can learn a projection function $\mathcal{P}$ that maps a signal $\mathbf{v}$ to the set defined by the classifier. The learned projection function $\mathcal{P}$ can then replace the proximal operator (3), and we simply update $\mathbf{x}$ via

$$\mathbf{x}^{(k+1)} \leftarrow \mathcal{P}(\mathbf{z}^{(k)} - \mathbf{u}^{(k)}). \tag{5}$$

An illustration of the idea is shown in Figure 2.

There are some caveats for this approach. First, when the decision function of the classifier $\mathcal{D}$ is non-convex, the overall optimization becomes non-convex. For solving general non-convex optimization problems, the convergence result is not guaranteed. Based on the theorems for the convergence of non-convex ADMM [47], we provide the following theorem to the proposed ADMM framework.

Theorem 1. *Assume that the function $\mathcal{P}$ solves the proximal operator (3). If the gradient of $\phi(x)$ is Lipschitz continuous and with large enough ρ, the ADMM algorithm is guaranteed to attain a stationary point.*

The proof follows directly from [47] and we omit the details here. Although Theorem 1 only guarantees convergence to stationary points instead of the optimal solution as other non-convex formulations, it ensures that the algorithm will not diverge after several iterations. Second, we initialize the scaled dual variables $\mathbf{u}$ with zeros and $\mathbf{z}^{(0)}$ with the pseudo-inverse of the least-square term. Since we initialize $\mathbf{u}^0 = \mathbf{0}$, the input to the proximal operator $\mathbf{v}^{(k)}$=$\mathbf{z}^{(k)}$$-$$\mathbf{u}^{(k)} = \mathbf{z}^{(k)} + \sum_{i=1}^{k}\left(\mathbf{x}^{(i)} - \mathbf{z}^{(i)}\right) \approx \mathbf{z}^{(k)}$ resembles an image. Thereby, even though it is in general difficult to fit a projection function from any signal in $\mathbb{R}^d$ to the natural image space, we expect that the projection function only needs to deal with inputs that are close to images, and we train the projection function with slightly perturbed images from the dataset. Third, techniques like denoising autoencoders learn projection-like operators and, in principle, can be used in place of a proximal operator; however, our empirical findings suggest that ignoring the projection cost $\|\mathbf{v} - \mathcal{P}(\mathbf{v})\|^2$ and simply minimizing the reconstruction loss $\|\mathbf{x}_0 - \mathcal{P}(\mathbf{v})\|^2$, where $\mathbf{v}$ is a perturbed image from $\mathbf{x}_0$, leads to instability in the ADMM iterations.

3.3. Implementation details

An overview of the framework is illustrated in Figure 3. The projection operator $\mathcal{P}$ is implemented by a typical convolutional autoencoder, the classifier $\mathcal{D}$ and an auxiliary latent-space classifier $\mathcal{D}_\ell$ (whose use will be discussed below) are implemented by residual nets [25]. The architectures of the networks are discussed in the supplemental materials. Our code and trained models are online [1]. Below, we will discuss the choices made when designing the framework.

Choice of activation function. We use cross entropy loss as the discriminative loss to the classifiers. Since ϕ is the decision function of $\mathcal{D}$, we have $\phi(\mathbf{x}) = \log(\sigma(\mathcal{D}(\mathbf{x})))$, where σ is the sigmoid function. According to Theorem 1, we need the gradient of ϕ to be Lipschitz continuous. Thus, in order to make $\mathcal{D}$ differentiable, we choose the smooth exponential linear unit [12] as its activation function, instead of rectified linear units. To bound the gradients of $\mathcal{D}$ w.r.t. $\mathbf{x}$, we truncate the weights of the network after each iteration.

Image perturbation. While adding Gaussian noise may be the simplest method to perturb an image, we found that the projection network will easily overfit the Gaussian noise and become a dedicated Gaussian denoiser. Since during the ADMM process, the inputs to the projection network,

$\mathbf{z}^{(k)} - \mathbf{u}^{(k)}$, do not usually follow a Gaussian distribution, an overfitted projection network may fail to project the general signals produced by the ADMM process. To avoid overfitting, we generate perturbed images with two methods — adding Gaussian noise with spatially-varying standard deviations and smoothing the input images. The detailed implementation of image perturbation can be found in the supplemental material. We only use the smoothed images on ImageNet and MS-Celeb-1M datasets.

Training procedure. One way to train the classifier $\mathcal{D}$ is to feed $\mathcal{D}$ natural images from a dataset and their perturbed counterparts. Nevertheless, we expect the projected images produced by the projector $\mathcal{P}$ be closer to the dataset $\mathcal{M}$ (natural images) than the perturbed images. Therefore, we jointly train two networks using adversarial learning. The projector $\mathcal{P}$ is trained to minimize (3), that is, confusing the classifier $\mathcal{D}$ by projecting $\mathbf{v}$ to the natural image set defined by the decision boundary of $\mathcal{D}$. When the projector improves and generates outputs that are within or closer to the boundary, the classifier can be updated to tighten its decision boundary. Although we start from a different perspective from [21], the joint training procedure described above can also be understood as a two player game in adversarial learning, where the projector and the classifier have adversarial objectives.

Specifically, we optimize the projection network with the following objective function:

$$\min_{\boldsymbol{\theta}_{\mathcal{P}}} \sum_{\mathbf{x}\in\mathcal{M},\mathbf{v}\sim f(\mathbf{x})} \lambda_1\|\mathbf{x}-\mathcal{P}(\mathbf{x})\|^2 + \lambda_2\|\mathbf{x}-\mathcal{P}(\mathbf{v})\|^2 + \cdots$$
$$\cdots \lambda_3\|\mathbf{v}-\mathcal{P}(\mathbf{v})\|^2 - \lambda_4 \log\left(\sigma(\mathcal{D}_\ell\circ\mathcal{E}(\mathbf{v}))\right) - \lambda_5 \log\left(\sigma(\mathcal{D}\circ\mathcal{P}(\mathbf{v}))\right), \tag{6}$$

where $\boldsymbol{\theta}_{\mathcal{P}}$ is the parameters of the projection network $\mathcal{P}$, f is the function we used to generate perturbed images, and the first two terms in (6) are similar to (denoising) autoencoders and are added to help the training procedure. The remaining terms in (6) form the projection loss as we need in (3). We use two classifiers $\mathcal{D}$ and $\mathcal{D}_\ell$, for the output (image) space and the latent spaces of the projector ($\mathcal{E}(\mathbf{v})$ in Figure 3), respectively. The latent-space classifier $\mathcal{D}_\ell$ is added to further help the training procedure. Essentially, $\mathcal{D}_\ell$ encourages the perturbed images and their corresponding clean images to share the same encoding. More intuition about the latent-space classifier can be found in [32]. We also find that adding $\mathcal{D}_\ell$ helps the projector avoid overfitting. In all of our experiments, we set $\lambda_1 = 0.01, \lambda_3 = 0.005$, $\lambda_2 = 1.0$, $\lambda_4 = 0.0001$, and $\lambda_5 = 0.001$.

3.4. Relationship to other techniques

Many recent works solve linear inverse problems by unrolling the optimization process into the network architecture [3, 6, 22, 26]. Since the linear operator A is incorporated in the architecture, these networks are problem-specific. The

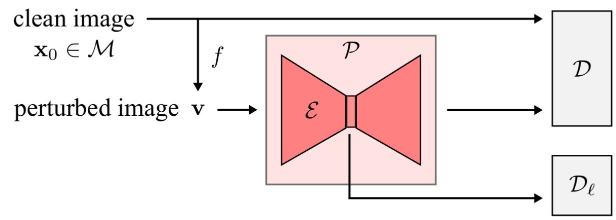

Figure 3: Block diagram of the framework. The adversarial learning is conducted on both image and latent spaces of $\mathcal{P}$.

proposed method is also similar to the denoising-based approximate message passing algorithm [34] and plug-and-play priors [45], which replace the proximal operator with an image denoiser.

Adversarial learning and denoising autoencoder. In terms of architecture, the proposed framework is very similar to adversarial learning [10, 21] and denoising autoencoder [38, 46]. Compared to adversarial learning, that matches the probability distributions of the dataset and the generated images, the proposed framework is based on the geometric perspective and the ADMM framework. Our use of the adversarial training is simply for learning a tighter decision boundary, based on the hypothesis that images generated by $\mathcal{P}$ should be closer, in terms of ℓ_2 distance, to $\mathcal{X}$ than the arbitrarily perturbed images. Compared to denoising autoencoder, the projection network $\mathcal{P}$ is encouraged to project perturbed images $\mathbf{x}_0 + \mathbf{n}$ to the closest $\mathbf{x}$ in $\mathcal{X}$, instead of the original image $\mathbf{x}_0$. In our empirical experience, the difference helps stabilize the ADMM process.

Other related methods. Many concurrent works also propose to solve generic linear inverse problems by learning proximal operators [33,48]. Meinhardt *et al.* [33] replace the proximal operator with a denoising network. Xiao *et al.* [48] use a modified multi-stage non-linear diffusion process [11] to learn the proximal operator.

Dave *et al.* [16] and Bora *et al.* [5] learn generative models of natural images and solve linear inverse problems by performing maximum a posteriori inference. Their algorithms need to compute the gradient of the networks in each iteration, which can be computationally expensive when the networks are very deep and complex. In contrast, the proposed method directly provides the solution to the $\mathbf{x}$-update (5) and is thus computationally efficient.

3.5. Limitations

Unlike traditional signal priors whose weights λ can be adjusted at the time of solving the optimization problem (1), the prior weight of the proposed framework is fixed once the projection network is trained. While an ideal projection operator should not be affected by the value of the prior weights, sometimes, it may be preferable to control the effect of the signal prior to the solution. In our experiments, we find

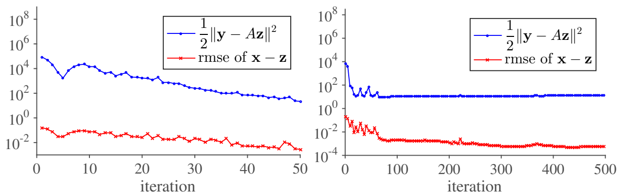

Figure 4: Convergence of the ADMM algorithms for compressive sensing (left) and scattered inpainting (right) of Figure 1.

that adjusting ρ sometimes has similar effects as adjusting λ.

The convergence analysis of ADMM in Theorem 1 is based on the assumption that the projection network can provide global optimum of (3). However, in practice the optimality is not guaranteed. While there are convergence analyses with inexact proximal operators, the general properties are too complicated to analyze for deep neural nets. In practice, we find that for problems like pixelwise inpainting, compressive sensing, $2\times$ super-resolution and scattered inpainting the proposed framework converges gracefully, as shown in Figure 4, but for more challenging problems like image inpainting with large blocks and $4\times$-super-resolution on ImageNet dataset, we sometimes need to stop the ADMM procedure early (by monitoring the residual $\|\mathbf{x}^{(k)} - \mathbf{z}^{(k)}\|$).

4. Experiments

We evaluate the proposed framework on the MNIST dataset [30], MS-Celeb-1M dataset [23], ImageNet dataset [41], and LabelMe dataset [42], whose descriptions are listed in Table 1.

For each of the datasets, we perform the following tasks:

(i) *Compressive sensing.* We use $m \times d$ random Gaussian matrices of different compression ($\frac{m}{d}$) as the linear operator A. The images are vectorized into d-dimensional vectors $\mathbf{x}$ and multiplied with the random Gaussian matrices to form $\mathbf{y}$.

(ii) *Pixelwise inpainting and denoising.* We randomly drop pixels (independent of channels) by filling zeros and add Gaussian noise with different standard deviations.

(iii) *Scattered inpainting.* We randomly drop 10 small blocks by filling zeros. Each block is of 10% width and height of the input.

(iv) *Blockwise inpainting.* We fill the center 30% region of the input images with zeros.

(v) *Super resolution.* We downsample the images into 50% and 25% of the original width and height using box-averaging algorithm.

Configurations of specially-trained networks. For each task (except for $4\times$-super resolution and for scattered inpainting), we train a deep neural net using context encoder [38] with adversarial training. For compressive sensing, we design the network based on the work of [35], which applies

dataset	# of samples	resolution
MNIST (hand-written digits)	60k + 10k	28×28×1
MS-Celeb (faces of 100k people)	8 million	64×64×3
ImageNet (natural images on the web)	1.2 million + 100k	64×64×3
LabelMe (natural images on the web)	$2,920 + 1,133$	64×64×3

Table 1: Datasets used to examine the proposed framework. For MS-Celeb-1M, we randomly select images of $73,678$ people as the training set and use the rest as the test set. The images are resized to the listed resolution before the training procedure.

$A^\top$ to the linear measurements and resize it into the image size to operate in image space. The measurement matrix A is a random Gaussian matrix and is fixed. For pixelwise inpainting and denoise, we randomly drop 50% of the pixels and add Gaussian noise with $\sigma = 0.5$ for each training instances. For blockwise inpainting, we drop a block with 30% size of the images at a random location in the images. For $2\times$-super resolution, we follow the work of Dong *et al.* [17] which first upsamples the low-resolution images to the target resolution using bicubic interpolation. We do not train a network for $4\times$-super resolution and for scattered inpainting — to demonstrate that the specially-trained networks do not generalize well to similar tasks. Since the inputs to the $2\times$-super resolution network are bicubic-upsampled images, we also apply the upsampling to $\frac{1}{4}$-resolution images and feed them to the same network. We also feed scattered inpainting inputs to the blockwise inpainting network.

Configurations of wavelet sparsity prior. We compare the proposed framework with the traditional signal prior using ℓ_1-norm of wavelet coefficients. We tune the weight of the ℓ_1 prior, λ, based on the dataset. For pure image denoising task, we will compare with the state-of-the-art algorithm BM3D [13] in the supplementals.

Results. For each of the experiments, we use $\rho = 0.3$ if not mentioned. The results on MNIST, MS-Celeb-1M, and ImageNet dataset are shown in Figures 5, 6, and 7, respectively. We also apply the same projection network trained on ImageNet dataset on the test set of LabelMe dataset. We list the statistics of peak-to-noise ratio (PSNR) values of the reconstruction outputs in Table 2. In addition, we use the same projection network on the image shown in Figure 1, which was not from any of the datasets above and can be found in [2]. To deal with the 384×512 image, when solving the projection operation (3), we apply the projection network on 64×64 patches and stitch the results. The reconstruction outputs are shown in Figure 1, and their statistics of each iteration of ADMM are shown in Figure 4.

As can be seen from the results, using the proposed projection operator/network learning from datasets enables us to solve more challenging problems than using the traditional wavelet sparsity prior. In Figures 5 and 6, while the tradi-

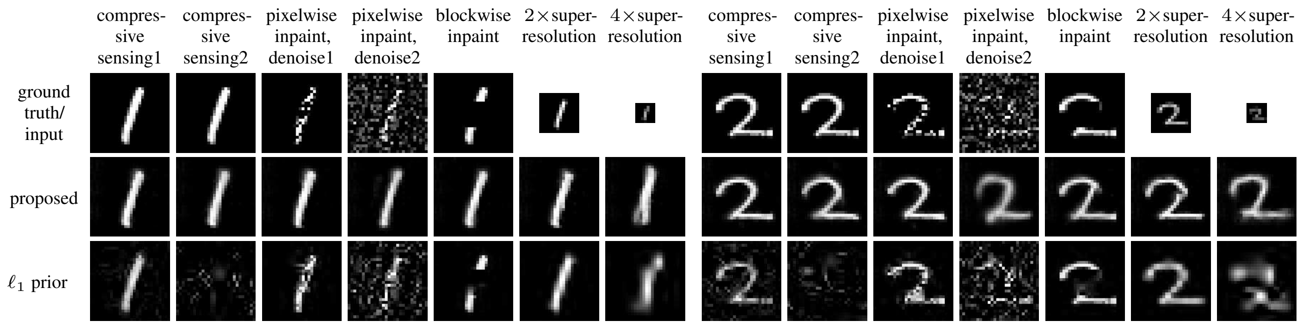

Figure 5: Results on MNIST dataset. Since the input of compressive sensing cannot be visualized, we show the ground truth instead. Compressive sensing 1 uses $\frac{m}{d} = 0.3$ and compressive sensing 2 uses $\frac{m}{d} = 0.03$. Pixelwise inpaint 1 drops 50% of the pixels, and pixelwise inpaint 2 drops 70% of the pixels and adds Gaussian noise with $\sigma = 0.3$. We use $\rho = 0.1$ for pixelwise inpainting and $\rho = 0.05$ for blockwise inpainting.

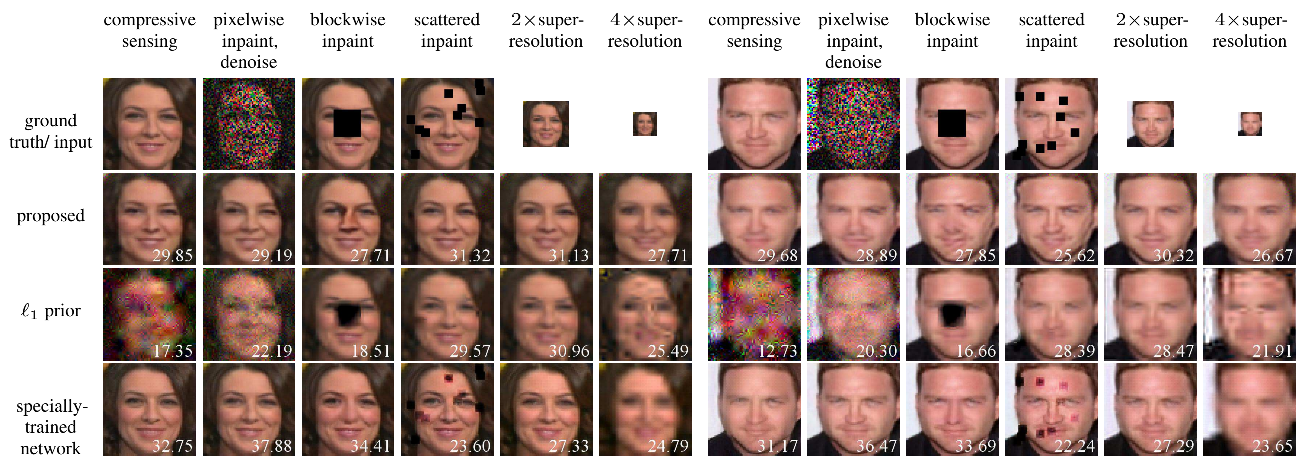

Figure 6: Results on MS-Celeb-1M dataset. The PSNR values are shown in the lower-right corner of each image. For compressive sensing, we test on $\frac{m}{d} = 0.1$. For pixelwise inpainting, we drop 50% of the pixels and add Gaussian noise with $\sigma = 0.1$. We use $\rho = 1.0$ on both super resolution tasks.

tional ℓ_1-prior of wavelet coefficients is able to reconstruct images from compressive measurements with $\frac{m}{d} = 0.3$, it fails to handle larger compression ratios like $\frac{m}{d} = 0.1$ and 0.03. Similar observations can be seen on pixelwise inpainting of different dropping probabilities and scattered and blockwise inpainting. In contrast, since the proposed projection network is tailored to the images in the datasets, it enables the ADMM algorithm to solve challenging problems like compressive sensing with small $\frac{m}{d}$ and blockwise inpainting on MS-Celeb dataset.

Robustness to changes in linear operator and to noise. Even though the specially-trained networks are able to generate state-of-the-art results on their designing tasks, they are unable to deal with similar problems, even with a slight change of the linear operator A. For example, as shown in Figure 6, the blockwise inpainting network is able to deal with much larger vacant regions; however, it overfits the problem and fails to fill contents to smaller blocks in scattered inpainting problems. The $2\times$-super resolution network also fails to reconstruct higher resolution images for $4\times$-super

task	**ℓ_1 prior**	**proposed**	**specially-trained**
compressive sensing (10×)	13.01 (±2.75)	25.43 (±3.74)	25.18 (±2.82)
pixelwise inpaint, denoise	20.68 (±1.65)	26.29 (±1.98)	30.13 (±1.66)
2× super-resolution	27.30 (±2.50)	27.11 (±3.21)	22.59 (±2.89)
scattered inpaint	27.85 (±2.58)	25.69 (±3.45)	18.30 (±2.55)

(a) ImageNet

task	**ℓ_1 prior**	**proposed**	**specially-trained**
compressive sensing (10×)	13.79 (±3.67)	27.34 (±5.15)	27.49 (±4.16)
pixelwise inpaint, denoise	21.72 (±2.17)	27.71 (±3.05)	30.93 (±1.96)
2× super-resolution	29.00 (±4.08)	28.52 (±4.64)	20.79 (±4.08)
scattered inpaint	30.17 (±3.96)	28.71 (±5.26)	18.65 (±3.12)

(b) LabelMe

Table 2: Average and standard deviation of PSNR values on 100k randomly chosen test images from ImageNet and the whole LabelMe test dataset. Note that we apply the same projection network trained with ImageNet on LabelMe. The similarity in the performance across the two datasets shows the robustness of the projection network.

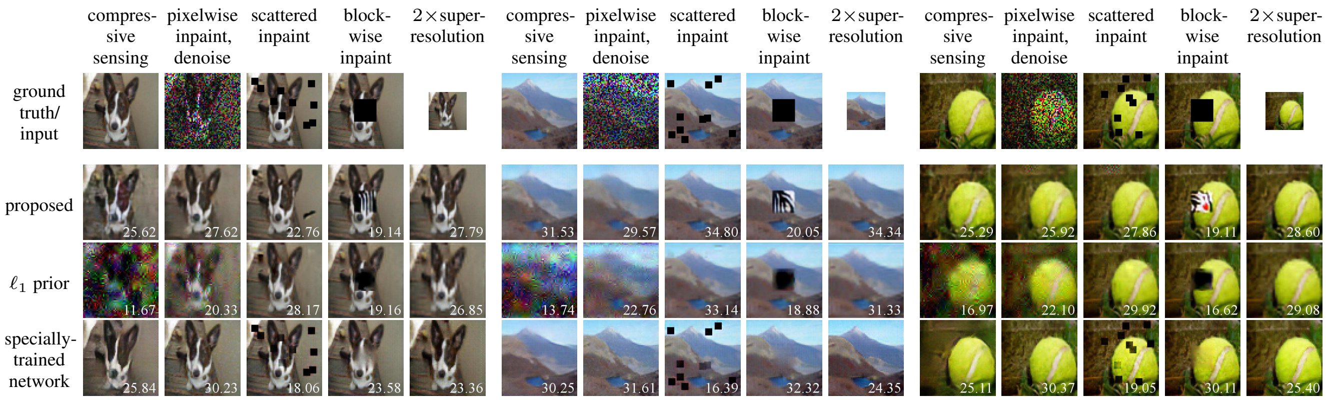

Figure 7: Results on ImageNet dataset. The PSNR values are shown in the lower-right corner of each image. Compressive sensing uses $\frac{m}{d} = 0.1$. For pixelwise inpainting, we drop 50% of the pixels and add Gaussian noise with $\sigma = 0.1$. We use $\rho = 0.05$ on scattered inpainting and $\rho = 0.5$ on super resolution.

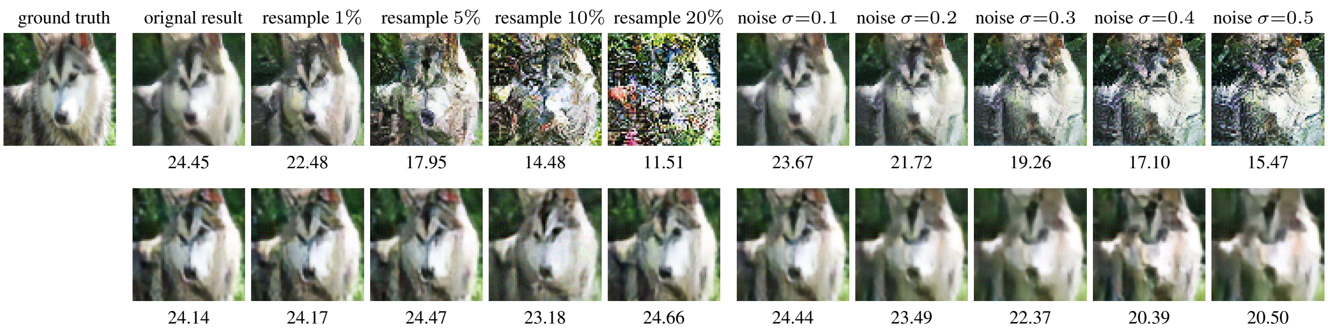

Figure 8: Comparison on the robustness to the linear operator A and noise on compressive sensing. The results of the specially-trained network and the proposed method are shown at the top and bottom row, respectively, along with their PSNR values. We use $\rho = 0.5$ for $\sigma = 0.2$, $\rho = 0.7$ for $\sigma = 0.3$, $\rho = 1.0$ for $\sigma = 0.4$, $\rho = 1.1$ for $\sigma = 0.5$, and $\rho = 0.3$ for all other cases.

resolution tasks, even though both inputs are upsampled using bicubic algorithm beforehand. We extend this argument with a compressive sensing example. We start from the random Gaussian matrix A_0 used to train the compressive sensing network, and we progressively resample elements in A_0 from the same distribution constructing A_0. As shown in Figure 8, once the portion of resampled elements increases, the specially-trained network fails to reconstruct the inputs, even though the new matrices are still Gaussian. The network also shows lower tolerance to Gaussian noise added to the clean linear measurements $\mathbf{y} = A_0\mathbf{x}_0$. In comparison, the proposed projector network is robust to changes of linear operators and noise.

Failure cases. The proposed projection network can fail to solve very challenging problems like the blockwise inpainting on ImageNet dataset, which has higher varieties in image contents than the other two datasets we test on. As shown in Figure 7, the proposed projection network tries to fill in random edges in the missing regions. In these cases, the projection network fails to project inputs to the natural image set, and thereby, violates our assumption in Theorem 1 and affects the overall ADMM framework. Even though increasing ρ can improve the convergence, it may produce low-quality, overly smoothed outputs.

5. Conclusion

In this paper, we propose a general framework to implicitly learn a signal prior — in the form of a projection operator — for solving generic linear inverse problems. The learned projection operator enjoys the flexibility of deep neural nets and wide applicability of traditional signal priors. With the ability to solve generic linear inverse problems like denoising, inpainting, super-resolution and compressive sensing, the proposed framework resolves the scalability of specially-trained networks. This characteristic significantly lowers the cost to design specialized hardware (ASIC for example) to solve image processing tasks. Thereby, we envision the projection network to be embedded into consumer devices like smart phones and autonomous vehicles to solve a variety of image processing problems.

References

[1] Implementation of the proposed method and trained models. https://github.com/image-science-lab/OneNet. 4

[2] One ring to rule them all. https://flic.kr/p/mGjhs7. 6

[3] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *arXiv preprint arXiv:1704.04058*, 2017. 5

[4] M. Aharon, M. Elad, and A. Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 3

[5] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. *arXiv preprint arXiv:1703.03208*, 2017. 5

[6] M. Borgerding and P. Schniter. Onsager-corrected deep learning for sparse linear inverse problems. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016. 5

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. 3

[8] E. J. Candes, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006. 2

[9] T. F. Chan, J. Shen, and H.-M. Zhou. Total variation wavelet inpainting. *Journal of Mathematical imaging and Vision*, 25(1):107–125, 2006. 2

[10] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[11] Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[12] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016. 4

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. In *Signal Processing with Adaptive Sparse Structured Representations*, 2009. 6

[14] R. Dahl, M. Norouzi, and J. Shlens. Pixel recursive super resolution. *arXiv preprint arXiv:1702.00783*, 2017. 3

[15] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. Image restoration using online photo collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 3

[16] A. Dave, A. K. Vadathya, and K. Mitra. Compressive image recovery using recurrent generative model. In *IEEE International Conference on Image Processing (ICIP)*, 2017. 3, 5

[17] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*, 2014. 2, 6

[18] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011. 2

[19] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995. 2

[20] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies. Data compression and harmonic analysis. *IEEE Transactions on Information Theory*, 44(6):2435–2476, 1998. 2

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3, 5

[22] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *International Conference on Machine Learning (ICML)*, 2010. 5

[23] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision (ECCV)*, 2016. 6

[24] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG)*, 26(3):4, 2007. 3

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[26] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017. 5

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 3

[28] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[30] G. Loosli, S. Canu, and L. Bottou. Training invariant support vector machines using selective sampling. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, 2007. 6

[31] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241, 2008. 2

[32] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 5

[33] T. Meinhardt, M. Möller, C. Hazirbas, and D. Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. *arXiv preprint arXiv:1704.03488*, 2017. 5

[34] C. A. Metzler, A. Maleki, and R. G. Baraniuk. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, 62(9):5117–5144, 2016. 5

[35] A. Mousavi and R. G. Baraniuk. Learning to invert: Signal recovery via deep convolutional networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, 2017. 2, 3, 6

[36] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. In *Allerton Conference on Communication, Control, and Computing*, 2015. 2, 3

[37] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[38] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 5, 6

[39] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003. 2

[40] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[42] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)*, 77(1):157–173, 2008. 6

[43] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009. 3

[44] L. Theis and M. Bethge. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems (NIPS)*. 2015. 3

[45] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2013. 5

[46] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008. 5

[47] Y. Wang, W. Yin, and J. Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *arXiv preprint arXiv:1511.06324*, 2015. 4

[48] L. Xiao, F. Heide, W. Heidrich, B. Schölkopf, and M. Hirsch. Discriminative transfer learning for general image restoration. *arXiv preprint arXiv:1703.09245*, 2017. 5

[49] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2, 3

[50] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010. 3

[51] Y.-T. Zhou, R. Chellappa, A. Vaid, and B. K. Jenkins. Image restoration using a neural network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1141–1151, 1988. 3