Compressed Sensing using Generative Models

Ashish Bora Ajil Jalal Eric Price Alex Dimakis

UT Austin

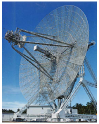
Compressed Sensing

- Want to recover a signal (e.g. an image) from noisy measurements.

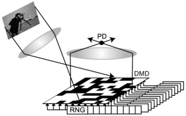
Compressed Sensing

- Want to recover a signal (e.g. an image) from noisy measurements.

Medical Imaging



Astronomy



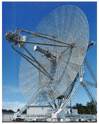
Single-Pixel Camera

Oil Exploration

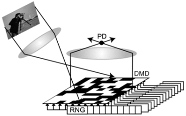
Compressed Sensing

- Want to recover a signal (e.g. an image) from noisy measurements.

Medical Imaging



Astronomy



Single-Pixel Camera

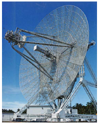
Oil Exploration

- *Linear* measurements: Choose $A \in \mathbb{R}^{m \times n}$, see $y = Ax$.

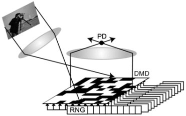
Compressed Sensing

- Want to recover a signal (e.g. an image) from noisy measurements.

Medical Imaging



Astronomy



Single-Pixel Camera

Oil Exploration

- *Linear* measurements: Choose $A \in \mathbb{R}^{m \times n}$, see $y = Ax$.
- How many measurements m to learn the signal?

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

5MB

36MB

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

5MB

36MB

  - This is why compression is possible.

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

5MB

36MB

  - This is why compression is possible.
- Ideal answer: (information in image) / (new info. per measurement)

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

5MB

36MB

  - This is why compression is possible.
- Ideal answer: (information in image) / (new info. per measurement)
  - Image "compressible" $\implies$ information in image is small.

Compressed Sensing

- Given linear measurements $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- How many measurements m to learn the signal x?
  - Naively: $m \geq n$ or else underdetermined: multiple x possible.
  - But most x aren't *plausible*.

5MB

36MB

  - This is why compression is possible.
- Ideal answer: (information in image) / (new info. per measurement)
  - Image "compressible" $\implies$ information in image is small.
  - Measurements "incoherent" $\implies$ most info new.

Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.

Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: Find "most compressible" image that fits measurements.

Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: Find "most compressible" image that fits measurements.
- Three questions:

Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: Find "most compressible" image that fits measurements.
- Three questions:
  - How should we formalize that an image is "compressible"?

Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: Find "most compressible" image that fits measurements.
- Three questions:
  - How should we formalize that an image is "compressible"?
  - What algorithm for recovery?

Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: Find "most compressible" image that fits measurements.
- Three questions:
  - How should we formalize that an image is "compressible"?
  - What algorithm for recovery?
  - How to choose the measurement matrix?

Compressed Sensing

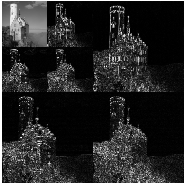
- Standard compressed sensing: *sparsity* in some basis

Compressed Sensing

- Standard compressed sensing: *sparsity* in some basis

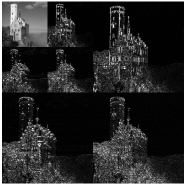
Compressed Sensing

- Standard compressed sensing: *sparsity* in some basis



Compressed Sensing

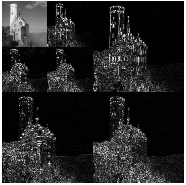
- Standard compressed sensing: *sparsity* in some basis



  - Sparsity + other constraints ("structured sparsity")

Compressed Sensing

- Standard compressed sensing: *sparsity* in some basis



  - Sparsity + other constraints ("structured sparsity")
- This talk: new method.

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \ \|Ax - y\|_2^2$$

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
    - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \; \|Ax - y\|_2^2 + \lambda \|x\|_1$$

Standard Compressed Sensing Formalism

"Compressible" $=$ "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \ \|Ax - y\|_2^2 + \lambda \|x\|_1$$

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

  $$min_x \; \|Ax - y\|_2^2 + \lambda \|x\|_1$$
- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

  - Reconstruction accuracy proportional to model accuracy.

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \ \|Ax - y\|_2^2 + \lambda \|x\|_1$$

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candes-Romberg-Tao 2006]

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \; \|Ax - y\|_2^2 + \lambda \|x\|_1$$

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candes-Romberg-Tao 2006]
  - A satisfies REC if for all k-sparse x, $\|Ax\| \geq \gamma x$

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \; \|Ax - y\|_2^2 + \lambda \|x\|_1$$

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candes-Romberg-Tao 2006]
  - A satisfies REC if for all k-sparse x, $\|Ax\| \geq \gamma x$
  - If A satisfies REC, then LASSO recovers $\widehat{x}$ satisfying (1)

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO
  $$min_x \ \|Ax - y\|_2^2 + \lambda \|x\|_1$$
- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candes-Romberg-Tao 2006]
  - A satisfies REC if for all k-sparse x, $\|Ax\| \geq \gamma x$
  - If A satisfies REC, then LASSO recovers $\widehat{x}$ satisfying (1)
  - $m = O(k \log(n/k))$ suffices for A to satisfy REC.

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO

$$min_x \ \|Ax - y\|_2^2 + \lambda\|x\|_1$$

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \le O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$

  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candes-Romberg-Tao 2006]
  - A satisfies REC if for all k-sparse x, $\|Ax\| \ge \gamma x$
  - If A satisfies REC, then LASSO recovers $\widehat{x}$ satisfying (1)
  - $m = O(k \log(n/k))$ suffices for A to satisfy REC.
- Theorem [Do Ba-Indyk-Price-Woodruff 2010]

Standard Compressed Sensing Formalism

"Compressible" = "sparse"

- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Algorithm for recovery: LASSO
  $$min_x \ \|Ax - y\|_2^2 + \lambda\|x\|_1$$
- Goal: $\widehat{x}$ with
  $$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{1}$$
  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candes-Romberg-Tao 2006]
  - A satisfies REC if for all k-sparse x, $\|Ax\| \geq \gamma x$
  - If A satisfies REC, then LASSO recovers $\widehat{x}$ satisfying (1)
  - $m = O(k \log(n/k))$ suffices for A to satisfy REC.
- Theorem [Do Ba-Indyk-Price-Woodruff 2010]
  - $m = \Theta(k \log(n/k))$ is optimal

Alternatives to sparsity?

- Basis needs to be handcrafted

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

- Billions of natural images, millions of MRIs collected each year

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

- Billions of natural images, millions of MRIs collected each year
- Opportunity to improve structural understanding

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

- Billions of natural images, millions of MRIs collected each year
- Opportunity to improve structural understanding
- Better structural understanding should give fewer measurements

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

- Billions of natural images, millions of MRIs collected each year
- Opportunity to improve structural understanding
- Better structural understanding should give fewer measurements
- Best way to model images in 2017?

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

- Billions of natural images, millions of MRIs collected each year
- Opportunity to improve structural understanding
- Better structural understanding should give fewer measurements
- Best way to model images in 2017?
    - Deep convolutional neural networks.

Alternatives to sparsity?

- Basis needs to be handcrafted
- Does not capture the plausible vectors tightly : Too simplistic
- Ignores a lot of domain dependent structure

- Billions of natural images, millions of MRIs collected each year
- Opportunity to improve structural understanding
- Better structural understanding should give fewer measurements
- Best way to model images in 2017?
  - Deep convolutional neural networks.
  - In particular: *generative models*.

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.

Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...

Generative Models

BeGAN

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...

Generative Models

BeGAN

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
- Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].

Generative Models

BeGAN

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
- Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].
  - Blurrier, but maybe better coverage of the space.

Generative Models

BeGAN

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Active area of machine learning research in last few years.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
- Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].
  - Blurrier, but maybe better coverage of the space.

Suggestion for compressed sensing

Replace "x is k-sparse" by "x is in range of $G : \mathbb{R}^k \to \mathbb{R}^n$".

Our Results

"Compressible" $=$ "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.

Our Results

"Compressible" $=$ "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.

Our Results

"Compressible" = "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.
- Algorithm for recovery

Our Results

"Compressible" $=$ "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.
- Algorithm for recovery
  - $min_z \; \|Ax - y\|_2^2 + \lambda \|x\|_1$

Our Results

"Compressible" = "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.
- Algorithm for recovery
  - $min_z \; \|AG(z) - y\|_2^2 + \lambda \|z\|_2^2$

Our Results

"Compressible" $=$ "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.
- Algorithm for recovery
  - $min_z \; \|AG(z) - y\|_2^2 + \lambda \|z\|_2^2$
  - Backprop to get gradients wrt z.

Our Results

"Compressible" = "near range(G)"

- Want to estimate x from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.
- Algorithm for recovery
  - $min_z \; \|AG(z) - y\|_2^2 + \lambda \|z\|_2^2$
  - Backprop to get gradients wrt z.
  - Optimize with gradient descent

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2 \tag{2}$$

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \tag{2}$$

    - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
    - G is a d-layer ReLU-based neural network.
    - When A is random Gaussian matrix.
- Main Theorem II:
    - For any Lipschitz G, $m = O(k \log L)$ suffices.

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'), \|z'\|_2 \leq r} \|x - x'\|_2 + \delta \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'), \|z'\|_2 \leq r} \|x - x'\|_2 + \delta \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.
  - Morally the same $O(kd \log n)$ bound: $L, r, \delta^{-1} \sim n^{O(d)}$

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'), \|z'\|_2 \leq r} \|x - x'\|_2 + \delta \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.
  - Morally the same $O(kd \log n)$ bound: $L, r, \delta^{-1} \sim n^{O(d)}$
- Convergence:

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'),\|z'\|_2 \leq r} \|x - x'\|_2 + \delta \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.
  - Morally the same $O(kd \log n)$ bound: $L, r, \delta^{-1} \sim n^{O(d)}$
- Convergence:
  - Just like training, no proof that gradient descent converges

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'),\|z'\|_2 \leq r} \|x - x'\|_2 + \delta \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.
  - Morally the same $O(kd \log n)$ bound: $L, r, \delta^{-1} \sim n^{O(d)}$
- Convergence:
  - Just like training, no proof that gradient descent converges
  - Approximate solution approximately gives (2)

Our Results

"Compressible" = "near range(G)"

- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'), \|z'\|_2 \leq r} \|x - x'\|_2 + \delta \qquad (2)$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.
  - Morally the same $O(kd \log n)$ bound: $L, r, \delta^{-1} \sim n^{O(d)}$
- Convergence:
  - Just like training, no proof that gradient descent converges
  - Approximate solution approximately gives (2)
  - Can check that $\|\widehat{x} - x\|_2$ is small.

Our Results

"Compressible" = "near range(G)"

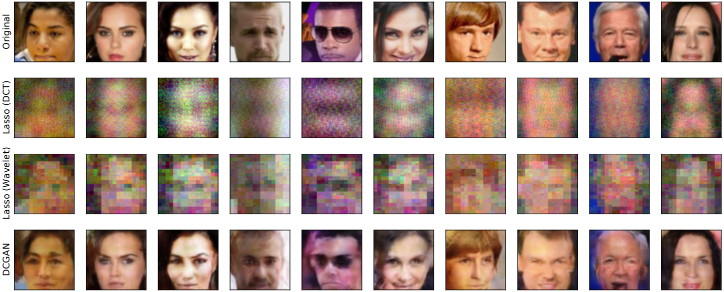
- Goal: $\widehat{x}$ with

$$\|x - \widehat{x}\|_2 \leq O(1) \cdot \min_{x'=G(z'), \|z'\|_2 \leq r} \|x - x'\|_2 + \delta \tag{2}$$

  - Reconstruction accuracy proportional to model accuracy.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
  - G is a d-layer ReLU-based neural network.
  - When A is random Gaussian matrix.
- Main Theorem II:
  - For any Lipschitz G, $m = O(k \log \frac{rL}{\delta})$ suffices.
  - Morally the same $O(kd \log n)$ bound: $L, r, \delta^{-1} \sim n^{O(d)}$
- Convergence:
  - Just like training, no proof that gradient descent converges
  - Approximate solution approximately gives (2)
  - Can check that $\|\widehat{x} - x\|_2$ is small.
  - In practice, optimization error is negligible.

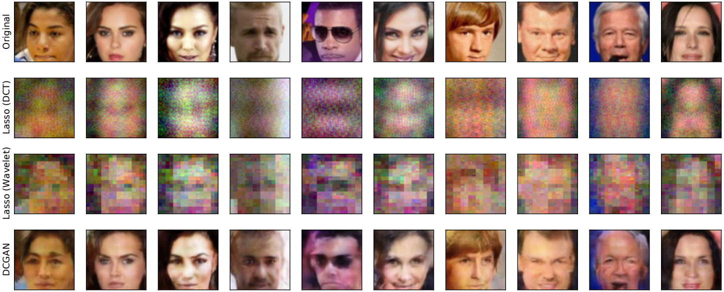
Experimental Results

Faces: $n = 64 \times 64 \times 3 = 12288$, $m = 500$

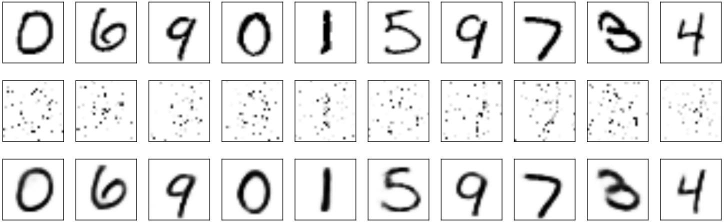


Experimental Results

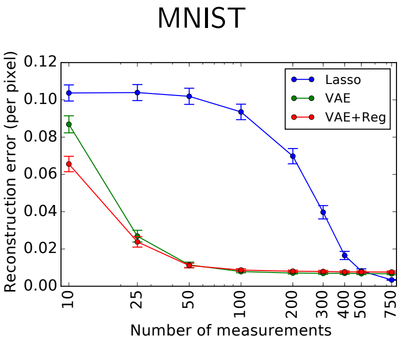
Faces: $n = 64 \times 64 \times 3 = 12288$, $m = 500$

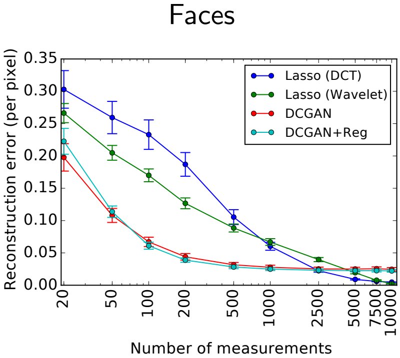


MNIST: $n = 28x28 = 784$, $m = 100$.

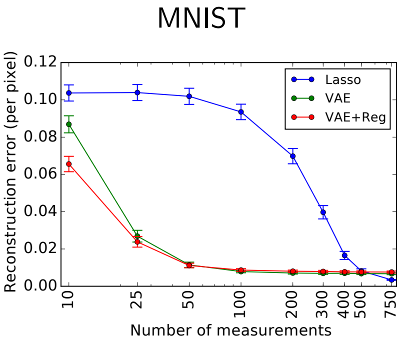


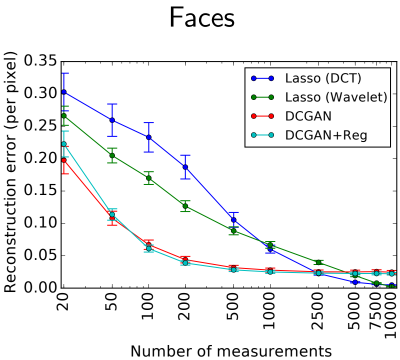
Experimental Results





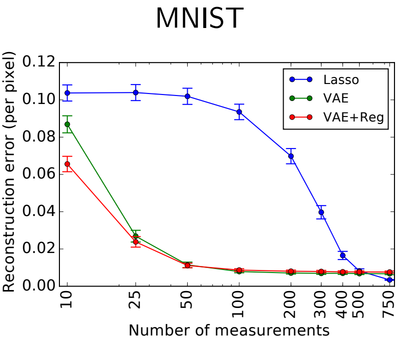
Experimental Results

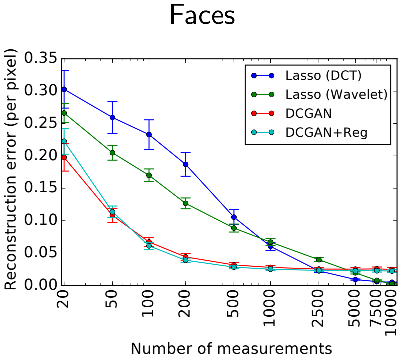




- For fixed G, have fixed k, so error stops improving after some point.

Experimental Results





- For fixed G, have fixed k, so error stops improving after some point.
- Larger m should use higher capacity G, so $\min \|x - G(z)\|$ smaller.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:
  - Each layer is $z \to \text{ReLU}(A_i z)$.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:
  - Each layer is $z \rightarrow \mathrm{ReLU}(A_i z)$.
  - $\mathrm{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:
  - Each layer is $z \to \mathrm{ReLU}(A_i z)$.
  - $\mathrm{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$
- Input to layer 1: single k-dimensional hyperplane.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:
  - Each layer is $z \to \mathrm{ReLU}(A_i z)$.
  - $\mathrm{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$
- Input to layer 1: single k-dimensional hyperplane.

Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

Proof Outline (ReLU-based networks)

- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:
  - Each layer is $z \to \mathrm{ReLU}(A_i z)$.
  - $\mathrm{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$
- Input to layer 1: single k-dimensional hyperplane.

Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- Induction: final output lies within n^{dk} k-dimensional hyperplanes.

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- $\mathrm{ReLU}(A_1 z)$ is linear, within any constant region of $\mathrm{sign}(A_1 z)$.

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- $\mathrm{ReLU}(A_1 z)$ is linear, within any constant region of $\mathrm{sign}(A_1 z)$.
- How many different patterns can $\mathrm{sign}(A_1 z)$ take?

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- $\text{ReLU}(A_1 z)$ is linear, within any constant region of $\text{sign}(A_1 z)$.
- How many different patterns can $\text{sign}(A_1 z)$ take?
- $k = 2$ version

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- $\mathrm{ReLU}(A_1 z)$ is linear, within any constant region of $\mathrm{sign}(A_1 z)$.
- How many different patterns can $\mathrm{sign}(A_1 z)$ take?
- $k = 2$ version: how many regions can n lines partition plane into?

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- $\mathrm{ReLU}(A_1 z)$ is linear, within any constant region of $\mathrm{sign}(A_1 z)$.
- How many different patterns can $\mathrm{sign}(A_1 z)$ take?
- $k = 2$ version: how many regions can n lines partition plane into?
  - $1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2}$.

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- $\mathrm{ReLU}(A_1 z)$ is linear, within any constant region of $\mathrm{sign}(A_1 z)$.
- How many different patterns can $\mathrm{sign}(A_1 z)$ take?
- $k = 2$ version: how many regions can n lines partition plane into?
  - $1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2}$.
  - n half-spaces divide $\mathbb{R}^k$ into less than n^k regions.

Proof of Lemma

Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- $A_1 z$ is k-dimensional hyperplane in $\mathbb{R}^n$.
- ReLU($A_1 z$) is linear, within any constant region of sign($A_1 z$).
- How many different patterns can sign($A_1 z$) take?
- $k = 2$ version: how many regions can n lines partition plane into?
  - $1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2}$.
  - n half-spaces divide $\mathbb{R}^k$ into less than n^k regions. ■

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \mathrm{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
    - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.
- Apply to δ-cover of range(G).

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.
- Apply to δ-cover of range(G).
  - Comes from δ/L-cover of domain(G).

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.
- Apply to δ-cover of range(G).
  - Comes from δ/L-cover of domain(G).
  - Size $(\frac{rL}{\delta})^k$: union bound works for $m = O(k \log \frac{rL}{\delta})$.

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.
- Apply to δ-cover of range(G).
  - Comes from δ/L-cover of domain(G).
  - Size $(\frac{rL}{\delta})^k$: union bound works for $m = O(k \log \frac{rL}{\delta})$.
- x_1, x_2 lie close to cover $\implies$ additive $\delta\|A\|$ loss:

$$\|Ax_1 - Ax_2\| \geq 0.9\|x_1 - x_2\| - O(\delta n)$$

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.
- Apply to δ-cover of range(G).
  - Comes from δ/L-cover of domain(G).
  - Size $(\frac{rL}{\delta})^k$: union bound works for $m = O(k \log \frac{rL}{\delta})$.
- x_1, x_2 lie close to cover $\implies$ additive $\delta\|A\|$ loss:

$$\|Ax_1 - Ax_2\| \geq 0.9\|x_1 - x_2\| - O(\delta n)$$

- Hence

$$\|\widehat{x} - x\|_2 \leq C \min_{x' \in \text{range}(G)} \|x' - x\|_2 + \delta n$$

Proof outline (Lipschitz networks)

- Need that if $x_1, x_2 \in \text{range}(G)$ are very different, then $\|Ax_1 - Ax_2\|$ is large.
  - Hence can distinguish with noise.
- True for *fixed* x_1, x_2 with $1 - e^{-\Omega(m)}$ probability.
- Apply to δ-cover of range(G).
  - Comes from δ/L-cover of domain(G).
  - Size $(\frac{rL}{\delta})^k$: union bound works for $m = O(k \log \frac{rL}{\delta})$.
- x_1, x_2 lie close to cover $\implies$ additive $\delta\|A\|$ loss:

$$\|Ax_1 - Ax_2\| \geq 0.9\|x_1 - x_2\| - O(\delta n)$$

- Hence

$$\|\widehat{x} - x\|_2 \leq C \min_{x' \in \text{range}(G)} \|x' - x\|_2 + \delta$$

Non-gaussian measurements

- $A =$ subset of Fourier matrix: MRI

Non-gaussian measurements

- $A =$ subset of Fourier matrix: MRI
- $A =$ Gaussian blur: superresolution

Non-gaussian measurements

- $A =$ subset of Fourier matrix: MRI
- $A =$ Gaussian blur: superresolution
- $A =$ diagonal with zeros: inpainting

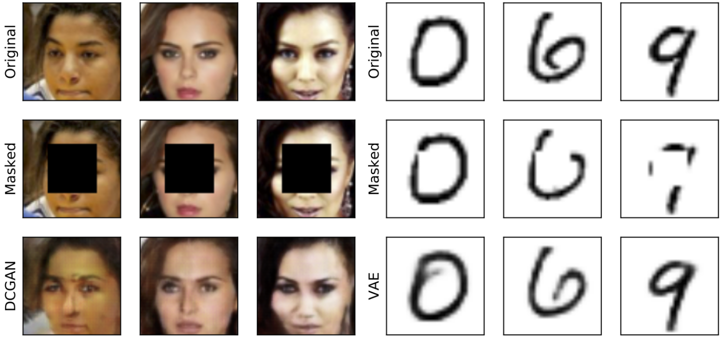
Non-gaussian measurements

- $A =$ subset of Fourier matrix: MRI
- $A =$ Gaussian blur: superresolution
- $A =$ diagonal with zeros: inpainting
- Algorithm can be applied, unifying problems.

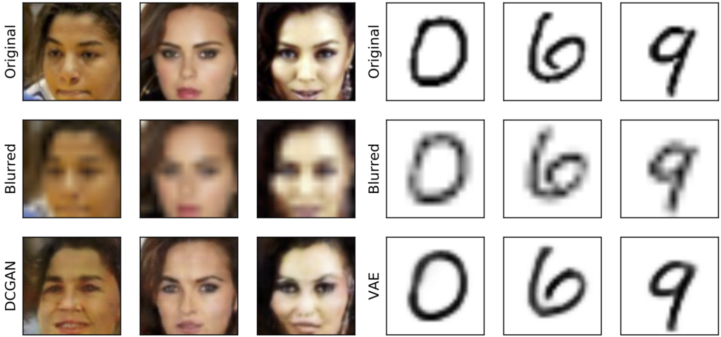
Non-gaussian measurements

- $A =$ subset of Fourier matrix: MRI
- $A =$ Gaussian blur: superresolution
- $A =$ diagonal with zeros: inpainting
- Algorithm can be applied, unifying problems.
  - Guarantee only holds if G and A are "incoherent".

Inpainting



Super-resolution



Summary

$$
\begin{aligned}
& \text{Number of measurements required to estimate an image} \\
= \;& \text{(information content of image) / (new information per measurement)}
\end{aligned}
$$

Summary

$$
\begin{aligned}
& \text{Number of measurements required to estimate an image} \\
= \quad & \text{(information content of image) / (new information per measurement)}
\end{aligned}
$$

- Generative models can bound information content as $O(kd \log n)$.

Summary

$$\begin{aligned} & \text{Number of measurements required to estimate an image} \\ = \quad & \text{(information content of image) / (new information per measurement)} \end{aligned}$$

- Generative models can bound information content as $O(kd \log n)$.
  - Open: do real networks require linear in d?

Summary

$$
\begin{aligned}
& \text{Number of measurements required to estimate an image} \\
= \;& \text{(information content of image) / (new information per measurement)}
\end{aligned}
$$

- Generative models can bound information content as $O(kd \log n)$.
  - Open: do real networks require linear in d?
- Generative models differentiable $\implies$ can optimize in practice.

Summary

$$\begin{aligned} & \text{Number of measurements required to estimate an image} \\ = \quad & \text{(information content of image) / (new information per measurement)} \end{aligned}$$

- Generative models can bound information content as $O(kd \log n)$.
  - Open: do real networks require linear in d?
- Generative models differentiable $\implies$ can optimize in practice.
- Gaussian measurements ensure independent information.

Future work

- More uses of differentiable compression?

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
    - *Metric* of how good generative model is.

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.
    - Bojanowski et al. 2017: impressive results with this.

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.
    - Bojanowski et al. 2017: impressive results with this.
- Computable proxy for Lipschitz parameter?

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.
    - Bojanowski et al. 2017: impressive results with this.
- Computable proxy for Lipschitz parameter?
  - Real networks may be much better than worst-case n^d.

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.
    - Bojanowski et al. 2017: impressive results with this.
- Computable proxy for Lipschitz parameter?
  - Real networks may be much better than worst-case n^d.
- Competitive guarantee for non-Gaussian A?

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.
    - Bojanowski et al. 2017: impressive results with this.
- Computable proxy for Lipschitz parameter?
  - Real networks may be much better than worst-case n^d.
- Competitive guarantee for non-Gaussian A?
  - Even nonlinear A?

Future work

- More uses of differentiable compression?
- More applications of distance to range of G?
  - *Metric* of how good generative model is.
  - *Objective to train* new generative model.
    - Bojanowski et al. 2017: impressive results with this.
- Computable proxy for Lipschitz parameter?
  - Real networks may be much better than worst-case n^d.
- Competitive guarantee for non-Gaussian A?
  - Even nonlinear A?

Thank You