Deep Compressed Sensing

Yan Wu [1]   Mihaela Rosca [1]   Timothy Lillicrap [1]

Abstract

Compressed sensing (CS) provides an elegant framework for recovering sparse signals from compressed measurements. For example, CS can exploit the structure of natural images and recover an image from only a few random measurements. CS is flexible and data efficient, but its application has been restricted by the strong assumption of sparsity and costly reconstruction process. A recent approach that combines CS with neural network generators has removed the constraint of sparsity, but reconstruction remains slow. Here we propose a novel framework that significantly improves both the performance and speed of signal recovery by jointly training a generator and the optimisation process for reconstruction via meta-learning. We explore training the measurements with different objectives, and derive a family of models based on minimising measurement errors. We show that Generative Adversarial Nets (GANs) can be viewed as a special case in this family of models. Borrowing insights from the CS perspective, we develop a novel way of improving GANs using gradient information from the discriminator.

1. Introduction

Encoding and decoding are central problems in communication (MacKay, 2003). Compressed sensing (CS) provides a framework that separates encoding and decoding into independent measurement and reconstruction processes (Candes et al., 2006; Donoho, 2006). Unlike commonly used auto-encoding models (Bourlard & Kamp, 1988; Kingma & Welling, 2013; Rezende et al., 2014), which feature end-to-end trained encoder and decoder pairs, CS reconstructs signals from low-dimensional measurements via online optimisation. This model architecture is highly flexible and sample efficient: high dimensional signals can be reconstructed from a few random measurements with little or no

---

[1]DeepMind, London, UK. Correspondence to: Yan Wu <yanwu@google.com>.

training at all. CS has been successfully applied in scenarios where measurements are noisy and expensive to take, such as in MRI (Lustig et al., 2007). Its sample efficiency enables the development of, for example, the "single pixel camera", which reconstructs a full resolution image from a single light sensor (Duarte et al., 2008).

However, the wide application of CS, especially in processing large scale data where modern deep learning approaches thrive, is hindered by its assumption of sparse signals and the slow optimisation process for reconstruction. Recently, Bora et al. (2017) combined CS with separately trained neural network generators. Although these pre-trained neural networks were not optimized for CS, they demonstrated reconstruction performance superior to existing methods such as the Lasso (Tibshirani, 1996). Here we propose the deep compressed sensing (DCS) framework in which neural networks can be trained from-scratch for both measuring and online reconstruction. We show that this framework leads naturally to a family of models, including GANs (Goodfellow et al., 2014), which can be derived by training the measurement functions with different objectives. In summary, this work contributes the following:

- We demonstrate how to train deep neural networks within the CS framework.

- We show that a meta-learned reconstruction process leads to a more accurate and orders of magnitudes faster method compared with previous models.

- We develop a new GAN training algorithm based on latent optimisation, which improves GAN performance. The non-saturated generator loss $-\ln(D(G(z)))$ emerges as a measurement error.

- We extend our framework to training semi-supervised GANs, and show that latent optimisation results in semantically meaningful latent spaces.

Notations

We use bold letters for vectors and matrices and normal letters for scalars. $\mathbb{E}_{p(x)}[f(x)]$ indicates taking the expectation of $f(x)$ over the distribution $p(x)$. We use subscriptions of Greek letters to indicate function parameters. For example, G_θ is a function parametrised by θ.
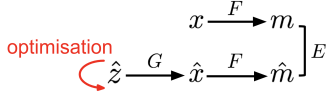
Figure 1. Illustration of Deep Compressed Sensing. $\mathbf{F}$ is a measurement process that produces a measurement of the signal, $\boldsymbol{m}$ and $\mathbf{G}$ is a generator that reconstructs the signal from a latent representation $\hat{z}$. The latent representation is optimised to minimise a measurement error $E_\theta(\boldsymbol{m}, \hat{\boldsymbol{m}})$.

2. Background

2.1. Compressed Sensing

Compressed sensing aims to recover signal $\boldsymbol{x}$ from a linear measurement $\boldsymbol{m}$:

$$\boldsymbol{m} = \mathbf{F}\,\boldsymbol{x} + \eta \tag{1}$$

where $\mathbf{F}$ is the $C \times D$ *measurement matrix*, and η is the measurement noise which is usually assumed to be Gaussian distributed. $\mathbf{F}$ is typically a "wide" matrix, such that $C \ll D$. As a result, the measurement $\boldsymbol{m}$ has much lower dimensionality compared with the original signal; solving $\boldsymbol{x}$ is generally impossible for such under-determined problems. The elegant CS theory shows that one can nearly perfectly recover $\boldsymbol{x}$ with high probability given a random matrix $\mathbf{F}$ and sparse $\boldsymbol{x}$ (Donoho, 2006; Candes et al., 2006). In practice, the requirement that $\boldsymbol{x}$ be sparse can be replaced by sparsity in a set of basis Φ, such as the Fourier basis or wavelet, so that $\Phi\,\boldsymbol{x}$ can be non-sparse signals such as natural images. Here we omit the basis Φ for brevity; the linear transform from Φ does not affect our following discussion.

At the centre of CS theory is the Restricted Isometry Property (RIP) [1], which is defined for $\mathbf{F}$ and the difference between signals $\boldsymbol{x}_1 - \boldsymbol{x}_2$ as

$$(1 - \delta)\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 \le \|\mathbf{F}\,(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|_2^2 \\ \le (1 + \delta)\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 \tag{2}$$

where $\delta \in (0, 1)$ is a small constant. The RIP states that the projection from $\mathbf{F}$ preserves the distance between two signals bounded by factors of $1 - \delta$ and $1 + \delta$. This property holds with high probability for various random matrices F and sparse signals x. It guarantees minimising the measurement error

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{m} - \mathbf{F}\,\boldsymbol{x}\|_2^2 \tag{3}$$

under the constraint that $\boldsymbol{x}$ is sparse, leads to accurate reconstruction $\hat{\boldsymbol{x}} \approx \boldsymbol{x}$ with high probability (Donoho, 2006;

---

[1] The theory can also be proved from the closely related and more general Restricted Eigenvalue condition (Bora et al., 2017). We focus on RIP in this form for its more straightforward connection with the training loss (see section 3.1).

Candes et al., 2006). This constrained optimisation problem is computationally intensive — a price for the measuring process that only requires sparse random projections of the signals (Baraniuk, 2007).

2.2. Compressed Sensing using Generative Models

The requirement of sparsity poses a strong restriction on CS. Sparse bases, such as the Fourier basis or wavelet, only partially relieve this constraints, since they are restricted to domains known to be sparse in these bases and cannot adapt to data distributions. Recently, Bora et al. (2017) proposed compressed sensing using generative models (CSGM) to relax this requirement. This model uses a *pre-trained* deep neural network G_θ (from a VAE or GAN) as the structural constraint in the place of sparsity. This generator maps a latent representation $\boldsymbol{z}$ to the signal space:

$$\boldsymbol{x} = G_\theta(\boldsymbol{z}) \tag{4}$$

Instead of requiring sparse signals, G_θ implicitly constrains output $\boldsymbol{x}$ in a low-dimensional manifold via its architecture and the weights adapted from data. This constraint is sufficient to provide a generalised Set-Restricted Eigenvalue Condition (S-REC) with random matrices, under which low reconstruction error can be achieved with high probability. A minimisation process similar to that in CS is used for reconstruction:

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} E_\theta(\boldsymbol{m}, \boldsymbol{z}) \tag{5}$$

$$E_\theta = \|\boldsymbol{m} - \mathbf{F}\,G_\theta(\boldsymbol{z})\|_2^2 \tag{6}$$

such that $\hat{\boldsymbol{x}} = G_\theta(\boldsymbol{z})$ is the reconstructed signal. In contrast to directly optimising the signal $\boldsymbol{x}$ in CS (eq.3), here optimisation is in the space of latent representation $\boldsymbol{z}$.

The $\arg\min$ operator in eq. 5 is intractable since E_θ is highly non-convex. It is therefore approximated using gradient descent starting from a randomly sampled point $\hat{\boldsymbol{z}} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$:

$$\hat{\boldsymbol{z}} \leftarrow \hat{\boldsymbol{z}} - \alpha\,\left.\frac{\partial E_\theta(\boldsymbol{m}, \boldsymbol{z})}{\partial \boldsymbol{z}}\right|_{\boldsymbol{z}=\hat{\boldsymbol{z}}} \tag{7}$$

where α is a learning rate. One can take a specified T steps of gradient descent. Typically, hundreds or thousands of gradient descent steps and several re-starts from the initial step are needed to obtain a sufficiently good $\hat{\boldsymbol{z}}$ (Bora et al., 2017; Bojanowski et al., 2018). This process is illustrated in Figure 1.

This work established the connection between compressed sensing and deep neural networks, and demonstrated performance superior to the Lasso (Tibshirani, 1996), especially when the number of measurements is small. The theoretical properties of CSGM have been more closely examined by Hand & Voroninski (2017), who also proved stronger

convergence guarantees. More recently, Dhar et al. (2018) proposed additional constraints to allow *sparse deviation* from the generative model's support set, thus improving generalisation. However, CSGM still suffers from two restrictions:

1. The optimisation for reconstruction is still slow, as it requires thousands of gradient descent steps.

2. It relies on random measurement matrices, which are known to be sub-optimal for highly structured signals such natural images. Learned measurements can perform significantly better (Weiss et al., 2007).

2.3. Model-Agnostic Meta Learning

Meta-learning, or learning to learn, allows a model adapting to new tasks by self-improving (Schmidhuber, 1987). Model-Agnostic Meta learning (MAML) provides a general method to adapt parameters for a number of tasks (Finn et al., 2017). Given a differentiable loss function $\mathcal{L}(\mathcal{T}_i; \theta)$ for task $\mathcal{T}_i$ sampled from the task distribution $p_{\text{task}}(\mathcal{T})$, the task-specific parameters are adapted by gradient descent from the initial parameters θ:

$$\theta_i \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\mathcal{T}_i; \theta) \qquad (8)$$

The initial parameters θ are trained to minimise the loss across all tasks

$$\min_\theta \mathbb{E}_{\mathcal{T}_i \sim p_{\text{task}}(\mathcal{T})} \left[\mathcal{L}(\mathcal{T}_i; \theta_i) \right] \qquad (9)$$

Multiple steps and more sophisticated optimisation algorithms can be used in the place of eq. 8. Despite $\mathcal{L}$ usually being a highly non-convex function, by back-propagating through the gradient-descent process, only a few gradient steps are sufficient to adapt to new tasks.

2.4. Generative Adversarial Networks

A Generative Adversarial Network (GAN) trains a parametrised generator G_θ to fool a discriminator D_ϕ that tries to distinguish real data from fake data sampled from the generator (Goodfellow et al., 2014). The generator G_θ is a deterministic function that transforms samples $\boldsymbol{z}$ from a source $p_{\boldsymbol{z}}(\boldsymbol{z})$ to the same space as the data $\boldsymbol{x}$, which has the distribution $p_{\text{data}}(\boldsymbol{x})$. This adversarial game can be summarised by the following min-max problem with the value function $V(G_\theta, D_\phi)$:

$$\min_{G_\theta} \max_{D_\phi} V(G_\theta, D_\phi) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} \left[\ln D_\phi(\boldsymbol{x}) \right]$$
$$+ \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} \left[\ln(1 - D_\phi(G_\theta(\boldsymbol{z}))) \right] \qquad (10)$$

GANs are usually difficult to train due to this adversarial game (Balduzzi et al., 2018). Training may either diverge or converge to bad equilibrium with, for example, collapsed modes, unless extra care is taken in designing and training the model (Radford et al., 2015; Salimans et al., 2016).

A widely adapted trick is using $-\ln(D(G(\boldsymbol{z})))$ as the objective for the generator (Goodfellow et al., 2014). Compared with eq. 10, this alternative objective avoids saturating the discriminator in the early stage of training when the generator is too weak. However, this objective voids most theoretical analyses (Hu et al., 2018), since the new adversarial objective is no longer a zero-sum game (eq. 10).

In most GAN models, discriminators become useless after training. Recently, Tao et al. (2018) and Azadi et al. (2019) proposed methods using the discriminator for importance sampling. Our work provides an alternative: our model moves latent representations to areas more likely to generate realistic images as deemed by the discriminator.

3. Deep Compressed Sensing

We start by showing the benefit of combining meta-learning with the model in Bora et al. (2017). We then generalise measurement matrices to parametrised measurement functions, including deep neural networks. While previous work relies on random projections as measurement functions, our approach learns measurement functions by imposing the RIP as a training objective. We then derive two novel models by imposing properties other than the RIP on the measurements, including a GAN model with discriminator-guided latent optimisation, which leads to more stable training dynamics and better results.

3.1. Compressed Sensing with Meta-Learning

We hypothesise that the run-time efficiency and performance in CSGM (Bora et al. 2017, section 2.2), can be improved by training the latent optimisation procedure using meta-learning, by back-propagating through the gradient descent steps (Finn et al., 2017). The latent optimisation procedure for CS models can take hundreds or thousands of steps. By employing meta-learning to optimise this optimisation procedure we aim to achieve similar results with far fewer updates.

To this end, the model parameters, as well as the latent optimisation procedure, are trained to minimise the expected measurement error:

$$\min_\theta \mathcal{L}_G, \text{ for } \mathcal{L}_G = \mathbb{E}_{\boldsymbol{x}_i \sim p_{\text{data}}(\boldsymbol{x})} \left[E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i) \right] \qquad (11)$$

where $\hat{\boldsymbol{z}}_i$ is obtained from gradient descent (eq. 7). The gradient descent in eq. 7 and the loss function in eq. 11 mirror their counterparts in MAML (eq. 8 and 9), except that:

1. Instead of the stochastic gradient computed in the outside loop, here each measurement error E_θ only de-

pends on a single sample z, so eq. 7 computes the exact gradient of E_θ.

2. The online optimisation is over latent variables rather than parameters. There are usually much fewer latent variables than parameters, so the update is quicker.

Like in MAML, we implicitly perform second order optimisation, by back-propagating through the latent optimisation steps which compute $\hat{z}_i$ when optimising eq. 11. We empirically observed that this dramatically improves the efficiency of latent optimisation, with only 3-5 gradient descent steps being sufficient to improve upon baseline methods.

Unlike Bora et al. (2017), we also train the generator G_θ. Merely minimising eq. 9 would fail — the generator can exploit $\mathbf{F}$ by mapping all $G_\theta(z)$ into the null space of $\mathbf{F}$. This trivial solution always gives zero measurement error, but may contain no useful information. Our solution is to enforce the RIP (eq. 2) via training, by minimising the *measurement loss*:

$$\mathcal{L}_F = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2} \left[\left(\|\mathbf{F}\left(\boldsymbol{x}_1 - \boldsymbol{x}_2\right)\|_2 - \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \right)^2 \right] \quad (12)$$

$\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ can be sampled in various ways. While the choice is not unique, it is important to sample from both the data distribution $p_{\text{data}}(\boldsymbol{x})$ and generated samples $G_\theta(\boldsymbol{z})$, so that the trained RIP holds for both real and generated data. In our experiments, we randomly sampled one image from the data and two generated images at the beginning and end of latent optimisation, then computed the average between the 3 pairs of losses between these 3 points as a form of "triplet loss".

Our algorithm is summarised in Algorithm 1. Since Algorithm 1 still uses a random measurement matrix $\mathbf{F}$, it can be used as any other CS algorithm when ground truth reconstructions are available for training the generator.

3.2. Deep Compressed Sensing with Learned Measurement Function

In Algorithm 1, we use the RIP property to train the generator. We can use the same approach and enforce the RIP property to learn the measurement function $\mathbf{F}$ itself, rather than using a random projection.

3.2.1. LEARNING MEASUREMENT FUNCTION

We start by generalising the measurement matrix $\mathbf{F}$ (eq. 1), and define a parametrised measurement function $m \leftarrow F_\phi(\boldsymbol{x})$. The model introduced in the previous section corresponds to a linear function $F_\phi(\boldsymbol{x}) = \mathbf{F}\,\boldsymbol{x}$; now both F_ϕ and G_θ can be deep neural networks. Similar to CS, the central problem in this generalised setting is inverting the measurement function to recover the signal $\boldsymbol{x} \leftarrow F_\phi^{-1}(\boldsymbol{m})$ via minimising the measurement error similar to eq. 6:

$$E_\theta(\boldsymbol{m}, \boldsymbol{z}) = \|\boldsymbol{m} - F_\phi(G_\theta(\boldsymbol{z}))\|_2^2 \quad (13)$$

Algorithm 1 Compressed Sensing with Meta Learning

Input: minibatchs of data $\{\boldsymbol{x}_i\}_{i=1}^N$, random matrix $\mathbf{F}$, generator G_θ, learning rate α, number of latent optimisation steps T
repeat
  Initialize generator parameters θ
  for $i = 1$ **to** N **do**
    Measure the signal $\boldsymbol{m}_i \leftarrow \mathbf{F}\,\boldsymbol{x}_i$
    Sample $\hat{\boldsymbol{z}}_i \sim p_{\boldsymbol{z}}(\boldsymbol{z})$
    for $t = 1$ **to** T **do**
      Optimise $\hat{\boldsymbol{z}}_i \leftarrow \hat{\boldsymbol{z}}_i - \frac{\partial}{\partial \boldsymbol{z}} E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i)$
    end for
  end for
  $\mathcal{L}_G = \frac{1}{N} \sum_{i=1}^N E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i)$
  Compute $\mathcal{L}_F$ using eq. 12
  Update $\theta \leftarrow \theta - \frac{\partial}{\partial \theta}(\mathcal{L}_G + \mathcal{L}_F)$
until reaches the maximum training steps

The distance preserving property as a counterpart of the RIP can be enforced by minimising a loss similar to eq. 12:

$$\mathcal{L}_F = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2} \left[\left(\|F_\phi(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|_2 - \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \right)^2 \right] \tag{14}$$

Minimising $\mathcal{L}_F$ provides a relaxation of the constraint specified by the RIP (eq. 2). When $\mathcal{L}_F$ is small, the projection from $\mathbf{F}$ better preserves the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. This relaxation enables us to transform the RIP into a training objective for the measurements, which can then be integrated into training other model components. Empirically, we found this relaxation leads to high quality reconstruction.

The rest of the algorithm is identical to Algorithm 1, except that we also update the measurement function's parameters ϕ. Consequently, different schemes can be employed to coordinate updating θ and ϕ, which will be discussed more in section 3.3. This extended algorithm is summarised in Algorithm 2. We call it Deep Compressed Sensing (DCS) to emphasise that both the measurement and reconstruction can be deep neural networks. Next, we turn to generalising the measurements to properties other than the RIP.

3.2.2. GENERALISED CS 1: CS-GAN

Here we consider an extreme case: a *one-dimensional* measurement that only encodes how likely an input is a real data point or fake one sampled from the generator. One way to formulate this is to train the measurement function F_ϕ using the following loss instead of eq. 14:

$$\mathcal{L}_F = \begin{cases} \|F_\phi(\boldsymbol{x}) - 1\|_2^2 & \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \\ \|F_\phi(\hat{\boldsymbol{x}})\|_2^2 & \hat{\boldsymbol{x}} \sim G_\theta(\hat{\boldsymbol{z}}), \forall \hat{\boldsymbol{z}} \end{cases} \tag{15}$$

Algorithm 2 then becomes the Least Squares Generative Adversarial Nets (LSGAN, Mao et al., 2017) with latent

Algorithm 2 Deep Compressed Sensing

Input: minibatchs of data $\{\boldsymbol{x}_i\}_{i=1}^{N}$, measurement function F_ϕ, generator G_θ, learning rate α, number of latent optimisation steps T
repeat
    Initialize generator parameters θ
    for $i = 1$ **to** N **do**
        Measure the signal $\boldsymbol{m}_i \leftarrow F_\phi(\boldsymbol{x}_i)$
        Sample $\hat{z}_i \sim p_{\boldsymbol{z}}(\boldsymbol{z})$
        for $t = 1$ **to** T **do**
            Optimise $\hat{z}_i \leftarrow \hat{z}_i - \frac{\partial}{\partial \boldsymbol{z}} E_\theta(\boldsymbol{m}_i, \hat{z}_i)$
        end for
    end for
    $\mathcal{L}_G = \frac{1}{N} \sum_{i=1}^{N} E_\theta(\boldsymbol{m}_i, \hat{z}_i)$
    Compute $\mathcal{L}_F$ using eq. 12
    Option 1 : joint update $\theta \leftarrow \theta - \frac{\partial}{\partial \theta}(\mathcal{L}_G + \mathcal{L}_F)$
    Option 2 : alternating update
            $\theta \leftarrow \theta - \frac{\partial}{\partial \theta}\mathcal{L}_G \qquad \phi \leftarrow \phi - \frac{\partial}{\partial \phi}\mathcal{L}_F$
    until reaches the maximum training steps

optimisation — they are exactly equivalent when the latent optimisation is disabled ($T = 0$, zero step). LSGAN is an alternative to the original GAN (Goodfellow et al., 2014) that can be motivated from Pearson χ^2 Divergence. To demonstrate a closer connection with original GANs (Goodfellow et al., 2014), we instead focus on another formulation whose measurement function is a binary classifier (the discriminator).

This is realised by using a binary classifier D_ϕ as the measurement function, where we can interpret $D_\phi(\boldsymbol{x})$ as the probability that $\boldsymbol{x}$ comes from the dataset. In this case, the measurement function is equivalent to the *discriminator* in GANs. Consequently, we change the the squared-loss in eq. 13 to the cross-entropy loss as the matching measurement loss function (Bishop, 2006) (ignoring the expectation over $\boldsymbol{x}$ for brevity):

$$\mathcal{L}_F = t(\boldsymbol{x}) \ln [D_\phi(\boldsymbol{x})] + (1 - t(\boldsymbol{x})) \ln [1 - D_\phi(\boldsymbol{x})] \quad (16)$$

where the binary scalar t is an indicator function identifies whether $\boldsymbol{x}$ is a real data point.

$$t(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \\ 0 & \boldsymbol{x} \sim G_\theta(\boldsymbol{z}), \forall \boldsymbol{z} \end{cases} \quad (17)$$

Similarly, a cross-entropy measurement error is employed to quantify the discrepancy between $D_\phi(G_\theta(\boldsymbol{z}))$ and the scalar measurement $m = D_\phi(\boldsymbol{x})$:

$$\begin{aligned} E_\theta(m, \boldsymbol{z}) = {} & m \ln [D_\phi(G_\theta(\boldsymbol{z}))] \\ & + (1 - m) \ln [1 - D_\phi(G_\theta(\boldsymbol{z}))] \end{aligned} \quad (18)$$

At the minimum of $\mathcal{L}_F = 0$ (eq. 16), the optimal measure-

ment function is achieved by the perfect classifier:

$$D_\phi(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \\ 0 & \boldsymbol{x} \sim G_\theta(\boldsymbol{z}), \forall \boldsymbol{z} \end{cases} \quad (19)$$

We can therefore simplify eq. 18 by replacing m with its target value 1 as in teacher-forcing (Williams & Zipser, 1989):

$$E(\boldsymbol{m}, \boldsymbol{z}) = \ln [D_\phi(G_\theta(\boldsymbol{z}))] \quad (20)$$

This objective recovers the vanilla GAN formulation with the commonly used alternative loss (Goodfellow et al., 2014), which we derived as a measurement error. When latent optimisation is disabled ($T = 0$), Algorithm 2 is identical to a vanilla GAN.

In our experiments (section 4.2), we observed that the additional latent optimisation steps introduced from the CS perspective significantly improved GAN training. We reckon this is because latent optimisation moves the representation to areas more likely to generate realistic images as deemed by the discriminator. Since the gradient descent process remains local, the latent representations are still spread broadly in latent space, which avoids mode collapse. Although a sufficiently powerful generator G_θ can transform the source $p_{\boldsymbol{z}}(\boldsymbol{z})$ into arbitrarily complex distribution, a more informative source, as implicitly manifested from the optimised $\boldsymbol{z}$, may significantly reduce the complexity required for G_θ, thus striking a better trade-off in terms of the overall computation.

3.2.3. GENERALISED CS 2: SEMI-SUPERVISED GANS

So far, we have shown two extreme cases of Deep Compressed Sensing: in one case, the distance preserving measurements (section 3.2.1) essentially encode all information for recovering the original signals; on the other hand, the CS-GAN (section 3.2.2) has one-dimensional measurements that only indicates whether signals are real or fake. We now seek a middle ground, by using measurements that preserve class information for labelled data.

We generalise CS-GAN by replacing the binary classifier (discriminator) D_ϕ with a multi-class classifier C_ϕ. For data with K classes, this classifier outputs $K + 1$ classes with the $(K + 1)$'th class reserved for "fake" data that comes from the generator. This specification is the same as the classifier used in semi-supervised GANs (SGANs, Salimans et al. (2016)). Consequently, we extend the binary indicator function in eq. 17 to multi-class indicator, so that its k'the element $t^k(\boldsymbol{x}) = 1$ when $\boldsymbol{x}$ in class k. The k'th output of the classifier $C_\phi^k(\boldsymbol{x})$ indicates the predicted probability that $\boldsymbol{x}$ is in the k'th class, and multi-class cross-entropy loss is

Table 1. A family of DCS models differentiated by the properties of measurements in comparison with CS. The CS measurement matrix does not need training, so it does not have a training loss.

MODEL	PROPERTY	LOSS
CS	RIP	N/A
DCS	TRAINED RIP	EQ. 14
CS-GAN	VALIDITY PRESERVING	EQ. 16
CS-SGAN	CLASS PRESERVING	EQ. 21

used for the measurement loss and measurement error:

$$\mathcal{L}_F = \sum_{k=1}^{K+1} t^k(\boldsymbol{x}) \ln \left[C_\phi^k(\boldsymbol{x}) \right] \tag{21}$$

$$E(\boldsymbol{m}, \boldsymbol{z}) = t^k(\boldsymbol{x}) \ln \left[C_\phi^k(G_\theta(\boldsymbol{z})) \right] \tag{22}$$

When latent optimisation is disabled ($T = 0$), the model is similar to other semi-supervised GANs (Salimans et al., 2016; Odena et al., 2017). However, when $T > 0$ the online optimisation moves latent representations towards regions representing particular classes. This provides a novel way of training conditional GANs.

Compared with conditional GANs which concatenate labels to latent variables (Mirza & Osindero, 2014), optimising latent variables is more adaptive and uses information from the entire model. Compared with Batch-Norm based methods (Miyato & Koyama, 2018), the information for conditioning is presented in the target measurements, and does not need to be trained as Batch-Norm statistics (Ioffe & Szegedy, 2015). Since both of these methods use separate sources (label inputs or batch statistics) to provide the condition, their latent variables tend to retain no information about the condition. Our model, on the other hand, distils the condition information into the latent representation, which results in semantically meaningful latent space (Figure 5).

3.3. Optimising Models

The three models we derived as examples in the DCS framework are summarised in Table 1 along side CS. The main difference between them lies is the training objective used for the measurement functions $\mathcal{L}_F$. Once $\mathcal{L}_F$ is specified, the generator objective $\mathcal{L}_G$, in the form of measurement error, can be derived follow suit. When $\mathcal{L}_F$ and $\mathcal{L}_G$ are adversarial, such as in the CS-GAN, F_ϕ and G_θ need to be optimised separately as in GANs. This is implemented as the alternating update option in Algorithm 2. In optimising the latent variables (eq. 7), we normalise $\hat{\boldsymbol{z}}$ after each gradient descent step, as in (Bojanowski et al., 2018). We treat the step size α in latent optimisation as a parameter and back-propagate through it in optimising the model loss function. An additional technique we found useful in stabilising CS-GAN training is to penalise the distance $\boldsymbol{z}$ moves

as an optimisation cost and add it to $\mathcal{L}_G$:

$$\mathcal{L}_O = \beta \cdot \|\hat{\boldsymbol{z}} - \boldsymbol{z}_0\|_2^2 \tag{23}$$

where β is a scalar controlling the strength of this regulariser. This regulariser encourages small moves of $\boldsymbol{z}$ in optimisation, and can be interpreted as approximating an *optimal transport* cost (Villani, 2008). We found a range of β from 1.0 to 10.0 made little difference in training, and used $\beta = 3.0$ in our experiments with CS-GAN.

4. Experiments

4.1. Deep Compressed Sensing for Reconstruction

We first evaluate the DCS model using the MNIST (Yann et al., 1998) and CelebA (Liu et al., 2015) datasets. To compare with the approach in Bora et al. (2017), we used the same generators as in their model. For the measurements functions, we considered both linear projection and neural networks. We considered both random projections and trained measurement functions, while the generator was always trained jointly with the latent optimisation process. Unless otherwise specified, we use 3 gradient descent steps for latent optimisation. More details, including hyperparameter values, are reported in the Appendix. Our code will be available at `https://github.com/deepmind/deep-compressed-sensing`.

Tables 2 and 3 summarise the results from our models as well as the baseline model from Bora et al. (2017). The reconstruction loss for the baseline model is estimated from Figure 1 in Bora et al. (2017). DCS performs significantly better than the baseline. In addition, while the baseline model used hundreds or thousands of gradient-descent steps with several re-starts, we only used 3 steps without any re-starting, achieving orders of magnitudes higher efficiency. Interestingly, for fixed F, random linear projections outperformed neural networks as the measurement functions in both datasets across different neural network structures (row 2 and 3 of Table 2 and 3). This empirical result is consistent with the optimality of random projections described in the compressed sensing literature and the more general Johnson-Lindenstrauss lemma (Donoho, 2006; Candes et al., 2006; Johnson & Lindenstrauss, 1984).

The advantage of neural networks manifested when F_ϕ was optimised; this variant reached the best performance in all scenarios. As argued in (Weiss et al., 2007), we observed that random projections are sub-optimal for highly structured signals such as images, as seen in the improved performance when optimising the measurement matrices (row 4 of Table 2 and 3). The reconstruction performance was further improve when the linear measurement projections were replaced by neural networks (row 5 of Table 2 and 3). Examples of reconstructed MNIST images from different models are shown in Figure 2.

Table 2. Reconstruction loss on MNIST test data using different measurement functions. All rows except the first are from our models. "±" shows the standard deviation across test samples. (L) indicates learned measurement functions. Lower is better.

MODEL	10	25 MEASUREMENTS	STEPS
BASELINE	54.8	17.2	10×1000
LINEAR	10.8 ± 3.8	6.9 ± 2.7	3
NN	12.5 ± 2.2	10.2 ± 1.7	3
LINEAR(L)	6.5 ± 2.1	$4. \pm 1.4$	3
NN(L)	$\mathbf{5.3 \pm 1.9}$	$\mathbf{3.4 \pm 1.2}$	3

Table 3. Reconstruction loss on CelebA test data using different measurement functions. All rows except the first are from our models. "±" shows the standard deviation across test samples. (L) indicates learned measurement functions. Lower is better.

MODEL	20	50 MEASUREMENTS	STEPS
BASELINE	156.8	82.3	2×500
LINEAR	34.7 ± 7.9	27.1 ± 6.1	3
NN	46.1 ± 8.9	41.2 ± 8.3	3
LINEAR(L)	26.2 ± 5.9	20.5 ± 4.3	3
NN(L)	$\mathbf{23.4 \pm 5.8}$	$\mathbf{18.5 \pm 4.3}$	3

Unlike autoencoder-based methods, our models were not trained with any pixel reconstruction loss, which we only use for testing. Despite this, our results are comparable with the recently proposed "Uncertainty Autoencoders" (Grover & Ermon, 2018). We have worse MNIST reconstructions: with 10 and 25 measurements, ours best model achieved 5.3 and 3.4 per-image reconstruction errors compared with theirs 3.8 and 2.5 (estimated from figure 4). However, we achieved better CelebA results: with 20 and 50 measurements, we have errors of 23.4 and 18.5 compared with their 27 and 22 (estimated from Figure 6).

4.2. CS-GANs

To evaluate our proposed CS-GANs, we first trained a small model on MNIST to demonstrate intuitively the ad-
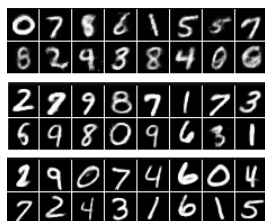

Figure 2. Reconstructions using 10 measurements from random linear projection (top), trained linear projection (middle), and trained neural network (bottom).

Figure 3. Samples from CS-GANs using 0 (left), 3 (central) and 5 (right) gradient descent steps in latent optimisation. The CS-GAN using 0 step was equivalent to a vanilla GAN.

vantage of latent optimisation. For quantitative evaluation, we trained larger and more standard models on CIFAR10 (Krizhevsky & Hinton, 2009), and evaluate them using the Inception Score (IS) (Salimans et al., 2016) and Frchet Inception Distance (FID) (Heusel et al., 2017). To our knowledge, latent optimisation has not been previously used to improving GANs, so our approach is orthogonal to existing methods such as Arjovsky et al. (2017); Miyato et al. (2018). We first compare our model with vanilla GANs, which is a special case of the CS-GAN (see section 3.2.2).

We use the same MNIST architectures as in section 4.1, but changed the the measurement function to a GAN discriminator (section 3.2.2). We use the alternating update option in Algorithm 2 in this setting. All other hyper-parameters are the same as in previous experiments. We use this relatively weak model to reveal failure modes as well as advantages of the CS-GAN. Figure 3 shows samples from models with the same setting but different latent optimisation iterations. The three panels show samples from models using 0, 3 and 5 gradient descent steps respectively. The model using 0 iteration was equivalent to a vanilla GAN. Optimising latent variables exhibits no mode collapse, one of the common failure modes of GAN training.

To confirm this advantage, we more systematically evaluate our method across a range of 144 hyper-parameters (similar to Kurach et al. (2018)). We use the CIFAR dataset which contains various categories of natural images, whose features from an Inception Network (Ioffe & Szegedy, 2015) are meaningful for evaluating the IS and FID. Other than the number of gradient descent steps (0 vs. 3) the model architectures and training procedures were identical. The change of IS and FID during training are plot in figure 4. CS-GANs achieved better performance in both IS and FID, and had less variance across the range of hyper-parameters. The blue horizontal lines at the bottom of Fig. 4 (left) and the top of Fig. 4 (right) shows failed vanilla GANs, but none of the CS-GANs diverged in training.

We also applied our latent optimisation method on Spectral-Normalised GANs (SN-GANs) (Miyato et al., 2018), which use Batch Normalisation (Ioffe & Szegedy, 2015) for the generator and Spectral Normalisation for the discriminator.

Table 4. Comparison with Spectral Normalised GANs.

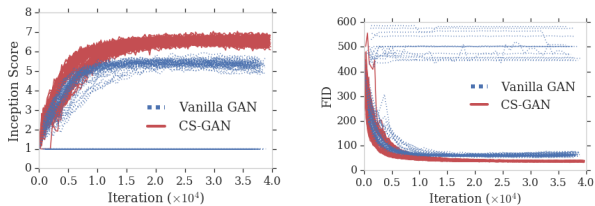|	SN-GAN	SN-GAN (OURS)	CS+SN-GAN
IS	7.42 ± 0.08	7.34 ± 0.07	$\mathbf{7.80 \pm 0.05}$
FID	29.3	29.53 ± 0.36	$\mathbf{23.13 \pm 0.50}$

Figure 4. Inception Score (higher is better) and FID (lower is better) during CIFAR training.

We compared our model with SN-GAN in Table 4: the SN-GAN column reproduces the numbers from (Miyato et al., 2018), and the next column are numbers from our replication of the same baseline. Our results demonstrate that deeper architectures, Batch Normalisation and Spectral Normalisation can further improve CS-GAN and that CS-GAN can improve upon a competitive baseline, SN-GAN.

4.3. CS-SGANs

We now experimentally assess our approach to use latent optimisation in semi-supervised GANs, CS-SGAN. We illustrate this extension with the MNIST dataset, and leave it to future work to study other applications. We keep all the hyper-parameters the same as in section 4.2, except changing the number of measurements to 11 for the 10 MNIST classes and 1 class reserved for generated samples. Samples from CS-SGAN can be seen in Figure 5 (left). Figure 5 (right) illustrates this with T-SNE (Maaten & Hinton, 2008) computed from 2000 random samples, where class labels are colour-coded. The latent space formed separated regions representing different digits. It is impossible to obtain such clustered latent space in typical conditional GANs (Mirza & Osindero, 2014; Miyato & Koyama, 2018), where labels are supplied as separate inputs while the random source only provides label-independent variations. In contrast, in our model the labels are distilled into latent representation via optimisation, leading to a more interpretable latent space.

5. Discussion

We present a novel framework for combining compressed sensing and deep neural networks. In this framework we trained both the measurement and generation functions, as well as the latent optimisation (i.e., reconstruction) procedure itself via meta-learning. Inspired by Bora et al. (2017),
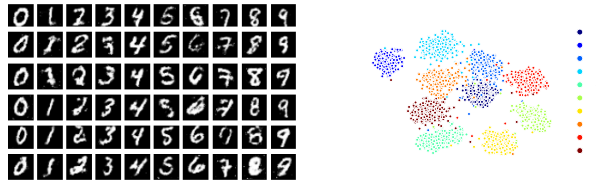

Figure 5. Left: samples from the generative classifier, and t-SNE illustration of the generator's latent space.

our approach significantly improves upon the performance and speed of reconstruction obtain in this work. In addition, we derived a family of models, including a novel GAN model, by expanding the set of properties we consider for the measurement function (Table 1).

Our method differs from existing algorithms that aim to combine compressed sensing with deep networks in that our approach preserves the online minimisation of measurement errors in generic neural networks. Previous attempts that combine CS and deep learning generally fall into two categories. One category of methods interprets taking compressed measurements and reconstructing from these measurements as an encoding-decoding problem and formulate the model as an autoencoder (Mousavi et al., 2015; Kulkarni et al., 2016; Mousavi et al., 2017; Grover & Ermon, 2018; Mousavi et al., 2018; Lu et al., 2018). Another category of methods are designed to mimic principled iterative CS algorithms using specialised network architectures (Metzler et al., 2017; Sun et al., 2016). In contrast, our framework maintains the separation of measurements from generation but still uses generic neural networks. Therefore, both the measurements and latent representation of the generator can be flexibly optimised for different, even adversarial, objectives, while taking advantage of powerful neural network architectures.

Moreover, we can train measurement functions with properties that are difficult or impossible to obtain from random or hand-crafted projections, thus broadening the range of problems that can be solved by minimising error measurements online. In other words, learning the measurement can be used as a useful stepping stone for learning complex tasks where the cost function is difficult to design directly. Our approach can also be interpreted as training implicit generative models, where explicit minimisation of divergences is replaced by statistical tests (Mohamed & Lakshminarayanan, 2016). We have illustrated this idea in the context of relatively simple tasks, but anticipate that complex tasks such as style transfer (Zhu et al., 2017), in areas already seen the applications of CS, as well as applications including MRI (Lustig et al., 2007) and unsupervised anomaly detection (Schlegl et al., 2017), may further benefit from our approach.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. Discriminator rejection sampling. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1GkToR5tm.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The mechanics of n-player differentiable games. *arXiv preprint arXiv:1802.05642*, 2018.

Baraniuk, R. G. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, 24(4):118–121, 2007.

Bishop, C. M. Pattern recognition and machine learning (information science and statistics) springer-verlag new york. *Inc. Secaucus, NJ, USA*, 2006.

Bojanowski, P., Joulin, A., Lopez-Pas, D., and Szlam, A. Optimizing the latent space of generative networks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 600–609, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/bojanowski18a.html.

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

Bourlard, H. and Kamp, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.

Candes, E. J., Romberg, J. K., and Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

Dhar, M., Grover, A., and Ermon, S. Modeling sparse deviations for compressed sensing using generative models. In *International Conference on Machine Learning*, pp. 1222–1231, 2018.

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

Duarte, M. F., Davenport, M. A., Takhar, D., Laska, J. N., Sun, T., Kelly, K. F., and Baraniuk, R. G. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Grover, A. and Ermon, S. Uncertainty autoencoders: Learning compressed representations via variational information maximization. *arXiv preprint arXiv:1812.10539*, 2018.

Hand, P. and Voroninski, V. Global guarantees for enforcing deep generative priors by empirical risk. *arXiv preprint arXiv:1705.07576*, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Hu, Z., Yang, Z., Salakhutdinov, R., and Xing, E. P. On unifying deep generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rylSzl-R-.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Johnson, W. B. and Lindenstrauss, J. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., and Ashok, A. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 449–458, 2016.

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Lu, X., Dong, W., Wang, P., Shi, G., and Xie, X. Convcsnet: A convolutional compressive sensing framework based on deep learning. *arXiv preprint arXiv:1801.10342*, 2018.

Lustig, M., Donoho, D., and Pauly, J. M. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

MacKay, D. J. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Metzler, C., Mousavi, A., and Baraniuk, R. Learned d-amp: Principled neural network based compressive image recovery. In *Advances in Neural Information Processing Systems*, pp. 1772–1783, 2017.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Miyato, T. and Koyama, M. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Mousavi, A., Patel, A. B., and Baraniuk, R. G. A deep learning approach to structured signal recovery. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1336–1343. IEEE, 2015.

Mousavi, A., Dasarathy, G., and Baraniuk, R. G. Deepcodec: Adaptive sensing and recovery via deep convolutional neural networks. *arXiv preprint arXiv:1707.03386*, 2017.

Mousavi, A., Dasarathy, G., and Baraniuk, R. G. A data-driven and distributed approach to sparse signal representation and recovery. In *International Conference on Learning Representations*, 2018.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2642–2651. JMLR. org, 2017.

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

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pp. 146–157. Springer, 2017.

Schmidhuber, J. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Sun, J., Li, H., Xu, Z., et al. Deep admm-net for compressive sensing mri. In *Advances in neural information processing systems*, pp. 10–18, 2016.

Tao, C., Chen, L., Henao, R., Feng, J., and Duke, L. C. Chi-square generative adversarial network. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4887–4896, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/tao18b.html.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Weiss, Y., Chang, H. S., and Freeman, W. T. Learning compressed sensing. In *Snowbird Learning Workshop, Allerton, CA*. Citeseer, 2007.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Yann, L., Corinna, C., and Burges, C. The mnist database of handwritten digits. *URL http://yhann. lecun. com/exdb/mnist*, 1998.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired
image-to-image translation using cycle-consistent adver-
sarial networks. *arXiv preprint*, 2017.