Compressed Sensing using Generative Models

Article · March 2017

CITATIONS
49

READS
126

4 authors, including:

Alexandros Dimakis
University of Texas at Austin
182 PUBLICATIONS **9,595** CITATIONS

SEE PROFILE

Compressed Sensing using Generative Models

Ashish Bora* Ajil Jalal† Eric Price‡ Alexandros G. Dimakis§

Abstract

The goal of compressed sensing is to estimate a vector from an underdetermined system of noisy linear measurements, by making use of prior knowledge on the structure of vectors in the relevant domain. For almost all results in this literature, the structure is represented by sparsity in a well-chosen basis. We show how to achieve guarantees similar to standard compressed sensing but without employing sparsity at all. Instead, we suppose that vectors lie near the range of a generative model $G : \mathbb{R}^k \to \mathbb{R}^n$. Our main theorem is that, if G is L-Lipschitz, then roughly $O(k \log L)$ random Gaussian measurements suffice for an ℓ_2/ℓ_2 recovery guarantee. We demonstrate our results using generative models from published variational autoencoder and generative adversarial networks. Our method can use 5-10x fewer measurements than Lasso for the same accuracy.

1 Introduction

Compressive or compressed sensing is the problem of reconstructing an unknown vector $x^* \in \mathbb{R}^n$ after observing $m < n$ linear measurements of its entries, possibly with added noise:

$$y = Ax^* + \eta,$$

where $A \in \mathbb{R}^{m \times n}$ is called the measurement matrix and $\eta \in \mathbb{R}^m$ is noise. Even without noise, this is an underdetermined system of linear equations, so recovery is impossible unless we make an assumption on the structure of the unknown vector x^*. We need to assume that the unknown vector is "natural," or "simple," in some application-dependent way.

The most common structural assumption is that the vector x^* is k-sparse in some known basis (or approximately k-sparse). Finding the sparsest solution to an underdetermined system of linear equations is NP-hard, but still convex optimization can provably recover the true sparse vector x^* if the matrix A satisfies conditions such as the Restricted Isometry Property (RIP) or the related Restricted Eigenvalue Condition (REC) [35, 7, 14, 6]. The problem is also called high-dimensional sparse linear regression and there is vast literature on establishing conditions for different recovery algorithms, different assumptions on the design of A and generalizations of RIP and REC for other structures, see *e.g.* [6, 33, 1, 30, 3].

This significant interest is justified since a large number of applications can be expressed as recovering an unknown vector from noisy linear measurements. For example, many tomography problems can be expressed in this framework: x^* is the unknown true tomographic image and the linear measurements are obtained by x-ray or other physical sensing system that produces sums or more general linear projections of the unknown pixels. Compressed sensing has been studied extensively for medical applications including computed tomography (CT) [8], rapid MRI [31] and neuronal spike train recovery [21]. Another impressive application is the "single pixel camera" [15], where digital micro-mirrors provide linear combinations to a single pixel sensor that then uses compressed sensing reconstruction algorithms to reconstruct an image. These results have been extended by combining sparsity with additional structural assumptions [4, 22], and by generalizations such as translating sparse vectors into low-rank matrices [33, 3, 17]. These

*University of Texas at Austin, Department of Computer Science, email: ashish.bora@utexas.edu
†University of Texas at Austin, Department of Electrical and Computer Engineering, email: ajiljalal@utexas.edu
‡University of Texas at Austin, Department of Computer Science, email: ecprice@cs.utexas.edu
§University of Texas at Austin, Department of Electrical and Computer Engineering, email: dimakis@austin.utexas.edu

results can improve performance when the structural assumptions fit the sensed signals. Other works perform "dictionary learning," seeking overcomplete bases where the data is more sparse (see [9] and references therein).

In this paper instead of relying on sparsity, we use structure from a *generative model*. Recently, several neural network based generative models such as variational auto-encoders (VAEs) [26] and generative adversarial networks (GANs) [19] have found success at modeling data distributions. In these models, the generative part learns a mapping from a low dimensional representation space $z \in \mathbb{R}^k$ to the high dimensional sample space $G(z) \in \mathbb{R}^n$. While training, this mapping is encouraged to produce vectors that resemble the vectors in the training dataset. We can therefore use any pre-trained generator to approximately capture the notion of an vector being "natural" in our domain: the generator defines a probability distribution over vectors in sample space and tries to assign higher probability to more likely vectors, for the dataset it has been trained on. We expect that vectors "natural" to our domain will be close to some point in the support of this distribution, *i.e.*, in the range of G.

Our Contributions: We present an algorithm that uses generative models for compressed sensing. Our algorithm simply uses gradient descent to optimize the representation $z \in \mathbb{R}^k$ such that the corresponding image $G(z)$ has small measurement error $\|AG(z) - y\|_2^2$. While this is a nonconvex objective to optimize, we empirically find that gradient descent works well, and the results can significantly outperform Lasso with relatively few measurements.

We obtain theoretical results showing that, as long as gradient descent finds a good approximate solution to our objective, our output $G(z)$ will be almost as close to the true x^* as the closest possible point in the range of G.

The proof is based on a generalization of the Restricted Eigenvalue Condition (REC) that we call the Set-Restricted Eigenvalue Condition (S-REC). Our main theorem is that if a measurement matrix satisfies the S-REC for the range of a given generator G, then the measurement error minimization optimum is close to the true x^*. Furthermore, we show that random Gaussian measurement matrices satisfy the S-REC condition with high probability for large classes of generators. Specifically, for d-layer neural networks such as VAEs and GANs, we show that $O(kd \log n)$ Gaussian measurements suffice to guarantee good reconstruction with high probability. One result, for ReLU-based networks, is the following:

Theorem 1.1. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a generative model from a d-layer neural network using ReLU activations. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(kd \log n)$, scaled so $A_{i,j} \sim N(0, 1/m)$. For any $x^* \in \mathbb{R}^n$ and any observation $y = Ax^* + \eta$, let $\widehat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum. Then with $1 - e^{-\Omega(m)}$ probability,

$$\|G(\widehat{z}) - x^*\|_2 \leq 6 \min_{z^* \in \mathbb{R}^k} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon.$$

Let us examine the terms in our error bound in more detail. The first two are the minimum possible error of any vector in the range of the generator and the norm of the noise; these are necessary for such a technique, and have direct analogs in standard compressed sensing guarantees. The third term ϵ comes from gradient descent not necessarily converging to the global optimum; empirically, ϵ does seem to converge to zero, and one can check post-observation that this is small by computing the upper bound $\|y - AG(\widehat{z})\|_2$.

While the above is restricted to ReLU-based neural networks, we also show similar results for arbitrary L-Lipschitz generative models, for $m \approx O(k \log L)$. Typical neural networks have poly(n)-bounded weights in each layer, so $L \leq n^{O(d)}$, giving for all activation functions the same $O(kd \log n)$ sample complexity as for ReLU networks.

Theorem 1.2. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be an L-Lipschitz function. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(k \log \frac{Lr}{\delta})$, scaled so $A_{i,j} \sim N(0, 1/m)$. For any $x^* \in \mathbb{R}^n$ and any observation $y = Ax^* + \eta$, let $\widehat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum over vectors with $\|\widehat{z}\|_2 \leq r$. Then with $1 - e^{-\Omega(m)}$ probability,

$$\|G(\widehat{z}) - x^*\|_2 \leq 6 \min_{\substack{z^* \in \mathbb{R}^k \\ \|z^*\|_2 \leq r}} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon + 2\delta.$$

The downside is two minor technical conditions: we only optimize over representations z with $\|z\|$ bounded by r, and our error gains an additive δ term. Since the dependence on these parameters is $\log(rL/\delta)$, and L is something like $n^{O(d)}$, we may set $r = n^{O(d)}$ and $\delta = 1/n^{O(d)}$ while only losing constant factors, making these conditions

very mild. In fact, generative models normally have the coordinates of z be independent uniform or Gaussian, so $\|z\| \approx \sqrt{k} \ll n^d$, and a constant signal-to-noise ratio would have $\|\eta\|_2 \approx \|x^*\| \approx \sqrt{n} \gg 1/n^d$.

We remark that, while these theorems are stated in terms of Gaussian matrices, the proofs only involve the distributional Johnson-Lindenstrauss property of such matrices. Hence the same results hold for matrices with subgaussian entries or fast-JL matrices [2].

2 Our Algorithm

All norms are 2-norms unless specified otherwise.

Let $x^* \in \mathbb{R}^n$ be the vector we wish to sense. Let $A \in \mathbb{R}^{m \times n}$ be the measurement matrix and $\eta \in \mathbb{R}^m$ be the noise vector. We observe the measurements $y = Ax^* + \eta$. Given y and A, our task is to find a reconstruction $\hat{x}$ close to x^*.

A generative model is given by a deterministic function $G : \mathbb{R}^k \to \mathbb{R}^n$, and a distribution P_Z over $z \in \mathbb{R}^k$. To generate a sample from the generator, we can draw $z \sim P_Z$ and the sample then is $G(z)$. Typically, we have $k \ll n$, *i.e.* the generative model maps from a low dimensional representation space to a high dimensional sample space.

Our approach is to find a vector in representation space such that the corresponding vector in the sample space matches the observed measurements. We thus define the objective to be

$$loss(z) = \|AG(z) - y\|^2 \tag{1}$$

By using any optimization procedure, we can minimize $loss(z)$ with respect to z. In particular, if the generative model G is differentiable, we can evaluate the gradients of the loss with respect to z using backpropagation and use standard gradient based optimizers. If the optimization procedure terminates at $\hat{z}$, our reconstruction for x^* is $G(\hat{z})$. We define the measurement error to be $\|AG(\hat{z}) - y\|^2$ and the reconstruction error to be $\|G(\hat{z}) - x^*\|^2$.

3 Related Work

Several recent lines of work explore generative models for reconstruction. The first line of work attempts to project an image on to the representation space of the generator. These works assume full knowledge of the image, and are special cases of the linear measurements framework where the measurement matrix A is identity. Excellent reconstruction results with SGD in the representation space to find an image in the generator range have been reported by [28] with stochastic clipping and [11] with logistic measurement loss. A different approach is introduced in [16] and [12]. In their method, a recognition network that maps from the sample space vector x to the representation space vector z is learned jointly with the generator in an adversarial setting.

A second line of work explores reconstruction with structured partial observations. The inpainting problem consists of predicting the values of missing pixels given a part of the image. This is a special case of linear measurements where each measurement corresponds to an observed pixel. The use of Generative models for this task has been studied in [38], where the objective is taken to be a combination of L_1 error in measurements and a perceptual loss term given by the discriminator. Super-resolution is a related task that attempts to increase the resolution of an image. We can view this problem as observing local spatial averages of the unknown higher resolution image and hence cast this as another special case of linear measurements. For prior work on super-resolution see *e.g.* [37, 13, 23] and references therein.

We also take note of the related work of [18] that connects model-based compressed sensing with the invertibility of Convolutional Neural Networks.

A related result appears in [5], which studies the measurement complexity of an RIP condition for smooth manifolds. This is analogous to our S-REC for the range of G, but the range of G is neither smooth (because of ReLUs) nor a manifold (because of self-intersection). Their recovery result was extended in [20] to unions of two manifolds.

4 Theoretical Results

We begin with a brief review of the Restricted Eigenvalue Condition (REC) in standard compressed sensing. The REC is a sufficient condition on A for robust recovery to be possible. The REC essentially requires that all "approximately sparse" vectors are far from the nullspace of the matrix A. More specifically, A satisfies REC for a constant $\gamma > 0$ if for all approximately sparse vectors x,

$$\|Ax\| \geq \gamma \|x\|. \tag{2}$$

It can be shown that this condition is sufficient for recovery of sparse vectors using Lasso. If one examines the structure of Lasso recovery proofs, a key property that is used is that the difference of any two sparse vectors is also approximately sparse (for sparsity up to $2k$). This is a coincidence that is particular to sparsity. By contrast, the difference of two vectors "natural" to our domain may not itself be natural. The condition we need is that the difference of any two natural vectors is far from the nullspace of A.

We propose a generalized version of the REC for a set $S \subseteq \mathbb{R}^n$ of vectors, the Set-Restricted Eigenvalue Condition (S-REC):

Definition 1. Let $S \subseteq \mathbb{R}^n$. For some parameters $\gamma > 0$, $\delta \geq 0$, a matrix $A \in \mathbb{R}^{m \times n}$ is said to satisfy the $S\text{-}REC(S, \gamma, \delta)$ if $\forall\, x_1, x_2 \in S$,

$$\|A(x_1 - x_2)\| \geq \gamma \|x_1 - x_2\| - \delta.$$

There are two main differences between the S-REC and the standard REC in compressed sensing. First, the condition applies to differences of vectors in an *arbitrary* set S of "natural" vectors, rather than just the set of approximately k-sparse vectors in some basis. This will let us apply the definition to S being the range of a generative model.

Second, we allow an additive slack term δ. This is necessary for us to achieve the S-REC when S is the output of general Lipschitz functions. Without it, the S-REC depends on the behavior of S at arbitrarily small scales. Since there are arbitrarily many such local regions, one cannot guarantee the existence of an A that works for all these local regions. Fortunately, as we shall see, poor behavior at a small scale δ will only increase our error by $\mathcal{O}(\delta)$.

The S-REC definition requires that for any two vectors in S, if they are significantly different (so the right hand side is large), then the corresponding measurements should also be significantly different (left hand side). Hence we can hope to approximate the unknown vector from the measurements, if the measurement matrix satisfies the S-REC.

But how can we find such a matrix? To answer this, we present two lemmas showing that random Gaussian matrices of relatively few measurements m satisfy the S-REC for the outputs of large and practically useful classes of generative models $G : \mathbb{R}^k \to \mathbb{R}^n$.

In the first lemma, we assume that the generative model $G(\cdot)$ is L-Lipschitz, *i.e.*, $\forall\, z_1, z_2 \in \mathbb{R}^k$, we have

$$\|G(z_1) - G(z_2)\| \leq L \|z_1 - z_2\|.$$

Note that state of the art neural network architectures with linear layers, (transposed) convolutions, max-pooling, residual connections, and all popular non-linearities satisfy this assumption. In Lemma 8.5 in the Appendix we give a simple bound on L in terms of parameters of the network; for typical networks this is $n^{O(d)}$. We also require the input z to the generator to have bounded norm. Since generative models such as VAEs and GANs typically assume their input z is drawn with independent uniform or Gaussian inputs, this only prunes an exponentially unlikely fraction of the possible outputs.

Lemma 4.1. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be L-Lipschitz. Let

$$B^k(r) = \{z \mid z \in \mathbb{R}^k, \|z\| \leq r\}$$

be an L_2-norm ball in $\mathbb{R}^k$. For $\alpha < 1$, if

$$m = \Omega\left(\frac{k}{\alpha^2} \log \frac{Lr}{\delta}\right),$$

then a random matrix $A \in \mathbb{R}^{m \times n}$ with IID entries such that $A_{ij} \sim \mathcal{N}\left(0, \frac{1}{m}\right)$ satisfies the $S\text{-}REC(G(B^k(r)), 1-\alpha, \delta)$ with $1 - e^{-\Omega(\alpha^2 m)}$ probability.

All proofs, including this one, are deferred to Appendix A.

Note that even though we proved the lemma for an L_2 ball, the same technique works for any compact set.

For our second lemma, we assume that the generative model is a neural network with such that each layer is a composition of a linear transformation followed by a pointwise non-linearity. Many common generative models have such architectures. We also assume that all non-linearities are piecewise linear with at most two pieces. The popular ReLU or LeakyReLU non-linearities satisfy this assumption. We do not make any other assumption, and in particular, the magnitude of the weights in the network do not affect our guarantee.

Lemma 4.2. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a d-layer neural network, where each layer is a linear transformation followed by a pointwise non-linearity. Suppose there are at most c nodes per layer, and the non-linearities are piecewise linear with at most two pieces, and let

$$m = \Omega\left(\frac{1}{\alpha^2} kd \log c\right)$$

for some $\alpha < 1$. Then a random matrix $A \in \mathbb{R}^{m \times n}$ with IID entries $A_{ij} \sim \mathcal{N}(0, \frac{1}{m})$ satisfies the S-$REC(G(\mathbb{R}^k), 1-\alpha, 0)$ with $1 - e^{-\Omega(\alpha^2 m)}$ probability.

To show Theorems 1.1 and 1.2, we just need to show that the S-REC implies good recovery. In order to make our error guarantee relative to ℓ_2 error in the image space $\mathbb{R}^n$, rather than in the measurement space $\mathbb{R}^m$, we also need that A preserves norms with high probability [10]. Fortunately, Gaussian matrices (or other distributional JL matrices) satisfy this property.

Lemma 4.3. Let $A \in \mathbb{R}^{m \times n}$ by drawn from a distribution that (1) satisfies the S-$REC(S, \gamma, \delta)$ with probability $1-p$ and (2) has for every fixed $x \in \mathbb{R}^n$, $\|Ax\| \leq 2\|x\|$ with probability $1 - p$.

For any $x^* \in \mathbb{R}^n$ and noise η, let $y = Ax^* + \eta$. Let $\widehat{x}$ approximately minimize $\|y - Ax\|$ over $x \in S$, *i.e.*,

$$\|y - A\widehat{x}\| \leq \min_{x \in S} \|y - Ax\| + \epsilon.$$

Then,

$$\|\widehat{x} - x^*\| \leq \left(\frac{4}{\gamma} + 1\right) \min_{x \in S} \|x^* - x\| + \frac{1}{\gamma}\left(2\|\eta\| + \epsilon + \delta\right)$$

with probability $1 - 2p$.

Combining Lemma 4.1, Lemma 4.2, and Lemma 4.3 gives Theorems 1.1 and 1.2. In our setting, S is the range of the generator, and $\widehat{x}$ in the theorem above is the reconstruction $G(\widehat{z})$ returned by our algorithm.

5 Models

In this section we describe the generative models used in our experiments. We used two image datasets and two different generative model types (a VAE and a GAN). This provides some evidence that our approach can work with many types of models and datasets.

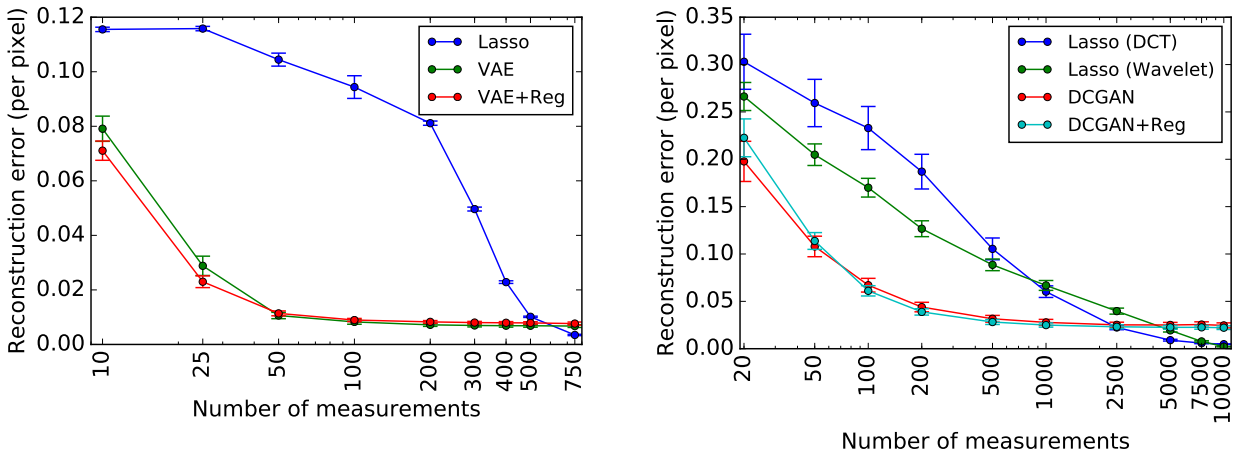
In our experiments, we found that it was helpful to add a regularization term $L(z)$ to the objective to encourage the optimization to explore more in the regions that are preferred by the respective generative models (see comparison to unregularized versions in Fig. 1). Thus the objective function we use for minimization is

$$\|AG(z) - y\|^2 + L(z).$$

Both VAE and GAN typically imposes an isotropic Gaussian prior on z. Thus $\|z\|^2$ is proportional to the negative log-likelihood under this prior. Accordingly, we use the following regularizer:

$$L(z) = \lambda \|z\|^2, \tag{3}$$

where λ measures the relative importance of the prior as compared to the measurement error.



(a) Results on MNIST

(b) Results on celebA

Figure 1: We compare the performance of our algorithm with baselines. We show a plot of per pixel reconstruction error as we vary the number of measurements. The vertical bars indicate 95% confidence intervals.

5.1 MNIST with VAE

The MNIST dataset consists of about $60,000$ images of handwritten digits, where each image is of size 28×28 [27]. Each pixel value is either 0 (background) or 1 (foreground). No pre-processing was performed. We trained VAE on this dataset. The input to the VAE is a vectorized binary image of input dimension 784. We set the size of the representation space $k = 20$. The recognition network is a fully connected $784 - 500 - 500 - 20$ network. The generator is also fully connected with the architecture $20 - 500 - 500 - 784$. We train the VAE using the Adam optimizer [25] with a mini-batch size 100 and a learning rate of 0.001.

We found that using $\lambda = 0.1$ in Eqn. (3) gave the best performance, and we use this value in our experiments.

The digit images are reasonably sparse in the pixel space. Thus, as a baseline, we use the pixel values directly for sparse recovery using Lasso. We set shrinkage parameter to be 0.1 for all the experiments.

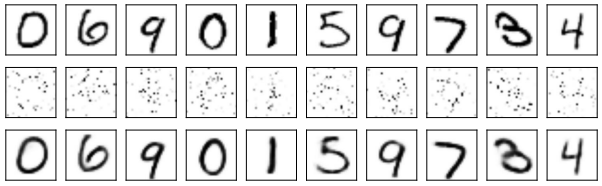
5.2 CelebA with DCGAN

CelebA is a dataset of more than $200,000$ face images of celebrities [29]. The input images were cropped to a 64×64 RGB image, giving $64 \times 64 \times 3 = 12288$ inputs per image. Each pixel value was scaled so that all values are between $[-1, 1]$. We trained a DCGAN [1] [34, 24] on this dataset. We set the input dimension $k = 100$ and use a standard normal distribution. The architecture follows that of [34]. The model was trained by one update to the discriminator and two updates to the generator per cycle. Each update used the Adam optimizer [25] with minibatch size 64, learning rate 0.0002 and $\beta_1 = 0.5$.

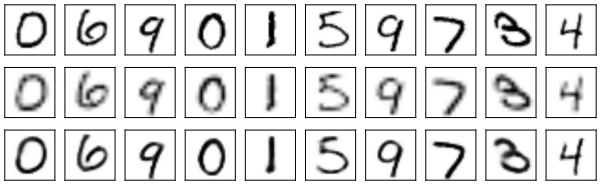
We found that using $\lambda = 0.001$ in Eqn. (3) gave the best results and thus, we use this value in our experiments.

For baselines, we perform sparse recovery using Lasso on the images in two domains: (a) 2D Discrete Cosine Transform (2D-DCT) and (b) 2D Daubechies-1 Wavelet Transform (2D-DB1). While the we provide Gaussian measurements of the original pixel values, the L_1 penalty is on either the DCT coefficients or the DB1 coefficients of each color channel of an image. For all experiments, we set the shrinkage parameter to be 0.1 and 0.00001 respectively for 2D-DCT, and 2D-DB1.

[1] Code reused from `https://github.com/carpedm20/DCGAN-tensorflow`



(a) We show original images (top row) and reconstructions by Lasso (middle row) and our algorithm (bottom row).



(b) We show original images (top row), low resolution version of original images (middle row) and reconstructions (last row).

Figure 2: Results on MNIST. Reconstruction with 100 measurements (left) and Super-resolution (right)

6 Experiments and Results

6.1 Reconstruction from Gaussian measurements

We take A to be a random matrix with IID Gaussian entries with zero mean and standard deviation of $1/m$. Each entry of noise vector η is also an IID Gaussian random variable. We compare performance of different sensing algorithms qualitatively and quantitatively. For quantitative comparison, we use the reconstruction error = $\|\hat{x} - x^*\|^2$, where $\hat{x}$ is an estimate of x^* returned by the algorithm. In all cases, we report the results on a held out test set, unseen by the generative model at training time.

6.1.1 MNIST

The standard deviation of the noise vector is set such that $\sqrt{\mathbb{E}[\|\eta\|^2]} = 0.1$. We use Adam optimizer [25], with a learning rate of 0.01. We do 10 random restarts with 1000 steps per restart and pick the reconstruction with best measurement error.

In Fig. 1a, we show the reconstruction error as we change the number of measurements both for Lasso and our algorithm. We observe that our algorithm is able to get low errors with far fewer measurements. For example, our algorithm's performance with 25 measurements matches Lasso's performance with 400 measurements. Fig. 2a shows sample reconstructions by Lasso and our algorithm.

However, our algorithm is limited since its output is constrained to be in the range of the generator. After 100 measurements, our algorithm's performance saturates, and additional measurements give no additional performance. Since Lasso has no such limitation, it eventually surpasses our algorithm, but this takes more than 500 measurements of the 784-dimensional vector. We expect that a more powerful generative model with representation dimension $k > 20$ can make better use of additional measurements.

6.1.2 celebA

The standard deviation of entries in the noise vector is set such that $\sqrt{\mathbb{E}[\|\eta\|^2]} = 0.01$. We optimize use Adam optimizer [25], with a learning rate of 0.1. We do 2 random restarts with 500 update steps per restart and pick the reconstruction with best measurement error.

In Fig. 1b, we show the reconstruction error as we change the number of measurements both for Lasso and our algorithm. In Fig. 3 we show sample reconstructions by Lasso and our algorithm. We observe that our algorithm is able to produce reasonable reconstructions with as few as 500 measurements, while the output of the baseline algorithms is quite blurry. Similar to the results on MNIST, if we continue to give more measurements, our algorithm saturates, and for more than 5000 measurements, Lasso gets a better reconstruction. We again expect that a more powerful generative model with $k > 100$ would perform better in the high-measurement regime.

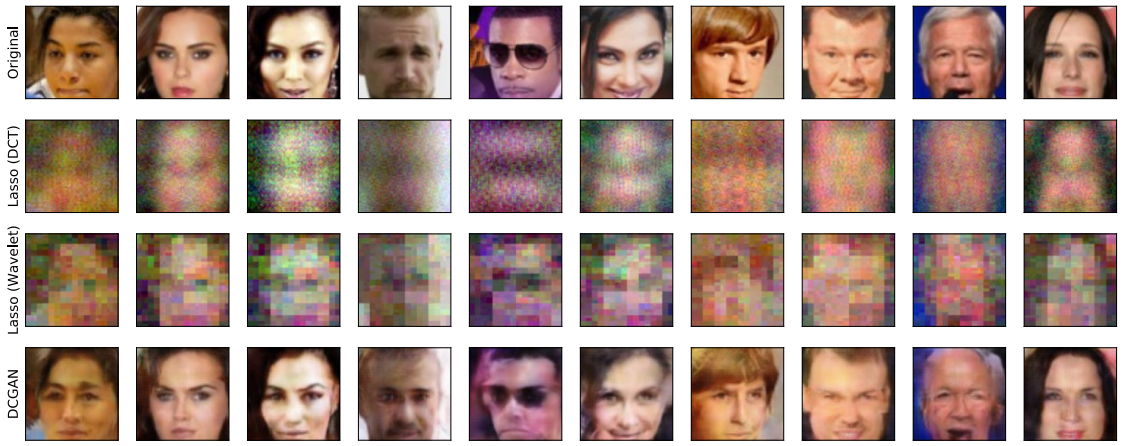


Figure 3: Reconstruction results on celebA with $m = 500$ measurements (of $n = 12288$ dimensional vector). We show original images (top row), and reconstructions by Lasso with DCT basis (second row), Lasso with wavelet basis (third row), and our algorithm (last row).

6.2 Super-resolution

Super-resolution is the task of constructing a high resolution image from a low resolution version of the same image. This problem can be thought of as special case of our general framework of linear measurements, where the measurements correspond to local spatial averages of the pixel values. Thus, we try to use our recovery algorithm to perform this task with measurement matrix A tailored to give only the relevant observations. We note that this measurement matrix may not satisfy the S-REC condition (with good constants γ and δ), and consequently, our theorems may not be applicable.

6.2.1 MNIST

We construct a low resolution image by spatial 2×2 pooling with a stride of 2 to produce a 14×14 image. These measurements are used to reconstruct the original 28×28 image. Fig. 2b shows reconstructions produced by our algorithm on images from a held out test set. We observe sharp reconstructions which closely match the fine structure in the ground truth.

6.2.2 celebA

We construct a low resolution image by spatial 4×4 pooling with a stride of 4 to produce a 16×16 image. These measurements are used to reconstruct the original 64×64 image. In Fig. 4 we show results on images from a held out test set. We see that our algorithm is able to fill in the details to match the original image.

6.3 Understanding sources of error

Although better than baselines, our reconstructions still admit some error. There are three sources of this error: (a) Representation error: the image being sensed is far from the range of the generator (b) Measurement error: The finite set of random measurements do not contain all the information about the unknown image (c) Optimization error: The optimization procedure did not find the best z.

In this section we present some experiments that suggest that the representation error is the dominant term. In our first experiment, we ensure that the representation error is zero, and try to minimize the sum of other two errors. In the second experiment, we ensure that the measurement error is zero, and try to minimize the sum of other two.

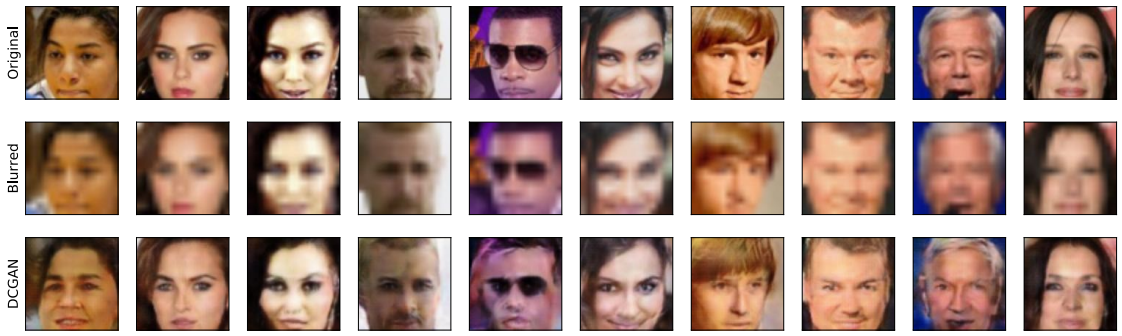


Figure 4: Super-resolution results on celebA. Top row has the original images. Second row shows the low resolution ($4x$ smaller) version of the original image. Last row shows the images produced by our algorithm.

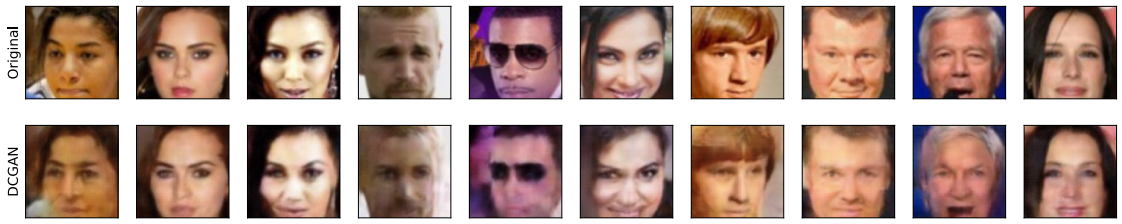
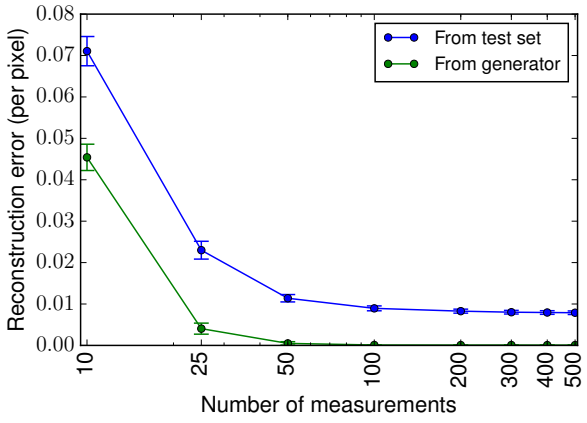
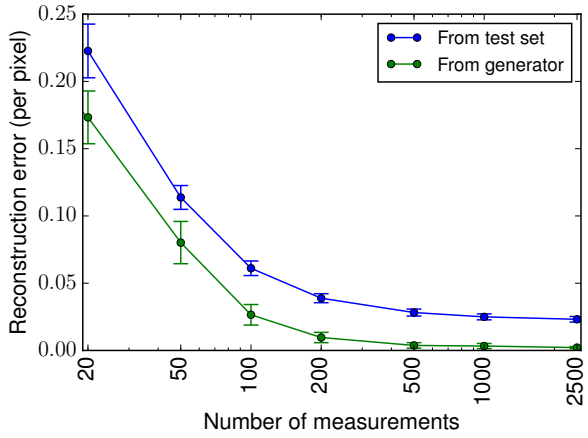


Figure 5: Results on the representation error experiments on celebA. Top row shows original images and the bottom row shows closest images found in the range of the generator.



(a) Results on MNIST



(b) Results on celebA

Figure 6: Reconstruction error for images in the range of the generator. The vertical bars indicate 95% confidence intervals.

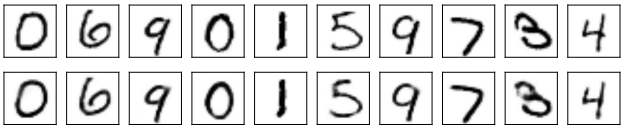


Figure 7: Results on the representation error experiments on MNIST. Top row shows original images and the bottom row shows closest images found in the range of the generator.

6.3.1 Sensing images from the range of the generator

Our first approach is to sense an image that *is* in the range of the generator. More concretely, we sample a z^* from P_Z. Then we pass it through the generator to get $x^* = G(z^*)$. Now, we pretend that this is a real image and try to sense that. This method eliminates the representation error and allows us to check if our gradient based optimization procedure is able to find z^* by minimizing the objective.

In Fig. 6a and Fig. 6b, we show the reconstruction error for images in the range of the generators trained on MNIST and celebA datasets respectively. We see that we get almost perfect reconstruction with very few measurements. This suggests that objective is being properly minimized and we indeed get $\hat{z}$ close to z^*. *i.e.* the sum of optimization error and the measurement error is not very large, in the absence of the representation error.

6.3.2 Quantifying representation error

We saw that in absence of the representation error, the overall error is small. However from Fig. 1, we know that the overall error is still non-zero. So, in this experiment, we seek to quantify the representation error, *i.e.*, how far are the real images from the range of the generator?

From the previous experiment, we know that the $\hat{z}$ recovered by our algorithm is close to z^*, the best possible value, if the image being sensed is in the range of the generator. Based on this, we make an assumption that this property is also true for real images. With this assumption, we get an estimate to the representation error as follows: We sample real images from the test set. Then we use the full image in our algorithm, *i.e.*, our measurement matrix A is identity. This eliminates the measurement error. Using these measurements, we get the reconstructed image $G(\hat{z})$ through our algorithm. The estimated representation error is then $\|G(\hat{z}) - x^*\|^2$. We repeat this procedure several times over randomly sampled images from our dataset and report average representation error values. The task of finding the closest image in the range of the generator has been studied in prior work [11, 16, 12].

On the MNIST dataset, we get average per pixel representation error of 0.005. The recovered images are shown in Fig. 7. In contrast with only 100 Gaussian measurements, we are able to get a per pixel reconstruction error of about 0.009.

On the celebA dataset, we get average per pixel representation error of 0.020. The recovered images are shown in Fig. 5. On the other hand, with only 500 Gaussian measurements, we get a per pixel reconstruction error of about 0.028.

These experiments suggest that the representation error is the major component of the total error. Thus, a more flexible generative model can help to decrease the overall error on both datasets.

7 Conclusion

We demonstrate how to perform compressed sensing using generative models from neural nets. These models can represent data distributions more concisely than standard sparsity models, while their differentiability allows for fast signal reconstruction. This will allow compressed sensing applications to make significantly fewer measurements.

Our theorems and experiments both suggest that, after relatively few measurements, the signal reconstruction gets close to the optimal within the range of the generator. To reach the full potential of this technique, one should use larger generative models as the number of measurements increase. Whether this can be expressed more concisely than by training multiple independent generative models of different sizes is an open question.

Generative models are an active area of research with ongoing rapid improvements. Because our framework applies to general generative models, this improvement will immediately yield better reconstructions with fewer measurements. We also believe that one could also use the performance of generative models for our task as one benchmark for the quality of different models.

Acknowledgements

We would like to thank Philipp Krähenbühl for helpful discussions.

References

[1] Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45, 2010.

[2] Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

[3] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

[4] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[5] Richard G Baraniuk and Michael B Wakin. Random projections of smooth manifolds. *Foundations of computational mathematics*, 9(1):51–77, 2009.

[6] Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

[7] Emmanuel J Candes, 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.

[8] Guang-Hong Chen, Jie Tang, and Shuai Leng. Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. *Medical physics*, 35(2):660–663, 2008.

[9] Guangliang Chen and Deanna Needell. Compressed sensing and dictionary learning. *Proceedings of Symposia in Applied Mathematics*, 73, 2016.

[10] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, 22(1):211–231, 2009.

[11] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *arXiv preprint arXiv:1611.05644*, 2016.

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[14] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[15] Marco F Duarte, Mark A Davenport, Dharmpal Takbar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.

[16] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[17] Rina Foygel and Lester Mackey. Corrupted sensing: Novel guarantees for separating structured signals. *IEEE Transactions on Information Theory*, 60(2):1223–1247, 2014.

[18] Anna C. Gilbert, Yi Zhang, Kibok Lee, Yuting Zhang, and Honglak Lee. Towards understanding the invertibility of convolutional neural networks. 2017.

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

[20] Chinmay Hegde and Richard G Baraniuk. Signal recovery on incoherent manifolds. *IEEE Transactions on Information Theory*, 58(12):7204–7214, 2012.

[21] Chinmay Hegde, Marco F Duarte, and Volkan Cevher. Compressive sensing recovery of spike trains using a structured sparsity model. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[22] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. A nearly-linear time framework for graph-structured sparsity. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 928–937, 2015.

[23] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[24] Taehoon Kim. A tensorflow implementation of "deep convolutional generative adversarial networks". https://github.com/carpedm20/DCGAN-tensorflow, 2017.

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

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017.

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[30] Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. In *Advances in Neural Information Processing Systems*, pages 2726–2734, 2011.

[31] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.

[32] Jiří Matoušek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2002.

[33] Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

[34] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[35] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[36] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[37] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[38] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.

8 Appendix A

Lemma 8.1. Given $S \subseteq \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, and $\gamma, \delta, \epsilon_1, \epsilon_2 > 0$, if matrix A satisfies the $S\text{-}REC(S, \gamma, \delta)$, then for any two $x_1, x_2 \in S$, such that $\|Ax_1 - y\| \leq \epsilon_1$ and $\|Ax_2 - y\| \leq \epsilon_2$, we have

$$\|x_1 - x_2\| \leq \frac{\epsilon_1 + \epsilon_2 + \delta}{\gamma}.$$

Proof.

$$\begin{aligned}
\|x_1 - x_2\| &\leq \frac{1}{\gamma}\left(\|Ax_1 - Ax_2\| + \delta\right), \\
&= \frac{1}{\gamma}\left(\|(Ax_1 - y) - (Ax_2 - y)\| + \delta\right), \\
&\leq \frac{1}{\gamma}\left(\|(Ax_1 - y)\| + \|(Ax_2 - y)\| + \delta\right), \\
&\leq \frac{\epsilon_1 + \epsilon_2 + \delta}{\gamma}.
\end{aligned}$$

□

8.1 Proof of Lemma 4.1

Definition 2. A random variable X is said to be $\text{subgamma}(\sigma, B)$ if $\forall \epsilon \geq 0$, we have

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq \epsilon\right) \leq 2 \max\left(e^{-\epsilon^2/(2\sigma^2)}, e^{-B\epsilon/2}\right).$$

Lemma 8.2. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be an L-Lipschitz function. Let $B^k(r)$ be the L_2-ball in $\mathbb{R}^k$ with radius r, $S = G(B^k(r))$, and M be a δ/L-net on $B^k(r)$ such that $|M| \leq k \log\left(\frac{4Lr}{\delta}\right)$. Let A be a $\mathbb{R}^{m \times n}$ random matrix with IID Gaussian entries with zero mean and variance $1/m$. If

$$m = \Omega\left(k \log \frac{Lr}{\delta}\right),$$

then for any $x \in S$, if $x' = \arg\min_{\widehat{x} \in G(M)} \|x - \widehat{x}\|$, we have $\|A(x - x')\| = \mathcal{O}(\delta)$ with probability $1 - e^{-\Omega(m)}$.

Note that for any given point x' in S, if we try to find its nearest neighbor of that point in an δ-net on S, then the difference between the two is at most the δ. In words, this lemma says that even if we consider measurements made on these points, *i.e.* a linear projection using a random matrix A, then as long as there are enough measurements, the difference between measurements is of the same order δ. If the point x' was in the net, then this can be easily achieved by Johnson-Lindenstrauss Lemma. But to argue that this is true for all x' in S, which can be an uncountably large set, we construct a chain of nets on S. We now present the formal proof.

Proof. Observe that $\frac{\|Ax\|^2}{\|x\|^2}$ is $\text{subgamma}\left(\frac{1}{\sqrt{m}}, \frac{1}{m}\right)$. Thus, for any $f > 0$,

$$\epsilon \geq 2 + \frac{4}{m} \log \frac{2}{f} \geq \max\left(\sqrt{\frac{2}{m} \log \frac{2}{f}}, \frac{2}{m} \log \frac{2}{f}\right)$$

is sufficient to ensure that

$$\mathbb{P}\left(\|Ax\| \geq (1 + \epsilon)\|x\|\right) \leq f.$$

Now, let $M = M_0 \subseteq M_1 \subseteq M_2, \cdots \subseteq M_l$ be a chain of epsilon nets of $B^k(r)$ such that M_i is a δ_i/L-net and $\delta_i = \delta_0/2^i$, with $\delta_0 = \delta$. We know that there exist nets such that

$$\log|M_i| \leq k \log\left(\frac{4Lr}{\delta_i}\right) \leq ik + k\log\left(\frac{4Lr}{\delta_0}\right).$$

Let $N_i = G(M_i)$. Then due to Lipschitzness of G, N_i's form a chain of epsilon nets such that N_i is a δ_i-net of $S = G(B^k(r))$, with $|N_i| = |M_i|$.

For $i \in \{0, 1, 2 \cdots, l-1\}$, let

$$T_i = \{x_{i+1} - x_i \mid x_{i+1} \in N_{i+1}, x_i \in N_i\}.$$

Thus,

$$\begin{aligned}
&|T_i| \leq |N_{i+1}||N_i|. \\
\implies &\log|T_i| \leq \log|N_{i+1}| + |\log|N_i|, \\
&\leq (2i+1)k + 2k\log\left(\frac{4Lr}{\delta_0}\right), \\
&\leq 3ik + 2k\log\left(\frac{4Lr}{\delta_0}\right).
\end{aligned}$$

Now assume $m = 3k\log\left(\frac{4Lr}{\delta_0}\right)$,

$$\log(f_i) = -(m + 4ik),$$

and

$$\begin{aligned}
\epsilon_i &= 2 + \frac{4}{m}\log\frac{2}{f_i}, \\
&= 2 + \frac{4}{m}\log 2 + 4 + \frac{16ik}{m}, \\
&= O(1) + \frac{16ik}{m}.
\end{aligned}$$

By choice of f_i and ϵ_i, we have $\forall i \in [l-1], \forall t \in T_i$,

$$\mathbb{P}\left(\|At\| > (1+\epsilon_i)\|t\|\right) \leq f_i.$$

Thus by union bound, we have

$$\mathbb{P}\left(\|At\| \leq (1+\epsilon_i)\|t\|, \forall i, \forall t \in T_i\right) \geq 1 - \sum_{i=0}^{l-1}|T_i|f_i.$$

Now,

$$\begin{aligned}
\log(|T_i|f_i) &= \log(|T_i|) + \log(f_i), \\
&\leq -k\log\left(\frac{4Lr}{\delta_0}\right) - ik, \\
&= -m/3 - ik. \\
\implies \sum_{i=0}^{l-1}|T_i|f_i &\leq e^{-m/3}\sum_{i=0}^{l-1}e^{-ik}, \\
&\leq e^{-m/3}\left(\frac{1}{1-e^{-1}}\right), \\
&\leq 2e^{-m/3}.
\end{aligned}$$

Observe that for any $x \in S$, we can write

$$x = x_0 + (x_1 - x_0) + (x_2 - x_1) \dots (x_l - x_{l-1}) + x^f.$$
$$x - x_0 = \sum_{i=0}^{l-1} (x_{i+1} - x_i) + x^f.$$

where $x_i \in N_i$ and $x_f = x - x_l$.

Since each $x_{i+1} - x_i \in T_i$, with probability at least $1 - 2e^{-m/3}$, we have

$$\begin{aligned}
\sum_{i=0}^{l-1} \|A(x_{i+1} - x_i)\| &= \sum_{i=0}^{l-1} (1 + \epsilon_i) \|(x_{i+1} - x_i)\|, \\
&\leq \sum_{i=0}^{l-1} (1 + \epsilon_i) \delta_i, \\
&= \delta_0 \sum_{i=0}^{l-1} \frac{1}{2^i} \left(O(1) + \frac{16ik}{m} \right), \\
&= O(\delta_0) + \delta_0 \frac{16k}{m} \sum_{i=0}^{l-1} \left(\frac{i}{2^i} \right), \\
&= O(\delta_0).
\end{aligned}$$

Now, $\|x^f\| = \|x - x_l\| \leq d_l = \frac{\delta_0}{2_l}$, and $\|x_{i+1} - x_i\| \leq \delta_i$ due to properties of epsilon-nets. We know that $\|A\| \leq 2 + \sqrt{n/m}$ with probability at least $1 - 2e^{-m/2}$ (Corollary 5.35 [36]). By setting $l = \log(n)$, we get that, $\|A\|\|x^f\| \leq \left(2 + \sqrt{\frac{n}{m}}\right) \frac{\delta_0}{2^l} = O(\delta_0)$ with probability $\geq 1 - 2e^{-m/2}$.

Combining these two results, and noting that it is possible to choose $x' = x_0$, we get that with probability $1 - e^{-\Omega(m)}$,

$$\begin{aligned}
\|A(x - x')\| &= \|A(x - x_0)\|, \\
&\leq \sum_{i=0}^{l-1} \|A(x_{i+1} - x_i)\| + \|Ax^f\|, \\
&= \mathcal{O}(\delta_0) + \|A\|\|x^f\|, \\
&= \mathcal{O}(\delta).
\end{aligned}$$

□

Lemma. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be L-Lipschitz. Let

$$B^k(r) = \{z \mid z \in \mathbb{R}^k, \|z\| \leq r\}$$

be an L_2-norm ball in $\mathbb{R}^k$. For $\alpha < 1$, if

$$m = \Omega\left(\frac{k}{\alpha^2} \log \frac{Lr}{\delta} \right),$$

then a random matrix $A \in \mathbb{R}^{m \times n}$ with IID entries such that $A_{ij} \sim \mathcal{N}\left(0, \frac{1}{m}\right)$ satisfies the S-$REC(G(B^k(r)), 1-\alpha, \delta)$ with $1 - e^{-\Omega(\alpha^2 m)}$ probability.

Proof. We construct a $\frac{\delta}{L}$-net, N, on $B^k(r)$. There exists a net such that

$$\log |N| \leq k \log \left(\frac{4Lr}{\delta} \right).$$

Since N is a $\frac{\delta}{L}$-cover of $B^k(r)$, due to the L-Lipschitz property of $G(\cdot)$, we get that $G(N)$ is a δ-cover of $G(B^k(r))$.

Let T denote the pairwise differences between the elements in $G(N)$, *i.e.*,

$$T = \{G(z_1) - G(z_2) \mid z_1, z_2 \in N\}.$$

Then,

$$\begin{aligned} & |T| \leq |N|^2, \\ \implies & \log|T| \leq 2\log|N|, \\ & \leq 2k \log\left(\frac{4Lr}{\delta}\right). \end{aligned}$$

For any $z, z' \in B^k$, $\exists\, z_1, z_2 \in N$, such that $G(z_1), G(z_2)$ are δ-close to $G(z)$ and $G(z')$ respectively. Thus, by triangle inequality,

$$\begin{aligned} \|G(z) - G(z')\| \leq & \|G(z) - G(z_1)\| + \\ & \|G(z_1) - G(z_2)\| + \\ & \|G(z_2) - G(z')\|, \\ \leq & \|G(z_1) - G(z_2)\| + 2\delta. \end{aligned}$$

Again by triangle inequality,

$$\begin{aligned} \|AG(z_1) - AG(z_2)\| \leq & \|AG(z_1) - AG(z)\| + \\ & \|AG(z) - AG(z')\| + \\ & \|AG(z') - AG(z_2)\|. \end{aligned}$$

Now, by Lemma 8.2, with probability $1 - e^{-\Omega(m)}$, $\|AG(z_1) - AG(z)\| = \mathcal{O}(\delta)$, and $\|AG(z') - AG(z_2)\| = \mathcal{O}(\delta)$. Thus,

$$\|AG(z_1) - AG(z_2)\| \leq \|AG(z) - AG(z')\| + \mathcal{O}(\delta).$$

By the Johnson-Lindenstrauss Lemma, for a fixed $x \in \mathbb{R}^n$, $\mathbb{P}\left[\|Ax\|^2 < (1-\alpha)\|x\|^2\right] < \exp(-\alpha^2 m)$. Therefore, we can union bound over all vectors in T to get

$$\mathbb{P}(\|Ax\|^2 \geq (1-\alpha)\|x\|^2,\ \forall x \in T) \geq 1 - e^{-\Omega(\alpha^2 m)}.$$

Since $\alpha < 1$, and $z_1, z_2 \in N$, $G(z_1) - G(z_2) \in T$, we have

$$\begin{aligned} (1-\alpha)\|G(z_1) - G(z_2)\| \leq & \sqrt{1-\alpha}\|G(z_1) - G(z_2)\|, \\ & \leq \|AG(z_1) - AG(z_2)\|. \end{aligned}$$

Combining the three results above we get that with probability $1 - e^{-\Omega(\alpha^2 m)}$,

$$\begin{aligned} (1-\alpha)\|G(z) - G(z')\| \leq & (1-\alpha)\|G(z_1) - G(z_2)\| + \mathcal{O}(\delta), \\ & \leq \|AG(z_1) - AG(z_2)\| + \mathcal{O}(\delta), \\ & \leq \|AG(z) - AG(z')\| + \mathcal{O}(\delta). \end{aligned}$$

Thus, A satisfies S-$REC(S, 1-\alpha, \delta)$ with probability $1 - e^{-\Omega(\alpha^2 m)}$.

□

8.2 Proof of Lemma 4.2

Lemma 8.3. Consider c different $k-1$ dimensional hyperplanes in $\mathbb{R}^k$. Consider the k-dimensional faces (hereafter called k-faces) generated by the hyperplanes, *i.e.* the elements in the partition of $\mathbb{R}^k$ such that relative to each hyperplane, all points inside a partition are on the same side. Then, the number of k-faces is $\mathcal{O}(c^k)$.

Proof. Proof is by induction, and follows [32].

Let $f(c,k)$ denote the number of $k-$faces generated in $\mathbb{R}^k$ by c different $(k-1)$-dimensional hyperplanes. As a base case, let $k=1$. Then $(k-1)$-dimensional hyperplanes are just points on a line. c points partition $\mathbb{R}$ into $c+1$ pieces. This gives $f(c,1)=\mathcal{O}(c)$.

Now, assuming that $f(c,k-1)=\mathcal{O}(c^{k-1})$ is true, we need to show $f(c,k)=\mathcal{O}(c^k)$. Assume we have $(c-1)$ different hyperplanes $H=\{h_1,h_2,\ldots,h_{c-1}\}\subset\mathbb{R}^k$, and a new hyperplane h_c is added. h_c intersects H at $(c-1)$ different $(k-2)$-faces given by $F=\{f_j \mid f_j=h_j\cap h_c, 1\leq j\leq (c-1)\}$. The $(k-2)$-faces in F partition h_c into $f(c-1,k-1)$ different $(k-1)$-faces. Additionally, each $(k-1)$-face in h_c divides an existing k-face into two. Hence the number of new k-faces introduced by the addition of h_c is $f(c-1,k-1)$. This gives the recursion

$$\begin{aligned} f(c,k) &= f(c-1,k)+f(c-1,k-1),\\ &= f(c-1,k)+\mathcal{O}(c^{k-1}),\\ &= \mathcal{O}(c^k). \end{aligned}$$

□

Lemma. Let $G:\mathbb{R}^k\to\mathbb{R}^n$ be a d-layer neural network, where each layer is a linear transformation followed by a pointwise non-linearity. Suppose there are at most c nodes per layer, and the non-linearities are piecewise linear with at most two pieces, and let

$$m=\Omega\left(\frac{1}{\alpha^2}kd\log c\right)$$

for some $\alpha<1$. Then a random matrix $A\in\mathbb{R}^{m\times n}$ with IID entries $A_{ij}\sim\mathcal{N}(0,\frac{1}{m})$ satisfies the $S\text{-}REC(G(\mathbb{R}^k),1-\alpha,0)$ with $1-e^{-\Omega(\alpha^2 m)}$ probability.

Proof. Consider the first layer of G. Each node in this layer can be represented as a hyperplane in $\mathbb{R}^k$, where the points on the hyperplane are those where the input to the node switches from one linear piece to the other. Since there are at most c nodes in this layer, by Lemma 8.3, the input space is partitioned by at most c different hyperplanes, into $\mathcal{O}(c^k)$ k-faces. Applying this over the d layers of G, we get that the input space $\mathbb{R}^k$ is partitioned into at most c^{kd} sets.

Recall that the non-linearities are piecewise linear, and the partition boundaries were made precisely at those points where the non-linearities change from one piece to another. This means that within each set of the input partition, the output is a linear function of the inputs. Thus $G(\mathbb{R}^k)$ is a union of c^{kd} different k-faces in $\mathbb{R}^n$.

We now use an oblivious subspace embedding to bound the number of measurements required to embed the range of $G(\cdot)$. For a single k-face $S\subseteq\mathbb{R}^n$, a random matrix $A\in\mathbb{R}^{m\times n}$ with IID entries such that $A_{ij}\sim\mathcal{N}\left(0,\frac{1}{m}\right)$ satisfies $S\text{-}REC(S,1-\alpha,0)$ with probability $1-e^{-\Omega(\alpha^2 m)}$ if $m=\Omega(k/\alpha^2)$.

Since the range of $G(\cdot)$ is a union of c^{kd} different k-faces, we can union bound over all of them, such that A satisfies the $S\text{-}REC(G(\mathbb{R}^k),1-\alpha,0)$ with probability $1-c^{kd}e^{-\Omega(\alpha^2 m)}$. Thus, we get that A satisfies the $S\text{-}REC(G(\mathbb{R}^k),1-\alpha,0)$ with probability $1-e^{-\Omega(\alpha^2 m)}$ if

$$m=\Omega\left(\frac{kd\log c}{\alpha^2}\right).$$

□

8.3 Proof of Lemma 4.3

Lemma. Let $A \in \mathbb{R}^{m \times n}$ by drawn from a distribution that (1) satisfies the S-$REC(S, \gamma, \delta)$ with probability $1-p$ and (2) has for every fixed $x \in \mathbb{R}^n$, $\|Ax\| \leq 2\|x\|$ with probability $1-p$. For any $x^* \in \mathbb{R}^n$ and noise η, let $y = Ax^* + \eta$. Let $\widehat{x}$ approximately minimize $\|y - Ax\|$ over $x \in S$, *i.e.*,

$$\|y - A\widehat{x}\| \leq \min_{x \in S} \|y - Ax\| + \epsilon.$$

Then

$$\|\widehat{x} - x^*\| \leq \left(\frac{4}{\gamma} + 1\right) \min_{x \in S} \|x^* - x\| + \frac{1}{\gamma}\left(2\|\eta\| + \epsilon + \delta\right)$$

with probability $1 - 2p$.

Proof. Let $\overline{x} = \arg\min_{x \in S} \|x^* - x\|$. Then we have by Lemma 8.1 and the hypothesis on $\widehat{x}$ that

$$\begin{aligned}
\|\overline{x} - \widehat{x}\| &\leq \frac{\|A\overline{x} - y\| + \|A\widehat{x} - y\| + \delta}{\gamma}, \\
&\leq \frac{2\|A\overline{x} - y\| + \epsilon + \delta}{\gamma}, \\
&\leq \frac{2\|A(\overline{x} - x^*)\| + 2\|\eta\| + \epsilon + \delta}{\gamma},
\end{aligned}$$

as long as A satisfies the S-REC, as happens with probability $1-p$. Now, since $\overline{x}$ and x^* are independent of A, by assumption we also have $\|A(\overline{x} - x^*)\| \leq 2\|\overline{x} - x^*\|$ with probability $1-p$. Therefore

$$\|x^* - \widehat{x}\| \leq \|\overline{x} - x^*\| + \frac{4\|\overline{x} - x^*\| + 2\|\eta\| + \epsilon + \delta}{\gamma}$$

as desired. □

8.4 Lipschitzness of Neural Networks

Lemma 8.4. Consider any two functions f and g. If f is L_f-Lipschitz and g is L_g-Lipschitz, then their composition $f \circ g$ is $L_f L_g$-Lipschitz.

Proof. For any two x_1, x_2,

$$\begin{aligned}
\|f(g(x_1)) - f(g(x_2))\| &\leq L_f \|g(x_1) - g(x_2)\|, \\
&\leq L_f L_g \|x_1 - x_2\|.
\end{aligned}$$

□

Lemma 8.5. If G is a d-layer neural network with at most c nodes per layer, all weights $\leq w_{\max}$ in absolute value, and M-Lipschitz non-linearity after each layer, then $G(\cdot)$ is L-Lipschitz with $L = (Mcw_{\max})^d$.

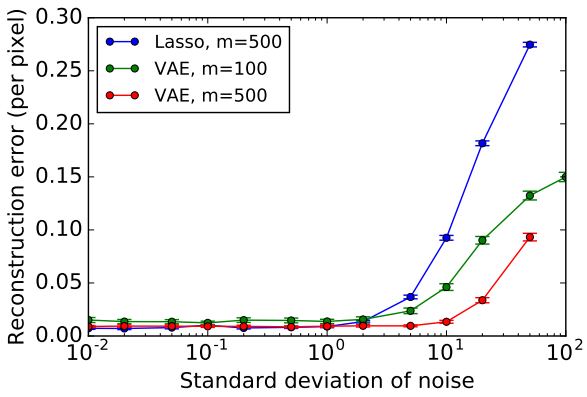
Proof. Consider any linear layer with input x, weight matrix W and bias vector b. Thus, $f(x) = Wx + b$. Now for any two x_1, x_2,

$$\begin{aligned}
\|f(x_1) - f(x_2)\| &= \|Wx_1 + b - Wx_2 + b\|, \\
&= \|W(x_1 - x_2)\|, \\
&\leq \|W\|\|(x_1 - x_2)\|, \\
&\leq cw_{\max}\|(x_1 - x_2)\|.
\end{aligned}$$

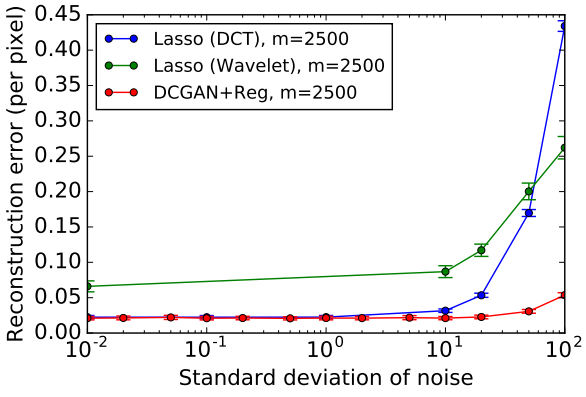
Let $f_i(\cdot), i \in [d]$ denote the function for the i-th layer in G. Since each layer is a composition of a linear function and a non-linearity, by Lemma 8.4, have that f_i is $Mcw_{\max}$-Lipschitz.

Since $G = f_1 \circ f_2 \circ \ldots f_d$, by repeated application of Lemma 8.4, we get that G is L-Lipschitz with $L = (Mcw_{\max})^d$. □

9 Appendix B



(a) Results on MNIST.



(b) Results on celebA.

Figure 8: Noise tolerance. We show a plot of per pixel reconstruction error as we vary the noise level ($\sqrt{\mathbb{E}[\|\eta\|^2]}$). The vertical bars indicate 95% confidence intervals.

9.1 Noise tolerance

To understand the noise tolerance of our algorithm, we do the following experiment: First we fix the number of measurements so that Lasso does as well as our algorithm. From Fig. 1a, and Fig. 1b we see that this point is at $m = 500$ for MNIST and $m = 2500$ for celebA. Now, we look at the performance as the noise level increases. Hyperparameters are kept fixed as we change the noise level for both Lasso and for our algorithm.

In Fig. 8a, we show the results on the MNIST dataset. In Fig. 8a, we show the results on celebA dataset. We observe that our algorithm has more noise tolerance than Lasso.

9.2 Other models

9.2.1 End to end training on MNIST

Instead of using a generative model to reconstruct the image, another approach is to learn from scratch a mapping that takes the measurements and outputs the original image. A major drawback of this approach is that it necessitates learning a new network if get a different set of measurements.

If we use a random matrix for every new image, the input to the network is essentially noise, and the network does not learn at all. Instead we are forced to use a fixed measurement matrix. We explore two approaches. First is to randomly sample and fix the measurement matrix and learn the rest of the mapping. In the second approach, we jointly optimize the measurement matrix as well.

We do this for 10, 20 and 30 measurements for the MNIST dataset. We did not use additive noise. The reconstruction errors are shown in Fig. 9. The reconstructions can be seen in Fig. 10.

9.3 More results

Here, we show more results on the reconstruction task, with varying number of measurements on both MNIST and celebA. Fig. 11 shows reconstructions on MNIST with 25, 100 and 400 measurements. Fig. 12, Fig. 13 and Fig. 14 show results on celebA dataset.

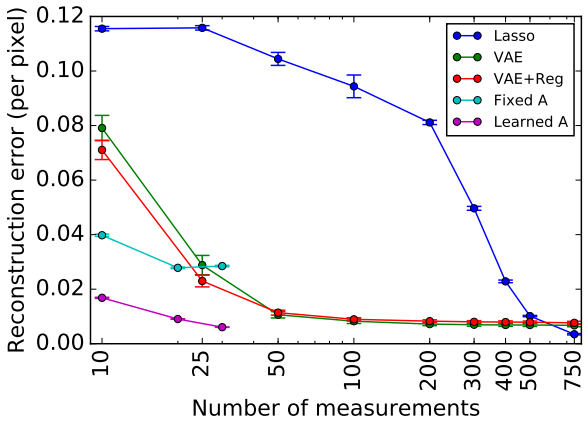


Figure 9: Results on End to End model on MNIST. We show per pixel reconstruction error vs number of measurements. 'Fixed A' and 'Learned A' are two end to end models. The end to end models get noiseless measurements, while the other models get noisy ones. The vertical bars indicate 95% confidence intervals.

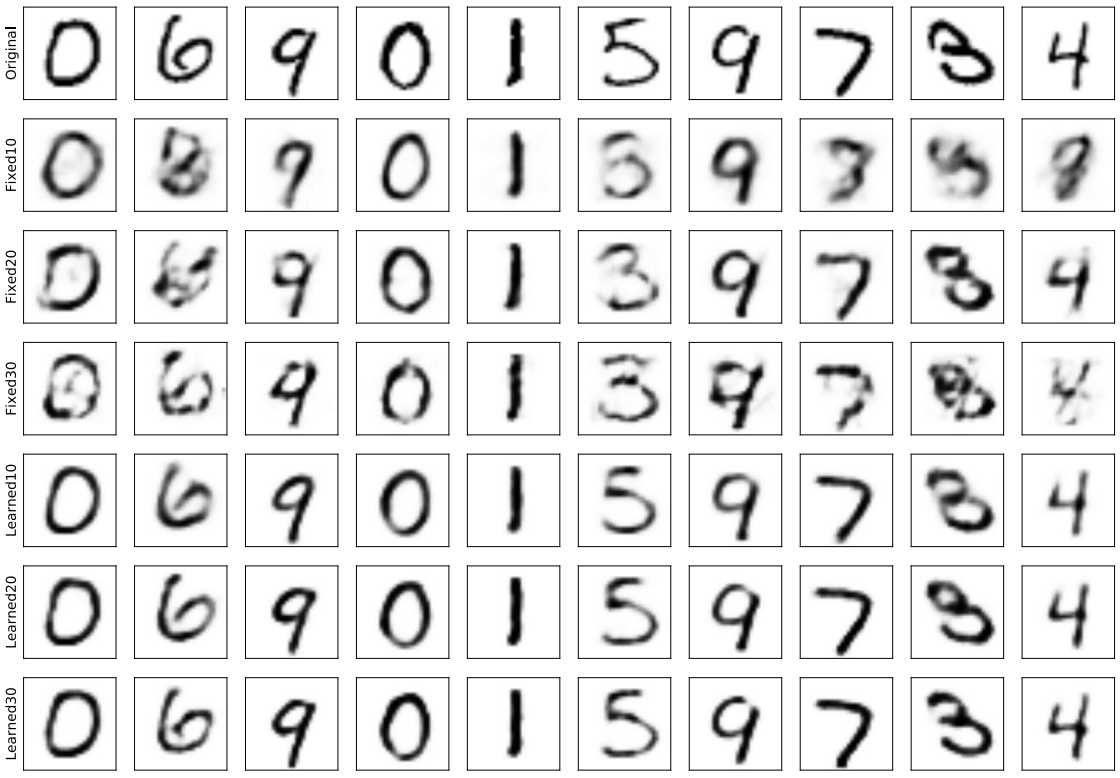
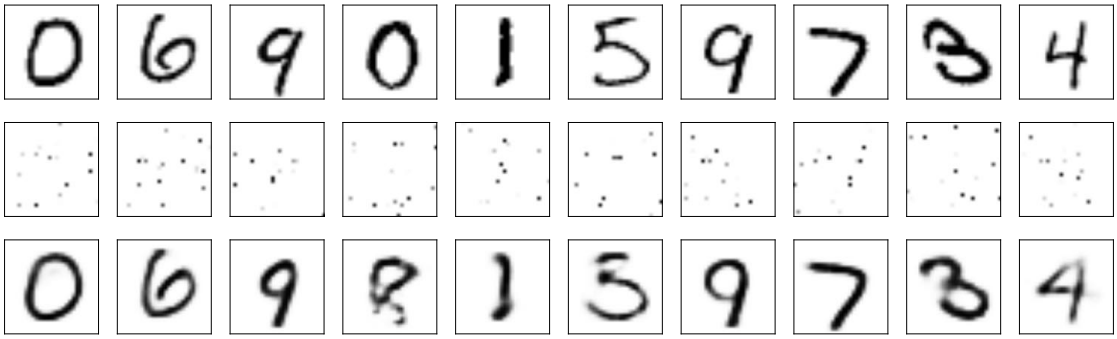
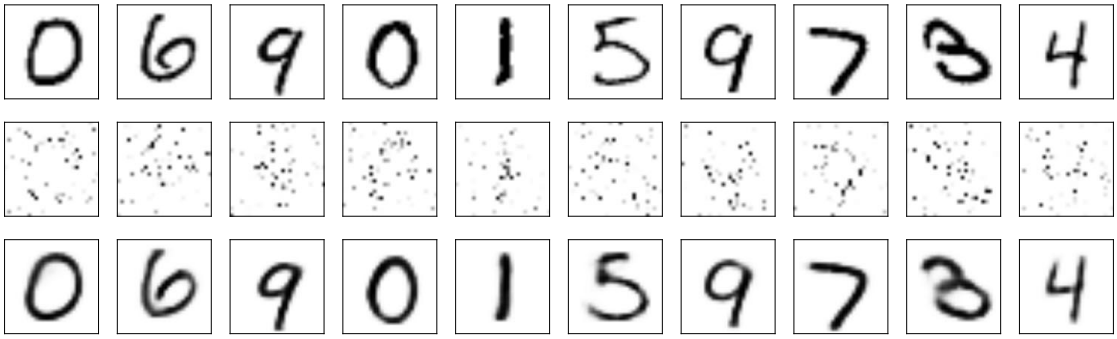


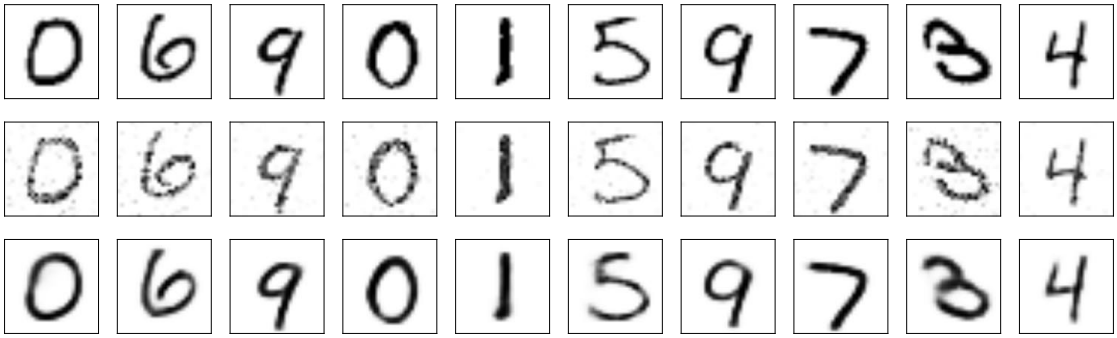
Figure 10: MNIST End to end learned model. Top row are original images. The next three are recovered by model with fixed random A, with 10, 20 and 30 measurements. Bottom three rows are with learned A and 10, 20 and 30 measurements.



(a) 25 measurements

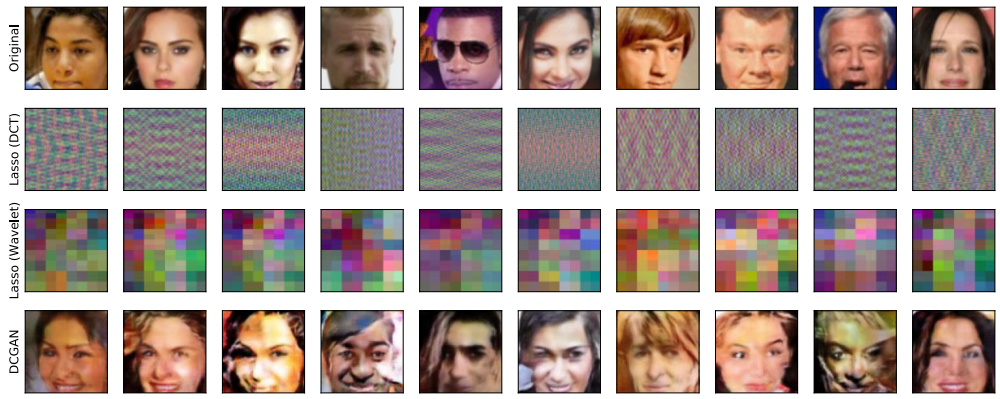


(b) 100 measurements

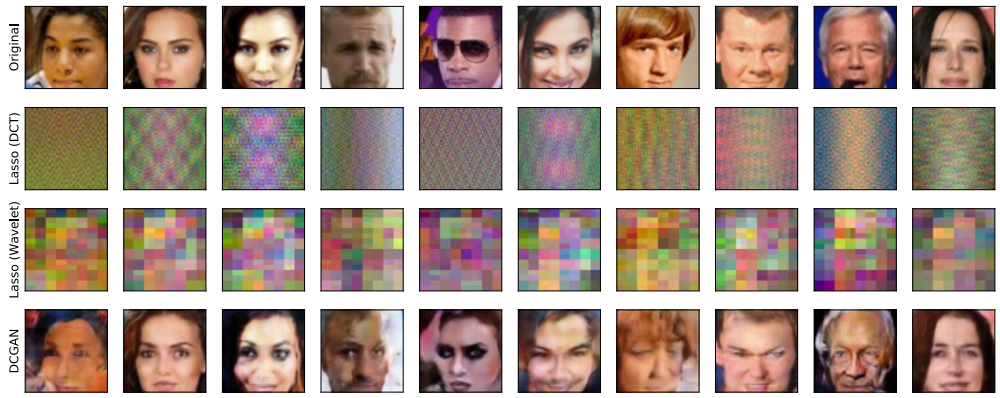


(c) 400 measurements

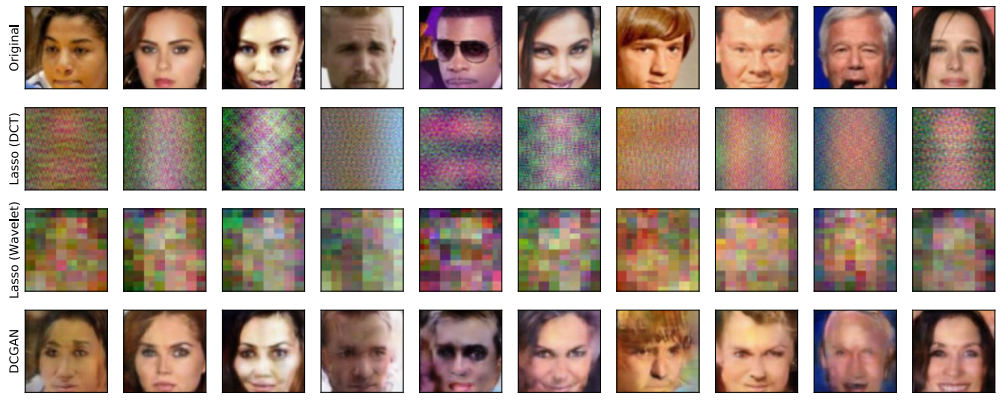
Figure 11: Reconstruction on MNIST. In each image, top row is ground truth, middle row is Lasso, bottom row is our algorithm.



(a) 50 measurements

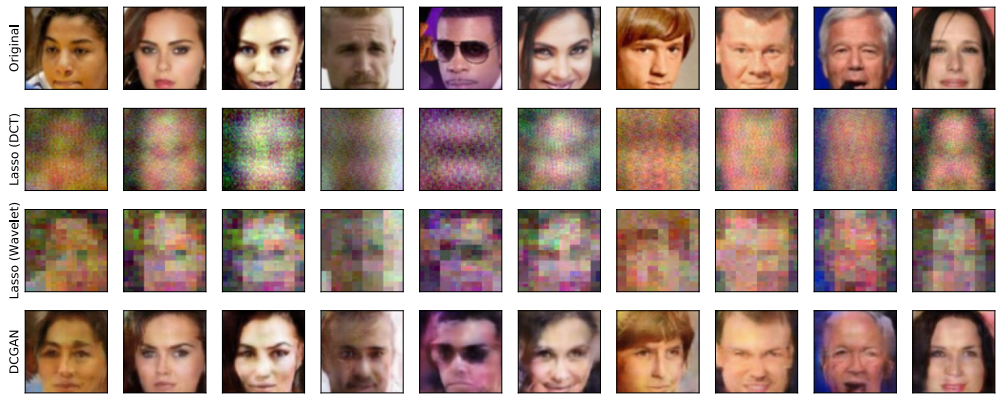


(b) 100 measurements

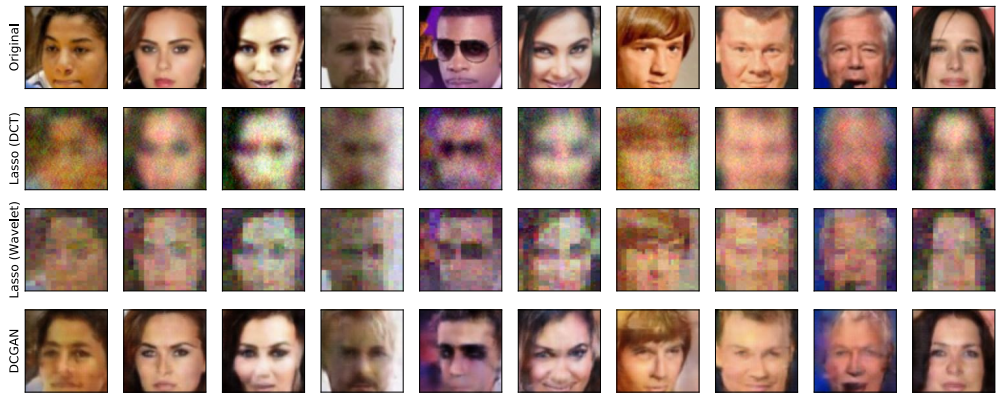


(c) 200 measurements

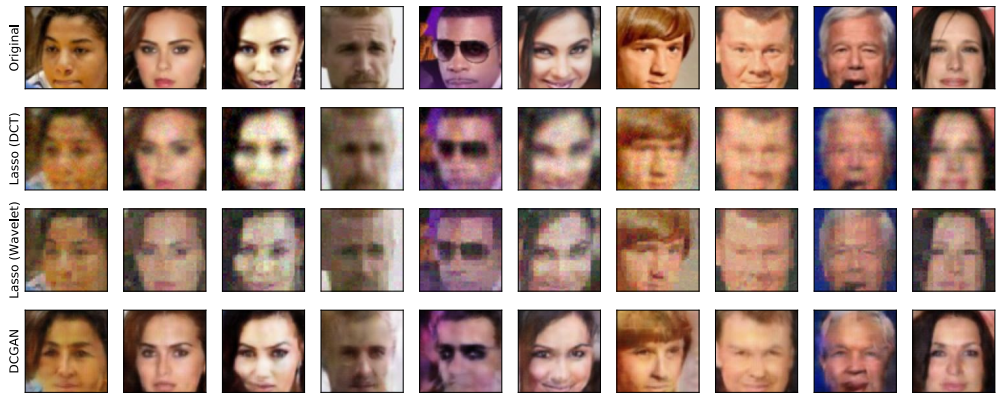
Figure 12: Reconstruction on celebA. In each image, top row is ground truth, subsequent two rows show reconstructions by Lasso (DCT) and Lasso (Wavelet) respectively. The bottom row is the reconstruction by our algorithm.



(a) 500 measurements

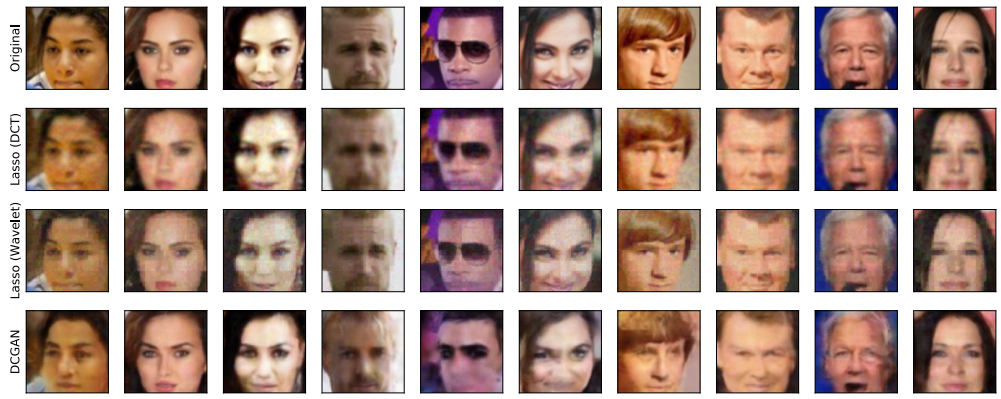


(b) 1000 measurements

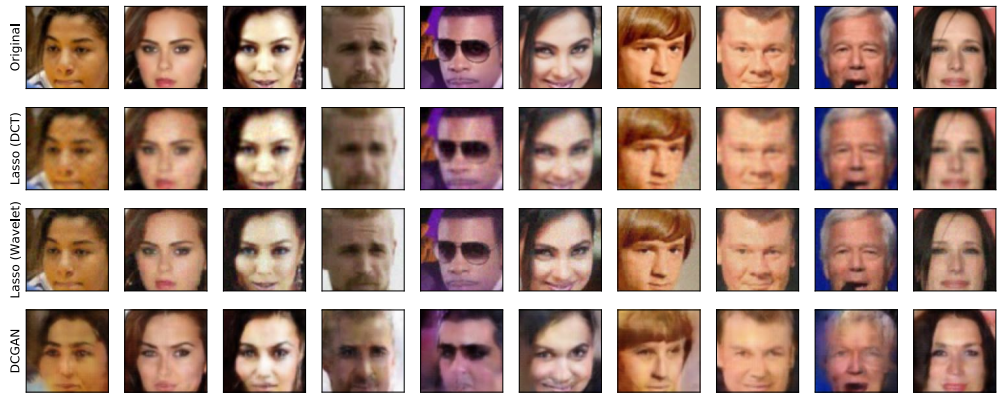


(c) 2500 measurements

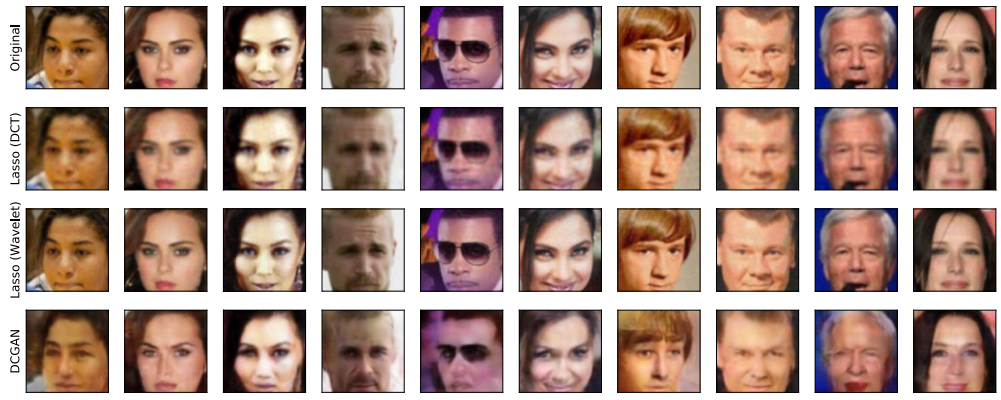
Figure 13: Reconstruction on celebA. In each image, top row is ground truth, subsequent two rows show reconstructions by Lasso (DCT) and Lasso (Wavelet) respectively. The bottom row is the reconstruction by our algorithm.



(a) 5000 measurements



(b) 7500 measurements



(c) 10000 measurements

Figure 14: Reconstruction on celebA. In each image, top row is ground truth, subsequent two rows show reconstructions by Lasso (DCT) and Lasso (Wavelet) respectively. The bottom row is the reconstruction by our algorithm.