FAST COMPRESSIVE SENSING RECOVERY USING GENERATIVE MODELS WITH STRUCTURED LATENT VARIABLES

Shaojie Xu *Sihan Zeng* *Justin Romberg*

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332
{kyle.xu, szeng30}@gatech.edu, {jrom}@ece.gatech.edu

ABSTRACT

Deep learning models have significantly improved the visual quality and accuracy on compressive sensing recovery. In this paper, we propose an algorithm for signal reconstruction from compressed measurements with image priors captured by a generative model. We search and constrain on latent variable space to make the method stable when the number of compressed measurements is extremely limited. We show that, by exploiting certain structures of the latent variables, the proposed method produces improved reconstruction accuracy and preserves realistic and non-smooth features in the image. Our algorithm achieves high computation speed by projecting between the original signal space and the latent variable space in an alternating fashion.

Index Terms— compressive sensing, imaging inverse problems, GAN, generative models, deep learning

1. INTRODUCTION

In compressive sensing (CS), we seek to reconstruct a high-dimensional signal after observing a small number of linearly coded measurements. Mathematically, given a vector $\boldsymbol{x} \in \mathbb{R}^N$, we obtain its linearly compressed representation $\boldsymbol{y}$ in a low dimensional space $\mathbb{R}^M$ $(M \ll N)$ by applying $\boldsymbol{y} = \boldsymbol{\Phi x}$, where $\boldsymbol{\Phi}$ is the compression matrix in $\mathbb{R}^{M \times N}$. In the case of useful compression, for two distinct signals $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in the original space, their corresponding representations $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in the compressed domain also need to be separate. As $\boldsymbol{\Phi}$ is an under-determined matrix with a null space, information can not be preserved for all vectors in $\mathbb{R}^N$. Fortunately, real-life signals often have prominent structures and only lie in some subspace of $\mathbb{R}^N$.

Traditional CS focuses on sparse signals. When a signal $\boldsymbol{x}$ has (or approximately has) a k-sparse representation in some basis, the compression matrix $\boldsymbol{\Phi}$ can be constructed to preserve the distances between two signals in the compressed domain, a property formally known as the *restricted isometry property* [1]. Common methods to construct $\boldsymbol{\Phi}$ include choosing each entry from an i.i.d. Gaussian [2], a fair Bernoulli, or an independent Sub-Gaussian distribution [3]. Similar results were also established for signals on a smooth manifold. [4]

In order to retrieve the original signal from the compressed measurements, some type of prior knowledge needs to be assumed. When the class of signal is well-studied, the prior knowledge comes from years of experience. For example, the sparsity model is often used to recover compressively sensed natural images. Now with the power of machine learning (ML), we can extract prior knowledge efficiently from collected datasets. The powerful function approximation capability of deep neural networks also allow us to discover and represent more complicated signal structures.

In this paper, we propose a fast compressive sensing recovery algorithm using generative models with structured latent variables. The prior information of the signals is captured by a generative adversarial network (GAN). The stability of the recovery algorithm is improved when the GAN's latent variable space is well structured. Based on Alternating Direction Methods of Multipliers (ADMM), our algorithm achieves high reconstruction speed by alternatively projecting between the original signal space and the latent variable space, without involving gradient descent (GD).

To the authors' knowledge, our work is the first to extend the ADMM-based CS recovery methods to GANs. Such an extension allows us to exploit the strong priors captured by GANs and significantly increases the recoverable compression ratio. Previous works on CS recovery using generative model either had limited model capacity [5] or was slow to carry out [6, 7, 8] due to relying heavily on GD. We demonstrate a structured latent variable space in GANs plays an important role in fast and stable recovery. Our proposed algorithm achieves comparable performance with a notable speed-up compared to the gradient-based methods. Although We present our results by solving compressive sensing recovery problems, our model can be easily generalized to solve other inverse imaging problems.

2. RELATED WORK

Signal recovery from the compressed measurements can be formulated as an optimization problem of the following form:

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) + \lambda J(\boldsymbol{x}) \tag{1}$$

The first term $F(\boldsymbol{x})$ is the fidelity term which controls how well a candidate signal $\boldsymbol{x}$ matches the measurement $\boldsymbol{y}$ in the compressed domain. A common choice of $F(\cdot)$ is to reflect the Euclidean distance in the compressed domain, such as $||\boldsymbol{y}-\boldsymbol{\Phi x}||_2^2$. The second term $J(\cdot)$ is the property term which encodes the properties the signal of interest must satisfy. In the case of sparse signals, $J(\cdot)$ can be $||\boldsymbol{x}||_0$ or the convex relaxation form $||\boldsymbol{x}||_1$. When a dataset is available, one can "learn" the properties of the signals using ML algorithms. Unsupervised learning methods such as Gaussian mixture model (GMM) [5] and variational autoencoder (VAE) [6] have been proposed. A scalar λ controls the trade-off between the fidelity term and the property term.

Recovering sparse signals is a well-studied area. Algorithms such as orthogonal matching pursuit [9], linear programming [1], least angle regression stagewise [10], soft thresholding [11], and approximate message passing [12] were proposed to solve the variations of the optimization problem (1).

When $J(\cdot)$ is learned directly from the data, it usually has a non-convex form. Although the global minimum is difficult to find, a local minimum may already result in satisfying results. One can also apply ADMM to (1), which leads to iterative solving steps:

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \arg\min_{\boldsymbol{x}}\ F(\boldsymbol{x}) + \frac{\rho}{2}||\boldsymbol{x}-\boldsymbol{s}^{(k)}+\boldsymbol{\mu}^{(k)}||_2^2 \\
\boldsymbol{s}^{(k+1)} &= \arg\min_{\boldsymbol{s}}\ \lambda J(\boldsymbol{s}) + \frac{\rho}{2}||\boldsymbol{x}^{(k+1)}-\boldsymbol{s}+\boldsymbol{\mu}^{(k)}||_2^2 \\
\boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \boldsymbol{x}^{(k+1)} - \boldsymbol{s}^{(k+1)}
\end{aligned} \tag{2}$$

ADMM introduces one auxiliary variable $\boldsymbol{s}$ and one dual variable $\boldsymbol{\mu}$. Let $F(\cdot)$ take the form of $||\boldsymbol{y}-\boldsymbol{\Phi x}||_2^2$, and the first update step in (2) can be easily computed by solving least squares. The second update step is a proximal operator of J, and can be understood as finding a signal $\boldsymbol{s}$ that is close to the target signal $\boldsymbol{x}^{(k+1)}+\boldsymbol{\mu}^{(k)}$ while satisfying the properties encoded in $J(\cdot)$. Such a formulation is often considered as a "denoising" step and inspires many recent works on "plug-and-play" methods. In these works, the second update step is replaced by state-of-the-art denoisers. Instead of explicitly learning $J(\cdot)$ to capture the statistics of the signals, denoisers are trained directly to mimic the behavior of the second update step with the properties of the signals embedded into the denoisers' design. Common denoiser choices include block-matching and 3D filtering (BM3D) [13, 14], and feed-forward neural networks [15, 16]. Adversarial training was proposed in [17, 18] to improve denoising and recovery. Discriminator were used to evaluate the denosing effect, however no generative model was trained to directly captured the statistics of the singal datasets.

In the aforementioned ADMM setup, $\boldsymbol{x}$ and $\boldsymbol{s}$ are both in the original space $\mathbb{R}^N$ (with the ADMM underlying constraint $\boldsymbol{x}=\boldsymbol{s}$). When generative models are used to capture the signal statistics, latent variables $\boldsymbol{z}\in\mathbb{R}^L$ are usually introduced. [5] used a GMM to model a smooth manifold and searched $\boldsymbol{z}^*$ in the latent space which has a closed form solution. [6] used a VAE to learn a non-linear mapping $G_{gen}(\cdot)$ from $\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma}^2)$ to $\boldsymbol{x}$ and applied GD to the optimization problem $\min_z ||\boldsymbol{y}-\boldsymbol{\Phi}G_{gen}(\boldsymbol{z})||_2^2+\lambda||\boldsymbol{z}||_2^2$. Recent developments in GAN [19, 20] shed new lights on learning signal statistics. [7] used a GAN to capture the image prior and applied GD to the same optimization problem as in [6] for recovery. [8] proposed an algorithm that alternates between one step GD on the fidelity term and searching the latent variable space with the latter still achieved by GD. These gradient based methods suffer from high computational complexity and slow recovery speed. We seek to combine the fast computation from ADMM-based methods and the strong prior-capture ability from the generative models to achieve fast CS recovery with ultra small number of measurements.

3. ALGORITHMS

In the proposed algorithms, we formulate the CS recovery problem as searching $\boldsymbol{x}$ and $\boldsymbol{z}$ in the original signal space and in the latent variable space, respectively:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{z}} \quad & ||\boldsymbol{y}-\boldsymbol{\Phi x}||_2^2+\lambda H(\boldsymbol{z}) \\
s.t. \quad & \boldsymbol{x}=G_{gen}(\boldsymbol{z})
\end{aligned} \tag{3}$$

We denote $G_{gen}(\cdot)$ as the generative model and $H(\cdot)$ as the function that captures the property that the latent variable $\boldsymbol{z}$ should satisfy. Even though the formulation above contains a non-linear equality constraint, we may still solve it by searching a stationary point of its augmented Lagrangian, which leads the following ADMM-like update steps:

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\rho\mathbf{I}\right)^{-1}\left(\boldsymbol{\Phi}^T\boldsymbol{y}+\rho(G_{gen}(\boldsymbol{z}^{(k)})-\boldsymbol{\mu}^{(k)})\right) \\
\boldsymbol{z}^{(k+1)} &= \arg\min_{\boldsymbol{z}}\ \lambda H(\boldsymbol{z})+\frac{\rho}{2}||\boldsymbol{x}^{(k+1)}-G_{gen}(\boldsymbol{z})+\boldsymbol{\mu}^{(k)}||_2^2 \\
\boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)}+\boldsymbol{x}^{(k+1)}-G_{gen}(\boldsymbol{z}^{(k+1)})
\end{aligned} \tag{4}$$

The signal property imposed by $H(\cdot)$ plays an important role in searching $\boldsymbol{z}$ in the latent variable space. In classic GAN models such as the deep convolutional GAN (DCGAN) proposed in [20], $\boldsymbol{z}$ is assumed to be drawn from a standard multivariate Gaussian distribution. Setting $H(\boldsymbol{z})=||\boldsymbol{z}||_2$ is a common practice to enforce such latent space structure. However, this latent variable space is still lack of interpretability and controllability, and how each latent dimension contributes to the generated signal is unclear [21, 22, 23]. The CS recovery performance bound provided in [6] shows the number of required compressed measurements grows linearly with the latent variable dimension.

To enforce better structured latent variable space, we propose training the generative models with an InfoGAN setup [21], where the each latent variable $\boldsymbol{z}$ is split into codewords $\boldsymbol{c}$ and "random-noise-like" variable $\boldsymbol{\gamma}$. An InfoGAN is trained to not only minimize the usual GAN's loss function, but also maximize the mutual information between the codewords $\boldsymbol{c}$ and the generated signals $G_{gen}(\boldsymbol{z})$. As a results, the codewords $\boldsymbol{c}$ were able to control most of the semantic meaning in the generated signals, while $\boldsymbol{\gamma}$ only adds small variations to the results. This well-structured latent variable space helps CS recovery: When the number of compressed measurements is sufficient, both $\boldsymbol{c}$ and $\boldsymbol{\gamma}$ may be inferred from the measurements, leading to more accurate reconstructions. When the number of compressed measurements is extremely limited, we can recover an approximate signal (within the small variation controlled by γ) as long as $\boldsymbol{c}$ can be inferred.

The x update step in equation (4) solves a least squares problem, while solving for the $\boldsymbol{z}$ update is computationally expensive with GD. Similar to the "plug-and-play" ADMM method, we propose to use a projector neural network G_{proj} to learn the solution to such an optimization problem. Notice that during each iteration, $\boldsymbol{x}^{(k+1)}$ contains the noise introduced by the previous least square update. We, therefore, propose to train G_{proj} using randomly sampled latent variables and the noisy version of their generated samples.

Alternatively, we can cascade the projector network and the generator network to form a network $G_{gen}(G_{proj}(\cdot))$ similar to an "autoencoder". We then draw samples directly from the dataset, and train the "autoencoder" to recover these samples from their noisy observations. G_{gen} is fixed during this training process. When this method is used, our proposed algorithm is similar to a "plug-and-play" ADMM model with a generative-model-based denoiser. Such a similarity may provide a better understanding about the convergence behavior of the proposed algorithm, while our previous derivation provides a better understanding about the importance of using well-structured latent variables. The complete version of the proposed algorithm is summarized in Algorithm 1.

Algorithm 1 Fast CS recovery using generative models (F-CSRG)

Train a generative model $G_{gen}(\cdot)$ on the dataset
Method I:
  Generate random latent variable $\boldsymbol{z}$s following its distribution.
  Generate random noise $\boldsymbol{\epsilon}$ according to some distribution.
  Construct noisy signals $\tilde{\boldsymbol{x}}$s such that $\tilde{\boldsymbol{x}} = G_{gen}(\boldsymbol{z}) + \boldsymbol{\epsilon}$
  Train a projector network $G_{proj}(\cdot)$ that maps $\tilde{\boldsymbol{x}}$ to $\boldsymbol{z}$
Method II:
  Draw samples $\boldsymbol{x}$ from the training set
  Generate random noise $\boldsymbol{\epsilon}$ according to some distribution.
  Construct noisy signals $\tilde{\boldsymbol{x}}$s such that $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$
  Train a projector network $G_{proj}(\cdot)$ such that $G_{gen}(G_{proj}(\cdot))$ maps $\tilde{\boldsymbol{x}}$ to $\boldsymbol{x}$. (G_{gen} is fixed)

For signal recovery:
Given compression matrix $\boldsymbol{\Phi}$, compressed measurements $\boldsymbol{y}$
while Stopping criteria not met **do**
  $\boldsymbol{x}^{(k+1)} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \rho\mathbf{I}\right)^{-1}\left(\boldsymbol{\Phi}^T\boldsymbol{y} + \rho(G_{gen}(\boldsymbol{z}^{(k)}) - \boldsymbol{\mu}^{(k)})\right)$
  $\boldsymbol{z}^{(k+1)} = G_{proj}(\boldsymbol{x}^{(k+1)})$
  $\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \boldsymbol{x}^{(k+1)} - G_{gen}(\boldsymbol{z}^{(k+1)})$
end while

4. TESTING AND RESULTS

4.1. MNIST Dataset

We tested the proposed algorithm using the MNIST digits dataset [24]. Selected recovered results are shown in Figure 1. We used an i.i.d. Gaussian random matrix $\boldsymbol{\Phi}$ for the compression. For comparison purposes, we also included the results of three baseline algorithms. The first algorithm is the "plug-and-play" ADMM with a denoising autoencoder (DAE). As this denoiser was trained directly in the pixel domain (mapping $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$ back to $\boldsymbol{x}$), it failed to capture a strong prior knowledge about the digits, and started to produce images with large artifacts as the compression rate went to 32x. The second baseline algorithm used the well-known total variance (TV) [25] as the regularizer. The third baseline algorithm used DCGAN [20] to capture the signal statistics, while the CS recovery was performed using GD as proposed in [6]. We used the same DCGAN's generator as G_{gen} in our F-CSRG algorithm and trained a neural network with fully connected layers (784-1024-512-256-100, ReLU activation) as G_{proj}. F-CSRG performed comparably with the gradient-descent-based algorithm, while only took 1/20 of the computational time. When the number of compressed measurements were significantly reduced, both DCGAN-based models broke down due to unstable projection caused by their less structured latent variable space. The best performance was achieved by incorporating InfoGAN (trained as suggested in [21]) into our algorithm, demonstrating the benefit of having well-structured latent variables.

We tested three fast recovery algorithms on MNIST digits' testing dataset and measured the Euclidean distance between the true images and the recovered images. The results are shown in Table 1. As the compression ratio increases, our algorithm with InfoGAN outperformed the other algorithms. We also used the classification accuracy as another metric to assess the recovery quality. We trained a convolutional neural network as the classifier which achieved 99.3% accuracy on the original MNIST testing dataset. We then applied this MNIST classifier to the images reconstructed by the three algorithms. As shown in Table 1, when the number of compressed measurements is extremely limited, our proposed InfoGAN algorithm achieved significantly higher accuracy than the other two methods, as a result of its structured latent variable space.

4.2. Celeb A Datasets

We also tested the proposed algorithm using the CelebA dataset [26]. Each image is of dimension 32×32, cropped

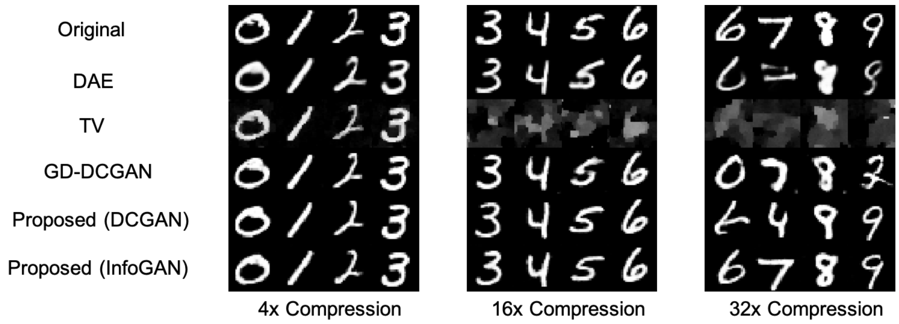

Fig. 1. Comparison of selected MNIST digits recovered by different algorithms.

Fig. 2. Comparison of selected CelebA images recovered by different algorithms

Compression Ratio	DAE	F-CSRG with DCGAN	F-CSRG with InfoGAN
4x	2.20 / 98.2%	2.25 / 98.3%	2.68 / 97.7%
8x	2.54 / 97.8%	2.72 / 97.3%	3.06 / 97.2%
16x	3.23 / 94.8%	3.70 / 91.7%	3.79 / 93.8%
32x	5.13 / 73.5%	5.86 / 66.4%	5.37 / 77.4%
64x	7.33 / 41.8%	7.91 / 36.2%	7.43 / 48.0%

Table 1. Average reconstruction error (measurened as the Euclidean distance) and classification accuracy of the reconstructed digits on MNIST digits' testing dataset

and downsampled from the original dataset. Generative models were trained based on standard DCGAN and InfoGAN architectures as proposed in the original papers. For InfoGAN, we used 5 categorical codes (one-hot encoding with 10 classes), 5 continuous codes, and noise of length 128, producing a latent variable space of length 183. We trained a fully connected network (1024-512-256-183, ReLu activation) as G_{proj}. Bases on whether the desired codeword is categorical or continuous, we used softmax as the activation function or simply skipped activation on the output layer. We tested the proposed algorithm with 4x, 8x, and 16x compression. For comparison purposes, we also included the results of two baseline algorithms. Similar to the testing on MNIST dataset, the first baseline algorithm used the TV regularization, and the second one used the "plug-and-play" ADMM with a DAE trained directly in the pixel domain. Selected recovered results are shown in Figure 2.

As the compression rate increases, the quality of TV recovered images quickly degrades and few features on the face can be recovered. DAE produces comparable or sometimes eve betternreconstruction in the case of 4x and 8x compression, but becomes unst ble undareextremely high compression rat.e In contrast, the proposed F-CSRG method is very stable against the drop in the number of compressed measurements. In addition, because generative model provides a strong image prior that assumes face images to have sharp features and not necessarily smooth everywhere, images reconstructed by the proposed algorithm preserve high-frequency contents of the original images. Out testing results also show that F-CSRG works better with an InfoGAN implementation than with a DCGAN. As we have discussed in previous sections, this improvement in recovered image quality likely comes from the more structured latent varialbe space produced by the InfoGAN's codewords [1].

5. CONCLUSION

This paper proposed an algorithm of using generative models to solve compressive sensing inverse problems. This method is fast to carry out, by exploiting a projector network, and is stable under high compression factor, by putting constraints on the latent variable space. It consistently produces high-quality images even when the observations are highly compressed. The proposed algorithm can be easily generalized to solve inverse imaging problems besides CS recovery.

[1] A Tensorflow implementation can be found at https://github.com/sihanzeng/f-csrg.

6. REFERENCES

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

[2] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[3] Emmanuel J Candès and Michael B Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[4] Richard G Baraniuk and Michael B Wakin, "Random projections of smooth manifolds," *Foundations of computational mathematics*, vol. 9, no. 1, pp. 51–77, 2009.

[5] Minhua Chen, Jorge Silva, John Paisley, Chunping Wang, David Dunson, and Lawrence Carin, "Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6140–6155, 2010.

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

[7] Maya Kabkab, Pouya Samangouei, and Rama Chellappa, "Task-aware compressed sensing with generative adversarial networks," *arXiv preprint arXiv:1802.01284*, 2018.

[8] Viraj Shah and Chinmay Hegde, "Solving linear inverse problems using gan priors: An algorithm with provable guarantees," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4609–4613.

[9] Joel A Tropp and Anna C Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[10] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[11] Neal Parikh, Stephen Boyd, et al., "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[12] David L Donoho, Arian Maleki, and Andrea Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," in *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[14] Christopher A Metzler, Arian Maleki, and Richard G Baraniuk, "Bm3d-amp: A new image recovery algorithm based on bm3d denoising," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3116–3120.

[15] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012, pp. 341–349.

[16] Harold C Burger, Christian J Schuler, and Stefan Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2392–2399.

[17] JH Rick Chang, Chun-Liang Li, Barnabas Poczos, BVK Vijaya Kumar, and Aswin C Sankaranarayanan, "One network to solve them all–solving linear inverse problems using deep projection models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5888–5897.

[18] Kai Fan, Qi Wei, Lawrence Carin, and Katherine A Heller, "An inner-loop free solution to inverse problems using deep neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 2370–2380.

[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*, 2014, pp. 2672–2680.

[20] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2016.

[21] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in neural information processing systems*, 2016, pp. 2172–2180.

[22] LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang, "Triple generative adversarial nets," in *Advances in neural information processing systems*, 2017, pp. 4088–4098.

[23] Zhijie Deng, Hao Zhang, Xiaodan Liang, Luona Yang, Shizhen Xu, Jun Zhu, and Eric P Xing, "Structured generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 3899–3909.

[24] http://yann.lecun.com/exdb/mnist/, "Mnist database of handwritten digits," .

[25] Emmanuel J Candès, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 12 2015.