Compressed Sensing and Generative Models

Ashish Bora  Ajil Jalal  Eric Price  Alex Dimakis

UT Austin
Talk Outline

1. Using generative models for compressed sensing

2. Learning generative models from noisy data
1 Using generative models for compressed sensing

2 Learning generative models from noisy data
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.
Compressed Sensing

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

- **Medical Imaging**
- **Astronomy**
- **Single-Pixel Camera**
- **Oil Exploration**

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

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

Medical Imaging  Astronomy  Single-Pixel Camera  Oil Exploration

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

Naively: $m \geq n$ or else underdetermined: multiple $x$ possible.

But most $x$ aren't plausible.

This is why compression is possible.

Ideal answer: $m > \text{(information in image)}$ (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

This is why compression is possible.

Ideal answer: $m > \text{(information in image)}$ (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.
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.

Ideal answer: $m > (\text{information in image}) \times (\text{new info. per measurement})$
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.

- This is why compression is possible.

5MB vs. 36MB

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
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.

- This is why compression is possible.
- Ideal answer:

$$m > \frac{(\text{information in image})}{(\text{new info. per measurement})}$$
Compressed Sensing

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

$$m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}}$$
Compressed Sensing

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

\[
m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}}
\]

- 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 $x$?

$$ m > \frac{(\text{information in image})}{(\text{new info. per measurement})} $$

- 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: the “most compressible” image that fits measurements.
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- 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: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?
- Short JPEG compression
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?
- Short–JPEG compression
  - Intractible to compute.
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?  
  - Short JPEG compression
    - Intractible to compute.
- Standard compressed sensing: *sparsity* in some basis
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?
- Short-JPEG compression
  - Intractible to compute.
- Standard compressed sensing: sparsity in some basis
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?
- **Short JPEG compression**
  - Intractible to compute.
- Standard compressed sensing: *sparsity* in some basis
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?
- Short JPEG compression
  - Intractible to compute.
- Standard compressed sensing: sparsity in some basis

- Sparsity + other constraints ("structured sparsity")
Compressed Sensing

- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the “most compressible” image that fits measurements.
- How should we formalize that an image is “compressible”?
- Short JPEG compression
  - Intractible to compute.
- Standard compressed sensing: sparsity in some basis
  - Sparsity + other constraints ("structured sparsity")
- This talk: different approach, no sparsity.
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 $\eta$, so $y = Ax$.

Goal: $\hat{x}$ with
\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2(1)
\]
with high probability.

Reconstruction accuracy proportional to model accuracy.

Theorem [Candes-Romberg-Tao 2006]
$m = \Theta(\frac{k \log(n/k)}{})$ suffices for (1).

Such an $\hat{x}$ can be found efficiently with the 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 $\eta$, so $y = Ax$. 

Theorem [Candès-Romberg-Tao 2006]

$m = \Theta(k \log(n/k))$ suffices for (1).

Such an $\hat{x}$ can be found efficiently with the 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 \( \eta \), so \( y = Ax \).
- Goal: \( \hat{x} \) with
  \[
  \|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2
  \]

  with high probability.

\[ (1) \]
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 $\eta$, so $y = Ax$.

- Goal: $\hat{x}$ with
  
  $$
  \|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2
  $$

  with high probability.
  - 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 $\eta$, so $y = Ax$.

- Goal: $\hat{x}$ with
  \[
  \|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2
  \]
  \[\text{(1)}\]
  with high probability.

  - Reconstruction accuracy proportional to model accuracy.

- Theorem [Candès-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 $\eta$, so $y = Ax$.
- Goal: $\hat{x}$ with
  \[\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2\]  
  with high probability.
  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candès-Romberg-Tao 2006]
  - $m = \Theta(k \log(n/k))$ suffices for (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 \( \eta \), so \( y = Ax \).
- Goal: \( \hat{x} \) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2
\]

(1)

with high probability.
- Reconstruction accuracy proportional to model accuracy.

- Theorem [Candès-Romberg-Tao 2006]
  - \( m = \Theta(k \log(n/k)) \) suffices for (1).
  - Such an \( \hat{x} \) can be found efficiently with the LASSO.
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
- Worldwide, 100 million MRIs taken per year.
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
- Worldwide, 100 million MRIs taken per year.
- Want a *data-driven model*.
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
- Worldwide, 100 million MRIs taken per year.
- Want a *data-driven model*.
  - Better structural understanding should give fewer measurements.

Best way to model images in 2018?
- In particular: generative models.
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
- Worldwide, 100 million MRIs taken per year.
- Want a *data-driven model*.
  - Better structural understanding should give fewer measurements.
- Best way to model images in 2018?
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
- Worldwide, 100 million MRIs taken per year.
- Want a *data-driven model*.
  - Better structural understanding should give fewer measurements.
- Best way to model images in 2018?
Alternatives to sparsity?

- MRI images are sparse in the wavelet basis.
- Worldwide, 100 million MRIs taken per year.
- Want a *data-driven model*.
  - Better structural understanding should give fewer measurements.
- Best way to model images in 2018?
  - In particular: *generative models.*
Random noise $z$
Generative Models

Random noise $z$
Generative Models

Random noise $z$ → Image
Training Generative Models

Random noise $z$
Training Generative Models

Random noise $z$ → \[ \text{transform} \] → \[ \text{network} \] → \[ \text{output} \] → Image
Training Generative Models

Random noise $z$
Training Generative Models

Random noise $z$ → Image $n$
Training Generative Models

Random noise $z$ → $n$ → Image
Training Generative Models

Random noise $z$ \[ k \] \[ n \] Image
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 Adversarial Networks (GANs) [Goodfellow et al. 2014]:
- Competition between generator and discriminator.
- W-GAN, BeGAN, InfoGAN, DCGAN, ...
- Remarkably effective at generating realistic-looking images.

Karras et al., 2018
Schawinski et al., 2017
Faces
Astronomy
Paganini et al., 2017
Particle Physics

Variational Auto-Encoders (VAEs) [Kingma & Welling 2013]:
- Blurrier, but maybe better coverage of the space.
Generative Models

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

Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
- Competition between generator and discriminator.
- W-GAN, BeGAN, InfoGAN, DCGAN, ...
- Remarkably effective at generating realistic-looking images.

Karras et al., 2018
Faces Schawinski et al., 2017
Astronomy Paganini et al., 2017
Particle Physics

Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].
- Blurrier, but maybe better coverage of the space.
Generative Models

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

  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Karras et al., 2018
Faces Schawinski et al., 2017
Astronomy Paganini et al., 2017
Particle Physics

Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].

- Blurrer, but maybe better coverage of the space.
Generative Models

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

Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:

- Competition between generator and discriminator.
- W-GAN, BeGAN, InfoGAN, DCGAN, ...
Generative Models

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

Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.
Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim \mathcal{N}(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Faces

Karras et al., 2018
Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$.
- When $z \sim \mathcal{N}(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Faces

Astronomy

Karras et al., 2018
Schawinski et al., 2017
Generative Models

- Want to model a distribution $\mathcal{D}$ of images.
- Function $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim \mathcal{D}$.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Karras et al., 2018  Schawinski et al., 2017  Paganini et al., 2017

Faces  Astronomy  Particle Physics
Generative Models

- Want to model a distribution \( \mathcal{D} \) of images.
- Function \( G : \mathbb{R}^k \rightarrow \mathbb{R}^n \).
- When \( z \sim N(0, I_k) \), then ideally \( G(z) \sim \mathcal{D} \).
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Faces

Astronomy

Particle Physics

Karras et al., 2018  Schawinski et al., 2017  Paganini et al., 2017

Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].
Generative Models

- Want to model a distribution $D$ of images.
- Function $G : \mathbb{R}^k \to \mathbb{R}^n$.
- When $z \sim N(0, I_k)$, then ideally $G(z) \sim D$.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].
  - Blurrier, but maybe better coverage of the space.

Karras et al., 2018  Schawinski et al., 2017  Paganini et al., 2017
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 Adversarial Networks (GANs) [Goodfellow et al. 2014]:
  - Competition between generator and discriminator.
  - W-GAN, BeGAN, InfoGAN, DCGAN, ...
  - Remarkably effective at generating realistic-looking images.

Suggestion for compressed sensing

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

Variational Auto-Encoders (VAEs) [Kingma & Welling 2013].
  - Blurrier, but maybe better coverage of the space.
Our Results

“Compressible” = “near range($G$)”

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

Goal:
\[
\hat{x} \text{ with } \|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

Reconstruction accuracy proportional to model accuracy.

We are given the generative model $G$: $\mathbb{R}^k \rightarrow \mathbb{R}^n$.

Main Theorem I:
$m = O(\sqrt{d \log n})$ suffices for (2).

$G$ is a $d$-layer ReLU-based neural network.

When $A$ is random Gaussian matrix.

Main Theorem II:

For any Lipschitz $G$, $m = O(\sqrt{k \log rL_\delta})$ suffices.

Morally the same $O(\sqrt{kd \log n})$ bound.

Ashish Bora, Ajil Jalal, **Eric Price**, Alex Dimakis (UT Austin)
Our Results

“Compressible” = “near range(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{k\text{-sparse } x'} \|x - x'\|_2
\]  

\[(2)\]

\(\hat{x}\) is a reconstruction of \(x\) with accuracy proportional to model accuracy.
Our Results

“Compressible” = “near range(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]  

(2)
Our Results

“Compressible” = “near range(G)"

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

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

- Reconstruction accuracy proportional to model accuracy.
Our Results

“Compressible” = “near range(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

(2)

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model \(G : \mathbb{R}^k \rightarrow \mathbb{R}^n\).
Our Results

“Compressible” = “near range($G$)"

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

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$.
- Main Theorem I: $m = O(kd \log n)$ suffices for (2).
Our Results

“Compressible” = “near range(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

1. Reconstruction accuracy proportional to model accuracy.
2. We are given the generative model \(G : \mathbb{R}^k \rightarrow \mathbb{R}^n\).

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

- Main Theorem II:

\[
\text{For any Lipschitz } G, m = O(kd \log rL \delta) \text{ suffices.}
\]

\[
\text{Morally the same } O(kd \log n) \text{ bound.}
\]
Our Results

“Compressible” = “near range(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model \(G : \mathbb{R}^k \rightarrow \mathbb{R}^n\).

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.

\[\text{Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)}\]
Our Results

“Compressible” = “near range(G)”

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

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

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$.

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(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

(2)

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model \(G : \mathbb{R}^k \rightarrow \mathbb{R}^n\).

- **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(\(G\))”

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' = G(z'), \|z'\|_2 \leq r} \|x - x'\|_2 + \delta
\]

(2)

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model \(G : \mathbb{R}^k \to \mathbb{R}^n\).

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)"

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

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

(2)

- Reconstruction accuracy proportional to model accuracy.
- We are given the generative model $G : \mathbb{R}^k \to \mathbb{R}^n$.
- 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.
Our Results (II)

“Compressible” = “near range(G)”

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

Goal: $\hat{x}$ with

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

$m = O(kd \log n)$ suffices for $d$-layer $G$. 
Our Results (II)

“Compressible” = “near range(G)”

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

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]  

(3)

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.

\[\]
Our Results (II)

“Compressible” = “near range(G)”

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

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

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - $k$ here can be much smaller
Our Results (II)

“Compressible” = “near range(\(G\))"

- Want to estimate \(x\) from \(y = Ax\), for \(A \in \mathbb{R}^{m \times n}\).
- Goal: \(\hat{x}\) with

\[
\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
\]

(3)

- \(m = O(kd \log n)\) suffices for \(d\)-layer \(G\).
  - Compared to \(O(k \log n)\) for sparsity-based methods.
  - \(k\) here can be much smaller
  - \(d = 3 - 6\) for current generative models.
“Compressible” = “near range($G$)"

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

$$\|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2 \quad (3)$$

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - $k$ here can be much smaller
  - $d = 3 - 6$ for current generative models.
- Find $\hat{x} = G(\hat{z})$ by gradient descent on $\|y - AG(\hat{z})\|_2$. 
Our Results (II)

“Compressible” = “near range($G$)”

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

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

(3)

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - $k$ here can be much smaller
  - $d = 3 - 6$ for current generative models.
- Find $\hat{x} = G(\hat{z})$ by gradient descent on $||y - AG(\hat{z})||_2$.
  - Just like for training, no proof this converges
Our Results (II)

“Compressible” = “near range($G$)"

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

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

(3)

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - $k$ here can be much smaller
  - $d = 3 - 6$ for current generative models.
- Find $\hat{x} = G(\hat{z})$ by gradient descent on $\|y - AG(\hat{z})\|_2$.
  - Just like for training, no proof this converges
  - Approximate solution approximately gives (3)
Our Results (II)

“Compressible” = “near range(G)”

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

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

(3)

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - $k$ here can be much smaller
  - $d = 3 - 6$ for current generative models.
- Find $\hat{x} = G(\hat{z})$ by gradient descent on $\|y - AG(\hat{z})\|_2$.
  - Just like for training, no proof this converges
  - Approximate solution approximately gives (3)
  - Can check that $\|\hat{x} - x\|_2$ is small.
Our Results (II)

“Compressible” = “near range($G$)”

- Want to estimate $x$ from $y = Ax$, for $A \in \mathbb{R}^{m \times n}$.
- Goal: $\hat{x}$ with
  \[
  \|x - \hat{x}\|_2 \leq O(1) \cdot \min_{x' \in \text{range}(G)} \|x - x'\|_2
  \]  
  (3)

- $m = O(kd \log n)$ suffices for $d$-layer $G$.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - $k$ here can be much smaller
  - $d = 3 - 6$ for current generative models.

- Find $\hat{x} = G(\hat{z})$ by gradient descent on $\|y - AG(\hat{z})\|_2$.
  - Just like for training, no proof this converges
  - Approximate solution approximately gives (3)
  - Can check that $\|\hat{x} - x\|_2$ is small.
  - In practice, optimization error is negligible.
Related Work

- Model-based compressed sensing (Baraniuk-Cevher-Duarte-Hegde ’10)

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$

  $\mathbf{mod}_{-}\text{based compressed sensing} \ (\text{Baraniuk-Cevher-Duarte-Hegde '10})$
Related Work

- Model-based compressed sensing (Baraniuk-Cevher-Duarte-Hegde ’10)
  - $k$-sparse + more $\implies O(k)$ measurements.
Related Work

- Model-based compressed sensing (Baraniuk-Cevher-Duarte-Hegde ’10)
  - $k$-sparse + more $\implies O(k)$ measurements.
- Projections on manifolds (Baraniuk-Wakin ’09, Eftekhar-Wakin ’15)
Related Work

- Model-based compressed sensing (Baraniuk-Cevher-Duarte-Hegde ’10)
  - \( k \)-sparse + more \( \implies \mathcal{O}(k) \) measurements.

- Projections on manifolds (Baraniuk-Wakin ’09, Eftekhar-Wakin ’15)
  - Conditions on manifold for which recovery is possible.
Related Work

- Model-based compressed sensing (Baraniuk-Cevher-Duarte-Hegde ’10)
  - $k$-sparse + more $\implies O(k)$ measurements.
- Projections on manifolds (Baraniuk-Wakin ’09, Eftekhar-Wakin ’15)
  - Conditions on manifold for which recovery is possible.
- Deep network models (Mousavi-Dasardathy-Baraniuk ’17, Chang et al ’17)
Related Work

- Model-based compressed sensing (Baraniuk-Cevher-Duarte-Hegde ’10)
  - $k$-sparse + more $\implies O(k)$ measurements.
- Projections on manifolds (Baraniuk-Wakin ’09, Eftekhari-Wakin ’15)
  - Conditions on manifold for which recovery is possible.
- Deep network models (Mousavi-Dasarathy-Baraniuk ’17, Chang et al ’17)
  - Train deep network to encode and/or decode.
Experimental Results

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

Original

![Original Image](image1)

![Original Image](image2)

![Original Image](image3)

![Original Image](image4)

![Original Image](image5)
Experimental Results

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

Original

Lasso (DCT)

Lasso (Wavelet)
Experimental Results

Faces: \( n = 64 \times 64 \times 3 = 12288, \ m = 500 \)
Experimental Results

MNIST: $n = 28 \times 28 = 784, \ m = 100$.

Original

Lasso

VAE

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
Compressed Sensing and Generative Models
Experimental Results

MNIST

![Graph showing reconstruction error for MNIST with different methods.]

Faces

![Graph showing reconstruction error for Faces with different methods.]

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.
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:
  - Each layer is $z \rightarrow \text{ReLU}(A_i z)$.
  - $\text{ReLU}(y_i) = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$
  - Input to layer 1: single $k$-dimensional hyperplane.

Induction: final output lies within $n^{dk}$ $k$-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 $z \rightarrow \text{ReLU}(A_i z)$.
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 $z \rightarrow \text{ReLU}(A_i z)$.
Proof Outline (ReLU-based networks)

- Show range($G$) lies within union of $n^d k$-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 $z \rightarrow \text{ReLU}(A_i z)$.
  - $\text{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$
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 $z \rightarrow \text{ReLU}(A_iz)$.
  - $\text{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$

- Input to layer 1: single $k$-dimensional hyperplane.
Proof Outline (ReLU-based networks)

- Show range($G$) lies within union of $n^d k$ $k$-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 $z \rightarrow \text{ReLU}(A_i z)$.
- $\text{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$

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 $z \rightarrow \text{ReLU}(A_i z)$.
  - $\text{ReLU}(y)_i = \begin{cases} y_i & y_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$

- 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.

- $z$ is $k$-dimensional.
Proof of Lemma

Layer 1’s output lies within a union of \( n^k \) \( k \)-dimensional hyperplanes.

- \( z \) is \( k \)-dimensional.
- \( \text{ReLU}(A_1 z) \) is linear, within any constant region of \( \text{sign}(A_1 z) \).
Proof of Lemma

Layer 1’s output lies within a union of \( n^k \) \( k \)-dimensional hyperplanes.

- \( z \) is \( k \)-dimensional.
- \( \text{ReLU}(A_1z) \) is linear, within any constant region of \( \text{sign}(A_1z) \).
- How many different patterns can \( \text{sign}(A_1z) \) take?
Proof of Lemma

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

- $z$ is $k$-dimensional.
- ReLU($A_1z$) is linear, within any constant region of sign($A_1z$).
- How many different patterns can sign($A_1z$) take?
- $k = 2$ version
Proof of Lemma

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

- $z$ is $k$-dimensional.
- $\text{ReLU}(A_1z)$ is linear, within any constant region of $\text{sign}(A_1z)$.
- How many different patterns can $\text{sign}(A_1z)$ 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.

- $z$ is $k$-dimensional.
- ReLU($A_1z$) is linear, within any constant region of sign($A_1z$).
- How many different patterns can sign($A_1z$) 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.

- $z$ is $k$-dimensional.
- ReLU($A_1z$) is linear, within any constant region of sign($A_1z$).
- How many different patterns can sign($A_1z$) 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.

- $z$ is $k$-dimensional.
- ReLU($A_1z$) is linear, within any constant region of sign($A_1z$).
- How many different patterns can sign($A_1z$) 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.

- $z$ is $k$-dimensional.
- ReLU($A_1z$) is linear, within any constant region of sign($A_1z$).
- How many different patterns can sign($A_1z$) take?
- $k = 2$ version: how many regions can $n$ lines partition plane into?

\[ 1 + (1 + 2 + \ldots + n) = \frac{n^2 + n + 2}{2}. \]

\[ n \text{ half-spaces divide } \mathbb{R}^k \text{ into less than } n^k \text{ regions.} \]
Proof of Lemma

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

- $z$ is $k$-dimensional.
- ReLU($A_1z$) is linear, within any constant region of sign($A_1z$).
- How many different patterns can sign($A_1z$) take?
- $k = 2$ version: how many regions can $n$ lines partition plane into?

\[ 1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2} \]

\[ n \text{ half-spaces divide } \mathbb{R}^k \text{ into less than } n^k \text{ regions.} \]

- Therefore $d$-layer network has $n^{dk}$ regions.
Proof outline (Lipschitz networks)

- Need that any two images have significantly different measurements.
Proof outline (Lipschitz networks)

- Need that any two images have significantly different measurements.
- Regular compressed sensing: Restricted Eigenvalue Condition:

\[ \|Ax\|_2 \geq \gamma \|x\|_2 \]

for every \( O(k) \)-sparse vector \( x \).
Proof outline (Lipschitz networks)

- Need that any two images have significantly different measurements.
- Regular compressed sensing: *Restricted Eigenvalue Condition:*

\[ \|Ax\|_2 \geq \gamma \|x\|_2 \]

for every \( O(k) \)-sparse vector \( x \).

- Based on coincidence: difference between \( k \)-sparse vectors is \( 2k \)-sparse.
Proof outline (Lipschitz networks)

- Need that any two images have significantly different measurements.
- Regular compressed sensing: *Restricted Eigenvalue Condition*:

\[ \|Ax\|_2 \geq \gamma \|x\|_2 \]

for every \(O(k)-\)sparse vector \(x\).
- Based on coincidence: difference between \(k\)-sparse vectors is \(2k\)-sparse.
  - But difference between “natural” images is not natural.
Proof outline (Lipschitz networks)

- Need that any two images have significantly different measurements.
- Regular compressed sensing: \textit{Restricted Eigenvalue Condition}:

\[
\|Ax\|_2 \geq \gamma \|x\|_2
\]

for every \(O(k)\)-sparse vector \(x\).

- Based on coincidence: difference between \(k\)-sparse vectors is \(2k\)-sparse.
  - But difference between “natural” images is not natural.

\textbf{Set-Restricted Eigenvalue Condition}

\[
\|Ax_1 - Ax_2\|_2 \geq \gamma \|x_1 - x_2\|_2
\]

for all \(x_1, x_2 \in S\).
Proof outline (Lipschitz networks)

- Need that any two images have significantly different measurements.
- Regular compressed sensing: *Restricted Eigenvalue Condition:*

\[ \|Ax\|_2 \geq \gamma \|x\|_2 \]

for every $O(k)$-sparse vector $x$.
- Based on coincidence: difference between $k$-sparse vectors is $2k$-sparse.
  - But difference between “natural” images is not natural.

Set-Restricted Eigenvalue Condition

\[ \|Ax_1 - Ax_2\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta \]

for all $x_1, x_2 \in S$.

- Not true without extra slack.
Proof outline (Lipschitz networks)

Set-Restricted Eigenvalue Condition

\[ \|Ax_1 - Ax_2\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta \]

for all \( x_1, x_2 \in S \).
Proof outline (Lipschitz networks)

Set-Restricted Eigenvalue Condition

\[ \|Ax_1 - Ax_2\|_2 \geq \gamma\|x_1 - x_2\|_2 - \delta \]

for all \( x_1, x_2 \in S \).

- True for fixed \( x_1, x_2 \) with \( 1 - e^{-\Omega(m)} \) probability.
Proof outline (Lipschitz networks)

**Set-Restricted Eigenvalue Condition**

\[ \|Ax_1 - Ax_2\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta \]

for all \( x_1, x_2 \in S \).

- True for *fixed* \( x_1, x_2 \) with \( 1 - e^{-\Omega(m)} \) probability.
- Apply to \( \delta \)-cover of range(\( G \)).
Proof outline (Lipschitz networks)

Set-Restricted Eigenvalue Condition

\[ \|Ax_1 - Ax_2\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta \]

for all \( x_1, x_2 \in S \).

- True for fixed \( x_1, x_2 \) with \( 1 - e^{-\Omega(m)} \) probability.
- Apply to \( \delta \)-cover of range(\( G \)).
  - Comes from \( \delta/L \)-cover of domain(\( G \)).
Proof outline (Lipschitz networks)

**Set-Restricted Eigenvalue Condition**

\[ \|Ax_1 - Ax_2\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta \]

for all \( x_1, x_2 \in S \).

- True for fixed \( x_1, x_2 \) with \( 1 - e^{-\Omega(m)} \) probability.
- Apply to \( \delta \)-cover of \( \text{range}(G) \).
  - Comes from \( \delta/L \)-cover of \( \text{domain}(G) \).
  - Size \( (\frac{rL}{\delta})^k \): union bound works for \( m = O(k \log \frac{rL}{\delta}) \).
Proof outline (Lipschitz networks)

Set-Restricted Eigenvalue Condition

\[ \|A x_1 - A x_2\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta \]
for all \(x_1, x_2 \in S\).

- True for fixed \(x_1, x_2\) with \(1 - e^{-\Omega(m)}\) probability.
- Apply to \(\delta\)-cover of \(\text{range}(G)\).
  - Comes from \(\delta/L\)-cover of \(\text{domain}(G)\).
  - Size \((\frac{rl}{\delta})^k\): union bound works for \(m = O(k \log \frac{rl}{\delta})\).
- Hence
\[
\|\hat{x} - x\|_2 \leq C \min_{x' \in \text{range}(G)} \|x' - x\|_2 + \delta
\]
Generative models can bound information content as $O(kd \log n)$.
Generative models differentiable $\Rightarrow$ can optimize in practice.
Gaussian measurements ensure independent information.

\[
m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}}
\]
Generative models can bound information content as \( O(kd \log n) \).

\[
m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}}
\]

- Generative models can bound information content as \( O(kd \log n) \).
Summary (part 1)

\[ m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}} \]

- Generative models can bound information content as \( O(kd \log n) \).
- Generative models differentiable \( \implies \) can optimize in practice.
Summary (part 1)

\[ m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}} \]

- Generative models can bound information content as \( O(kd \log n) \).
- Generative models differentiable \( \implies \) can optimize in practice.
- Gaussian measurements ensure independent information.
Generative models can bound information content as $O(kd \log n)$. Generative models differentiable $\implies$ can optimize in practice. Gaussian measurements ensure independent information.

- $O(1)$ approximation factor
Generative models can bound information content as $O(kd \log n)$.

Generative models differentiable $\implies$ can optimize in practice.

Gaussian measurements ensure independent information.

- $O(1)$ approximation factor $\iff$ $O(1)$ SNR
Summary (part 1)

\[ m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}} \]

- Generative models can bound information content as \( O(kd \log n) \).
- Generative models differentiable \( \Rightarrow \) can optimize in practice.
- Gaussian measurements ensure independent information.
  - \( O(1) \) approximation factor \( \iff \) \( O(1) \) SNR \( \iff \) \( O(1) \) bits each
Generative models can bound information content as $O(kd \log n)$.

- Generative models differentiable $\implies$ can optimize in practice.
- Gaussian measurements ensure independent information.
  - $O(1)$ approximation factor $\iff$ $O(1)$ SNR $\iff$ $O(1)$ bits each
- With random weights (i.e., before training) can prove more:
Generative models can bound information content as $O(kd \log n)$.

Generative models differentiable $\implies$ can optimize in practice.

Gaussian measurements ensure independent information.

- $O(1)$ approximation factor $\iff$ $O(1)$ SNR $\iff$ $O(1)$ bits each

With \textit{random} weights (i.e., before training) can prove more:

- The optimization has no local minima [Hand-Voroninski]
Generative models can bound information content as $O(kd \log n)$. Generative models differentiable $\implies$ can optimize in practice. Gaussian measurements ensure independent information.
  
  - $O(1)$ approximation factor $\iff$ $O(1)$ SNR $\iff$ $O(1)$ bits each

  With *random* weights (i.e., before training) can prove more:
  
  - The optimization has no local minima [Hand-Voroninski]
  - $L = O(1)$ not $n^d$ so $m = O(k \log n)$, if $k \ll n/d$. 

\[
m > \frac{\text{(information in image)}}{\text{(new info. per measurement)}}
\]
Extensions

- **Inpainting:**

![Inpainting Image]
Extensions

- Inpainting:
Extensions

- Inpainting:

  - $A$ is diagonal, zeros and ones.
Extensions

- **Inpainting:**

  - $A$ is diagonal, zeros and ones.

- **Deblurring:**

  ![Image of a blurred face with a block and the same face with the block removed]
Extensions

- **Inpainting:**

  - $A$ is diagonal, zeros and ones.

- **Deblurring:**
Talk Outline

1. Using generative models for compressed sensing

2. Learning generative models from noisy data
GAN Architecture

Z
GAN Architecture

Z → G

Generated image

Real image

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
GAN Architecture

G

Generated image

Z

Real image
GAN Architecture

Generated image

Real image

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
GAN Architecture

- **Z**
- **G**
- **D**
- Generated image
- Real image

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
GAN Architecture

![Diagram of GAN Architecture]

- **Z** fed into **G** (Generator) to produce a **Generated image**.
- Real image fed into **D** (Discriminator) to determine if it's **Real?**
GAN Architecture

Generator $G$ wants to fool the discriminator $D$. 

Empirically works for $G$, $D$ being convolutional neural nets.
Generator $G$ wants to fool the discriminator $D$.

If $G$, $D$ infinitely powerful: only pure Nash equilibrium when $G(Z)$ equals true distribution.
GAN Architecture

- Generator $G$ wants to fool the discriminator $D$.
- If $G, D$ infinitely powerful: only pure Nash equilibrium when $G(Z)$ equals true distribution.
- Empirically works for $G, D$ being convolutional neural nets.
One use of GANs: improving measurements

- A generative model tells you the *structure* of images.
One use of GANs: improving measurements

- A generative model tells you the *structure* of images.
- Can use this to improve estimates of future images:
One use of GANs: improving measurements

- A generative model tells you the structure of images.
- Can use this to improve estimates of future images:
  - Bora et al. 2017: compressed sensing
One use of GANs: improving measurements

- A generative model tells you the *structure* of images.
- Can use this to improve estimates of future images:
  - Bora et al. 2017: compressed sensing
  - Schawinski et al. 2017: denoising astronomical images
One use of GANs: improving measurements

- A generative model tells you the *structure* of images.
- Can use this to improve estimates of future images:
  - Bora et al. 2017: compressed sensing
  - Schawinski et al. 2017: denoising astronomical images
- Can hope for higher-quality images than your measurement system would otherwise get.
One use of GANs: improving measurements

- A generative model tells you the \textit{structure} of images.
- Can use this to improve estimates of future images:
  - Bora et al. 2017: compressed sensing
  - Schawinski et al. 2017: denoising astronomical images
- Can hope for higher-quality images than your measurement system would otherwise get.

Problem

If measuring images is hard/noisy, how do you collect a good data set?
One use of GANs: improving measurements

- A generative model tells you the *structure* of images.
- Can use this to improve estimates of future images:
  - Bora et al. 2017: compressed sensing
  - Schawinski et al. 2017: denoising astronomical images
- Can hope for higher-quality images than your measurement system would otherwise get.

Problem

If measuring images is hard/noisy, how do you collect a good data set?

- Answer: You can’t, easily.
One use of GANs: improving measurements

- A generative model tells you the *structure* of images.
- Can use this to improve estimates of future images:
  - Bora et al. 2017: compressed sensing
  - Schawinski et al. 2017: denoising astronomical images
- Can hope for higher-quality images than your measurement system would otherwise get.

Problem

If measuring images is hard/noisy, how do you collect a good data set?

- Answer: You can’t, easily.

Goal of this work

Can we learn a GAN from incomplete, noisy measurements of the desired images?
GAN training

Discriminator must distinguish real measurements from simulated measurements of fake images. Can try this for any measurement process $f$ you understand. Compatible with any GAN generator architecture.
GAN training

Ambient

GAN training

Z

G

Generated image

D

Real?

Discriminator must distinguish real measurements from simulated measurements of fake images.

Can try this for any measurement process you understand.

Compatible with any GAN generator architecture.

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
Ambient GAN training

Discriminator must distinguish real measurements from simulated measurements of fake images. Can try this for any measurement process you understand. Compatible with any GAN generator architecture.
AmbientGAN training

Discriminator must distinguish real measurements from simulated measurements of fake images. Can try this for any measurement process $f$ you understand. Compatible with any GAN generator architecture.

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
AmbientGAN training

- Discriminator must distinguish *real measurements* from *simulated measurements of fake images*.
AmbientGAN training

Discriminator must distinguish *real measurements* from *simulated measurements of fake images*

Can try this for any measurement process $f$ you understand.
AmbientGAN training

- Discriminator must distinguish *real measurements* from *simulated measurements of fake images*
- Can try this for any measurement process $f$ you understand.
- Compatible with any GAN generator architecture.
Measurement: Gaussian blur + Gaussian noise

- Measured

Gaussian blur + additive Gaussian noise attenuates high-frequency components.
Measurement: Gaussian blur + Gaussian noise

- Gaussian blur + additive Gaussian noise attenuates high-frequency components.
- Wiener baseline: deconvolve before learning GAN.

AmbientGAN better preserves high-frequency components.

▶ Uses DCGAN [Radford et al., 2015] for generator/discriminator.

Theorem: in the limit of dataset size and $G, D$ capacity $\to \infty$, Nash equilibrium of AmbientGAN is the true distribution.
Measurement: Gaussian blur + Gaussian noise

- Gaussian blur + additive Gaussian noise attenuates high-frequency components.
- Wiener baseline: deconvolve before learning GAN.
- AmbientGAN better preserves high-frequency components.
Measurement: Gaussian blur + Gaussian noise

- Gaussian blur + additive Gaussian noise attenuates high-frequency components.
- Wiener baseline: deconvolve before learning GAN.
- AmbientGAN better preserves high-frequency components.
  - Uses DCGAN [Radford et al., 2015] for generator/discriminator.
**Measurement:** Gaussian blur + Gaussian noise

- **Measured**
- **Wiener Baseline**
- **AmbientGAN**

- Gaussian blur + additive Gaussian noise attenuates high-frequency components.
- Wiener baseline: deconvolve before learning GAN.
- AmbientGAN better preserves high-frequency components.
  - Uses DCGAN [Radford et al., 2015] for generator/discriminator.
- Theorem: in the limit of dataset size and $G, D$ capacity $\to \infty$, Nash equilibrium of AmbientGAN is the true distribution.
Measurement: Obscured Square

- Obscure a random square containing 25% of the image.
Measurement: Obscured Square

- Obscure a random square containing 25% of the image.
- Inpainting followed by GAN training reproduces inpainting artifacts.
Measurement: Obscured Square

- Obscure a random square containing 25% of the image.
- Inpainting followed by GAN training reproduces inpainting artifacts.
- AmbientGAN gives much smaller artifacts.
Measurement: Obscured Square

- Obscure a random square containing 25% of the image.
- Inpainting followed by GAN training reproduces inpainting artifacts.
- AmbientGAN gives much smaller artifacts.
- No theorem: doesn’t know that eyes should have the same color.
Measurement: Limited View

- Motivation: learn the distribution of *panoramas* from the distribution of *photos*?
Measurement: Limited View

- Motivation: learn the distribution of *panoramas* from the distribution of *photos*?
  
  Measured

- Reveal a random square containing 25% of the image.
Measurement: Limited View

- Motivation: learn the distribution of *panoramas* from the distribution of *photos*?

- Measured
- AmbientGAN

- Reveal a random square containing 25% of the image.
- AmbientGAN still recovers faces.
Measurement: Dropout

Measured

- Drop each pixel independently with probability $p = 95\%$. 

Measurement: Dropout

- Drop each pixel independently with probability $p = 95\%$.
- Simple baseline does terribly.
Drop each pixel independently with probability $p = 95\%$.

Simple baseline does terribly.

AmbientGAN can still learn faces.
Measurement: Dropout

- Drop each pixel independently with probability $p = 95\%$.
- Simple baseline does terribly.
- AmbientGAN can still learn faces.
- Theorem: in the limit of dataset size and $G$, $D$ capacity $\to \infty$, Nash equilibrium of AmbientGAN is the true distribution.
1D Projections

- So far, measurements have all looked like images themselves.
So far, measurements have all looked like images themselves. What if we turn a 2D image into a 1D image?
1D Projections

- So far, measurements have all looked like images themselves.
- What if we turn a 2D image into a 1D image?
- Motivation: X-ray scans project 3D into 2D.
So far, measurements have all looked like images themselves.
What if we turn a 2D image into a 1D image?
Motivation: X-ray scans project 3D into 2D.
Face reconstruction is crude, but MNIST digits work decently:
1D Projections

- So far, measurements have all looked like images themselves.
- What if we turn a 2D image into a 1D image?
- Motivation: X-ray scans project 3D into 2D.
- Face reconstruction is crude, but MNIST digits work decently:
Quality as a function of dropout probability

- Quantify results using Inception score, for all its flaws.
Quality as a function of dropout probability

- Quantify results using Inception score, for all its flaws.
- Requires supervised learning task, so switch to CIFAR-10 and the Wasserstein GAN architecture of [Gulrajani et al. 2017].
Quality as a function of dropout probability

- Quantify results using Inception score, for all its flaws.
- Requires supervised learning task, so switch to CIFAR-10 and the Wasserstein GAN architecture of [Gulrajani et al. 2017].

![Graph showing Inception score as a function of block probability](image-url)
Robustness to model mismatch

- We assume we know the true measurement process.

![Inception score](image-url)
Robustness to model mismatch

- We assume we know the true measurement process.
- What happens if we get it wrong?
Robustness to model mismatch

- We assume we know the true measurement process.
- What happens if we get it wrong?
- On MNIST:

![Graph showing Inception score vs. Block probability (p) for AmbientGAN (ours)]
Compressed sensing

- Compressed sensing: learn an image $x$ from low-dimensional linear projection $Ax$. 

Theorem about unique Nash equilibrium in the limit.

Ashish Bora, Ajil Jalal, **Eric Price**, Alex Dimakis (UT Austin)
Compressed sensing

- Compressed sensing: learn an image $x$ from low-dimensional linear projection $Ax$.
Compressed sensing

- Compressed sensing: learn an image $x$ from low-dimensional linear projection $Ax$.
- AmbientGAN can learn the generative model from a dataset of projections $\{(A_i, A_ix_i)\}$.
Compressed sensing

- Compressed sensing: learn an image $x$ from low-dimensional linear projection $Ax$.
- AmbientGAN can learn the generative model from a dataset of projections $\{(A_i, A_ix_i)\}$.
- Beats standard sparse recovery (e.g. Lasso).

---

Theorem about unique Nash equilibrium in the limit.

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)
Compressed sensing

- Compressed sensing: learn an image $x$ from low-dimensional linear projection $Ax$.
- AmbientGAN can learn the generative model from a dataset of projections $\{(A_i, A_ix_i)\}$.
- Beats standard sparse recovery (e.g. Lasso).

Theorem about unique Nash equilibrium in the limit.
Summary

Plug the measurement process into the GAN architecture of your choice. The generator learns the pre-measurement ground truth better than if you denoise before training. This could let us learn distributions we have no data for. Read the paper ("AmbientGAN") for lots more experiments.

Ashish Bora, Ajil Jalal, Eric Price, Alex Dimakis (UT Austin)

Compressed Sensing and Generative Models
Summary

- Plug the measurement process into the GAN architecture of your choice.
Summary

- Plug the measurement process into the GAN architecture of your choice.
- The generator learns the pre-measurement ground truth better than if you denoise before training.

![Diagram of GAN architecture](image)

Ashish Bora, Ajil Jalal, **Eric Price**, Alex Dimakis (UT Austin)
Summary

- Plug the measurement process into the GAN architecture of your choice.
- The generator learns the pre-measurement ground truth better than if you denoise before training.
- Could let us learn distributions we have no data for.
Plug the measurement process into the GAN architecture of your choice.

The generator learns the pre-measurement ground truth better than if you denoise before training.

Could let us learn distributions we have no data for.

Read the paper ("AmbientGAN") for lots more experiments.
Conclusion and open questions

Main results:

- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.
- Finite-sample theorems for learning the generative model?
- Take Gaussian blur plus Gaussian noise.
- Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
- With $N$ data points, can we learn $\log N$ standard deviations?
- Better upper bound on complexity of generative models?
- Lipschitz parameter at initialization is much smaller than $n^{d}$...
- ...but we don’t actually expect it to be small after training.
- Can the reconstruction incorporate density over the manifold?
- Computational problem: pseudodeterminant of Jacobian matrix.
- Speed-up with linear sketching?
- More uses of differentiable compression?
Conclusion and open questions

Main results:

- Can use lossy measurements to learn a generative model of the underlying distribution.

- Take Gaussian blur plus Gaussian noise.

- Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.

- With $N$ data points, can we learn $\log N$ standard deviations?

- Better upper bound on complexity of generative models?

- Lipschitz parameter at initialization is much smaller than $n^{d/2}$...

- ...but we don’t actually expect it to be small after training.

- Can the reconstruction incorporate density over the manifold?

- Computational problem: pseudodeterminant of Jacobian matrix.

- Speed-up with linear sketching?

- More uses of differentiable compression?
Conclusion and open questions

Main results:

- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?

Take Gaussian blur plus Gaussian noise.

Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.

With $N$ data points, can we learn $\log N$ standard deviations?

Better upper bound on complexity of generative models?

- Lipschitz parameter at initialization is much smaller than $n/d$...
- ...but we don’t actually expect it to be small after training.

Can the reconstruction incorporate density over the manifold?

Computational problem: pseudodeterminant of Jacobian matrix.

Speed-up with linear sketching?

More uses of differentiable compression?
Conclusion and open questions

Main results:

- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.
- Finite-sample theorems for learning the generative model?
Conclusion and open questions

Main results:
- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?
- Take Gaussian blur plus Gaussian noise.
Conclusion and open questions

Main results:

- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?

- Take Gaussian blur plus Gaussian noise.
- Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.

- Computational problem: pseudodeterminant of Jacobian matrix.
- Speed-up with linear sketching?

- More uses of differentiable compression?
Conclusion and open questions

Main results:
- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?
- Take Gaussian blur plus Gaussian noise.
- Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
- With $N$ data points, can we learn $\log N$ standard deviations?
Conclusion and open questions

Main results:
- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?
- Take Gaussian blur plus Gaussian noise.
- Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
  - With $N$ data points, can we learn $\log N$ standard deviations?

Better upper bound on complexity of generative models?
Conclusion and open questions

- **Main results:**
  - Can use lossy measurements to learn a generative model of the underlying distribution.
  - Can use a generative model to reconstruct from lossy measurements.
- **Finite-sample theorems for learning the generative model?**
  - Take Gaussian blur plus Gaussian noise.
  - Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
  - With $N$ data points, can we learn $\log N$ standard deviations?
- **Better upper bound on complexity of generative models?**
  - Lipschitz parameter at initialization is much smaller than $n^d$...
Conclusion and open questions

- Main results:
  - Can use lossy measurements to learn a generative model of the underlying distribution.
  - Can use a generative model to reconstruct from lossy measurements.

- Finite-sample theorems for learning the generative model?
  - Take Gaussian blur plus Gaussian noise.
  - Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
  - With $N$ data points, can we learn $\log N$ standard deviations?

- Better upper bound on complexity of generative models?
  - Lipschitz parameter at initialization is much smaller than $n^d$...
  - ...but we don’t actually expect it to be small after training.
Conclusion and open questions

- **Main results:**
  - Can use lossy measurements to learn a generative model of the underlying distribution.
  - Can use a generative model to reconstruct from lossy measurements.
- **Finite-sample theorems for learning the generative model?**
  - Take Gaussian blur plus Gaussian noise.
  - Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
  - With $N$ data points, can we learn $\log N$ standard deviations?
- **Better upper bound on complexity of generative models?**
  - Lipschitz parameter at initialization is much smaller than $n^d$...
  - ...but we don’t actually expect it to be small after training.
- **Can the reconstruction incorporate density over the manifold?**
Conclusion and open questions

Main results:
- Can use lossy measurements to learn a generative model of the underlying distribution.
- Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?
- Take Gaussian blur plus Gaussian noise.
- Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
- With $N$ data points, can we learn log $N$ standard deviations?

Better upper bound on complexity of generative models?
- Lipschitz parameter at initialization is much smaller than $n^d$...
- ...but we don’t actually expect it to be small after training.

Can the reconstruction incorporate density over the manifold?
- Computational problem: pseudodeterminant of Jacobian matrix.
Conclusion and open questions

- **Main results:**
  - Can use lossy measurements to learn a generative model of the underlying distribution.
  - Can use a generative model to reconstruct from lossy measurements.

- **Finite-sample theorems for learning the generative model?**
  - Take Gaussian blur plus Gaussian noise.
  - Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
  - With $N$ data points, can we learn $\log N$ standard deviations?

- **Better upper bound on complexity of generative models?**
  - Lipschitz parameter at initialization is much smaller than $n^d$...
  - ...but we don’t actually expect it to be small after training.

- **Can the reconstruction incorporate density over the manifold?**
  - Computational problem: pseudodeterminant of Jacobian matrix.
  - Speed-up with linear sketching?
Conclusion and open questions

Main results:

▶ Can use lossy measurements to learn a generative model of the underlying distribution.
▶ Can use a generative model to reconstruct from lossy measurements.

Finite-sample theorems for learning the generative model?

▶ Take Gaussian blur plus Gaussian noise.
▶ Wiener filter before GAN: lose frequencies beyond $O(1)$ standard deviations.
▶ With $N$ data points, can we learn $\log N$ standard deviations?

Better upper bound on complexity of generative models?

▶ Lipschitz parameter at initialization is much smaller than $n^d$...
▶ ...but we don’t actually expect it to be small after training.

Can the reconstruction incorporate density over the manifold?

▶ Computational problem: pseudodeterminant of Jacobian matrix.
▶ Speed-up with linear sketching?

More uses of differentiable compression?