Compressed Sensing and Generative Models

Ashish Bora Ajil Jalal **Eric Price** Alex Dimakis

UT Austin

Talk Outline

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

Talk Outline

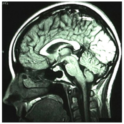
1. Compressed sensing
2. Using generative models for compressed sensing
3. Learning generative models from noisy data

Compressed Sensing

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

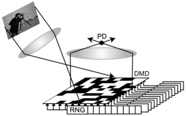
Compressed Sensing

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



Medical Imaging

Astronomy



Single-Pixel Camera

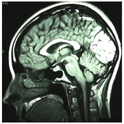
Oil Exploration

Genetic Testing

Streaming Algorithms

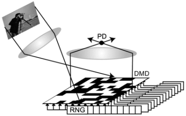
Compressed Sensing

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



Medical Imaging

Astronomy



Single-Pixel Camera

Oil Exploration

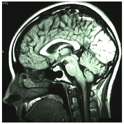
Genetic Testing

Streaming Algorithms

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

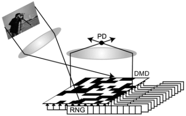
Compressed Sensing

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



Medical Imaging

Astronomy



Single-Pixel Camera

Oil Exploration

Genetic Testing

Streaming Algorithms

- *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?

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*.

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:

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

Compressed Sensing

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

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

Compressed Sensing

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

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

- 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?

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

- 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

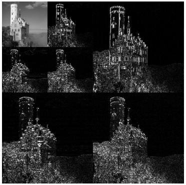
- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the "most compressible" image that fits measurements.
- How should we formalize that an image is "compressible"?
- Short JPEG compression

Compressed Sensing

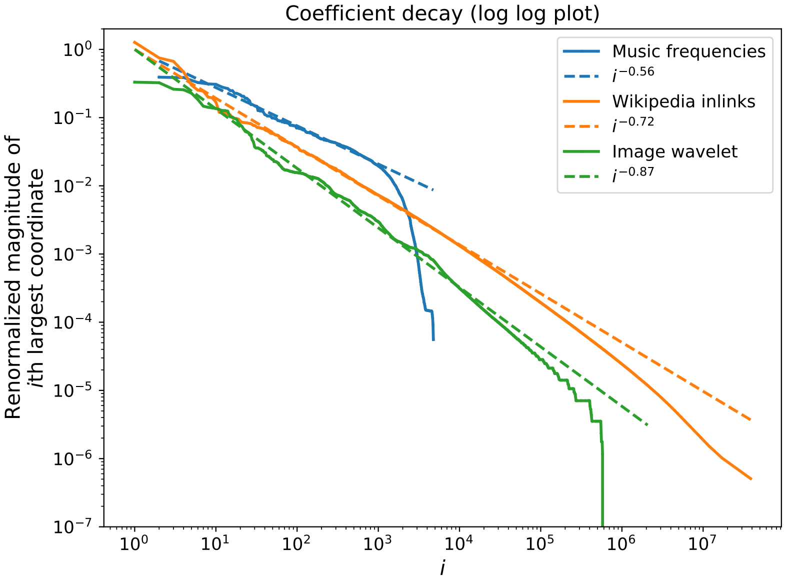
- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the "most compressible" image that fits measurements.
- How should we formalize that an image is "compressible"?
- ~~Short JPEG compression~~
  - Intractible to compute.

Compressed Sensing

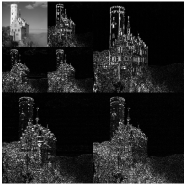
- Want to estimate $x \in \mathbb{R}^n$ from $m \ll n$ linear measurements.
- Suggestion: the "most compressible" image that fits measurements.
- How should we formalize that an image is "compressible"?
- ~~Short JPEG compression~~
  - Intractible to compute.
- Standard compressed sensing: *sparsity*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

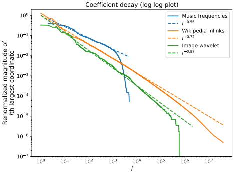


Approximate sparsity is common



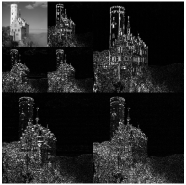
Sample complexity of sparse recovery

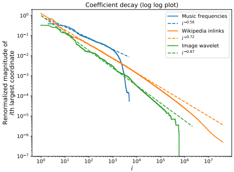




$$m \approx \frac{\text{(information in image)}}{\text{(new info. per measurement)}}$$

Sample complexity of sparse recovery

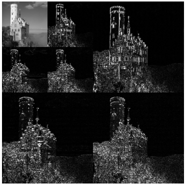


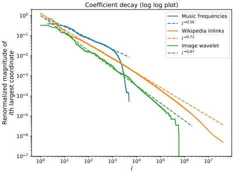


$$m \approx \frac{\text{(information in image)}}{\text{(new info. per measurement)}}$$

- If 99% of energy in largest k coordinates...

Sample complexity of sparse recovery

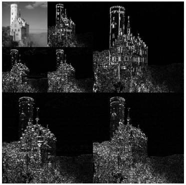


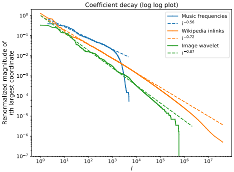


$$m \approx \frac{\text{(information in image)}}{\text{(new info. per measurement)}}$$

- If 99% of energy in largest k coordinates...
- Information in image is $\approx \log \binom{n}{k} \approx k \log n$

Sample complexity of sparse recovery





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

- If 99% of energy in largest k coordinates...
- Information in image is $\approx \log \binom{n}{k} \approx k \log n$
- New info. per measurement is hopefully $\approx \log 100 = \Theta(1)$

Compressed Sensing Formalism

"Compressible" = "sparse"

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

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 $y = Ax$.

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 $y = Ax$.
- Goal: $\widehat{x}$ with

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

with high probability.

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 $y = Ax$.
- Goal: $\widehat{x}$ with

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

with high probability.

  - Reconstruction accuracy proportional to model accuracy.

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 $y = Ax$.
- Goal: $\widehat{x}$ with

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

with high probability.
  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candès-Romberg-Tao 2006]

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 $y = Ax$.
- Goal: $\widehat{x}$ with

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

  with high probability.
  - Reconstruction accuracy proportional to model accuracy.
- Theorem [Candès-Romberg-Tao 2006]
  - $m = \Theta(k \log(n/k))$ suffices for (1).

Compressed Sensing Formalism

"Compressible" = "sparse"

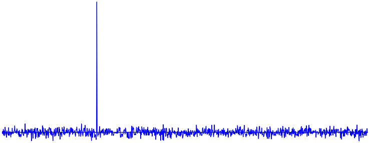
- Want to estimate x from $y = Ax + \eta$, for $A \in \mathbb{R}^{m \times n}$.
  - For this talk: ignore η, so $y = Ax$.
- Goal: $\widehat{x}$ with

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

  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 $\widehat{x}$ can be found efficiently with, e.g., the LASSO.

Lower bound: $k = 1$

- Hard case: x is random e_i plus Gaussian noise w with $\|w\|_2 \approx 1$.



- Robust recovery must locate i.
- Observations $\langle v, x \rangle = v_i + \langle v, w \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, for $z \sim N(0, 1)$.

1-sparse lower bound

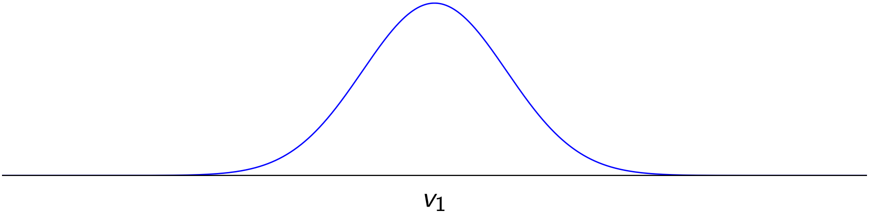
P-Woodruff '11

- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$

1-sparse lower bound

P-Woodruff '11

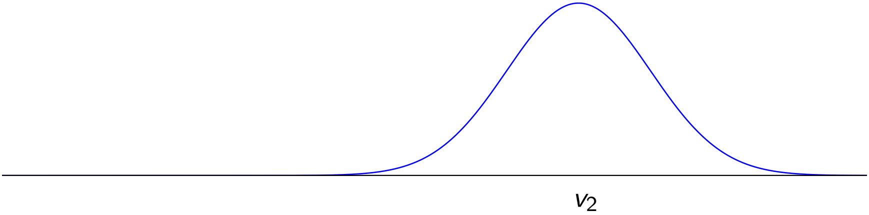
- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



1-sparse lower bound

P-Woodruff '11

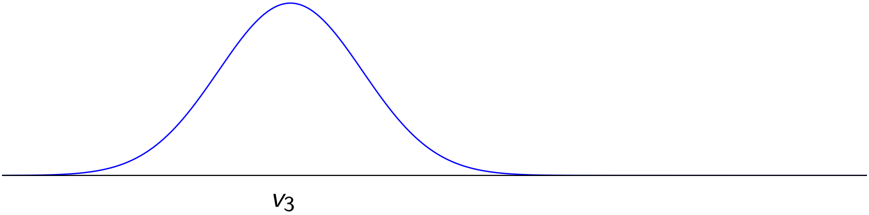
- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



1-sparse lower bound

P-Woodruff '11

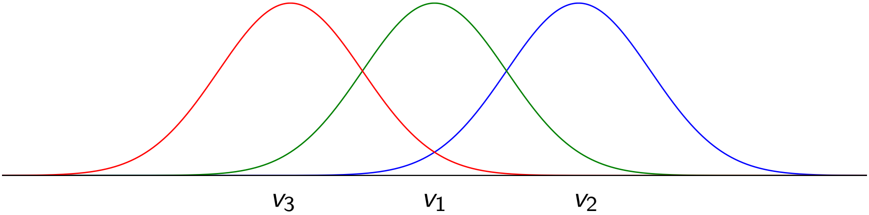
- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



1-sparse lower bound

P-Woodruff '11

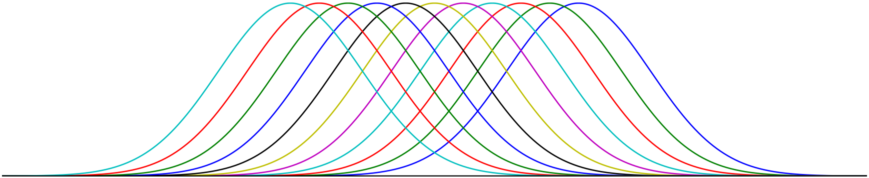
- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



1-sparse lower bound

P-Woodruff '11

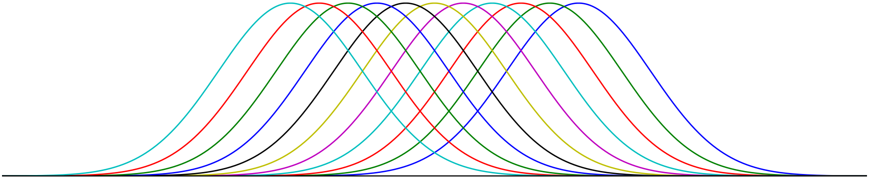
- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



1-sparse lower bound

P-Woodruff '11

- Observe $\langle v, x\rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon 1948: AWGN channel capacity is

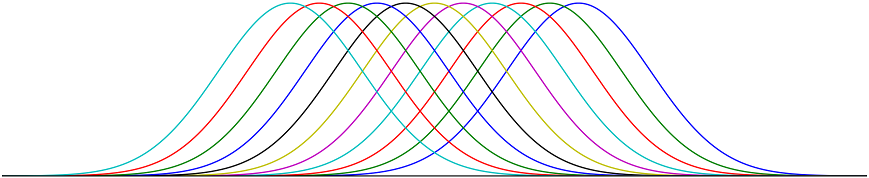
$$I(i, \langle v, x\rangle) \leq \frac{1}{2}\log(1 + \text{SNR})$$

where *SNR* denotes the "signal-to-noise ratio,"

1-sparse lower bound

P-Woodruff '11

- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon 1948: AWGN channel capacity is

$$I(i, \langle v, x \rangle) \leq \frac{1}{2} \log(1 + \text{SNR})$$

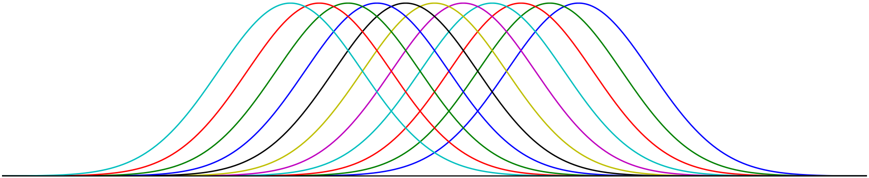
where *SNR* denotes the "signal-to-noise ratio,"

$$SNR = \frac{\mathbb{E}[\text{signal}^2]}{\mathbb{E}[\text{noise}^2]} \eqsim \frac{\mathbb{E}[v_i^2]}{\|v\|_2^2/n}$$

1-sparse lower bound

P-Woodruff '11

- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon 1948: AWGN channel capacity is

$$I(i, \langle v, x \rangle) \leq \frac{1}{2} \log(1 + \text{SNR})$$

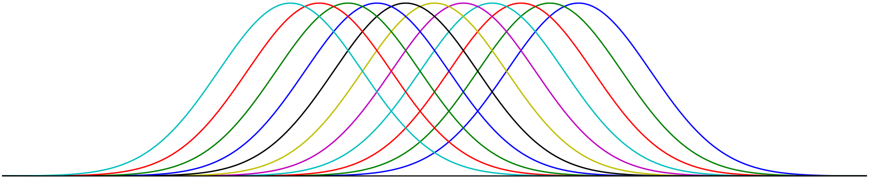
where *SNR* denotes the "signal-to-noise ratio,"

$$SNR = \frac{\mathbb{E}[\text{signal}^2]}{\mathbb{E}[\text{noise}^2]} \eqsim \frac{\mathbb{E}[v_i^2]}{\|v\|_2^2 / n} = 1$$

1-sparse lower bound

P-Woodruff '11

- Observe $\langle v, x \rangle = v_i + \frac{\|v\|_2}{\sqrt{n}} z$, where $z \sim N(0, \Theta(1))$



- Shannon 1948: AWGN channel capacity is

$$I(i, \langle v, x \rangle) \leq \frac{1}{2} \log(1 + \mathrm{SNR})$$

where *SNR* denotes the "signal-to-noise ratio,"

$$SNR = \frac{\mathbb{E}[\text{signal}^2]}{\mathbb{E}[\text{noise}^2]} \eqsim \frac{\mathbb{E}[v_i^2]}{\|v\|_2^2 / n} = 1$$

- (info. per measurement) $= O(1)$

Lower bound

P-Woodruff '11

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

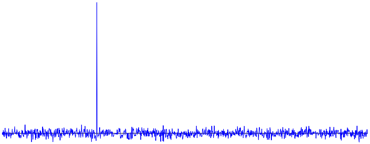
- (info. per measurement) $= O(1)$

Lower bound

P-Woodruff '11

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

- (info. per measurement) $= O(1)$
- $k = 1$: (information in image) $= \log n \implies m = \Omega(\log n)$

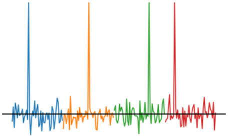


Lower bound

P-Woodruff '11

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

- (info. per measurement) $= O(1)$
- $k = 1$: (information in image) $= \log n \implies m = \Omega(\log n)$



- General k: $m = \Omega(\log \binom{n}{k}) = \Omega(k \log(n/k))$.

Talk Outline

Alternatives to sparsity?

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

Alternatives to sparsity?

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

- MRI images are sparse in the wavelet basis.

Alternatives to sparsity?

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

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

Alternatives to sparsity?

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

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

Alternatives to sparsity?

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

- 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.

Alternatives to sparsity?

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

- 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 2019?

Alternatives to sparsity?

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

- 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 2019?
  - Deep convolutional neural networks.

Alternatives to sparsity?

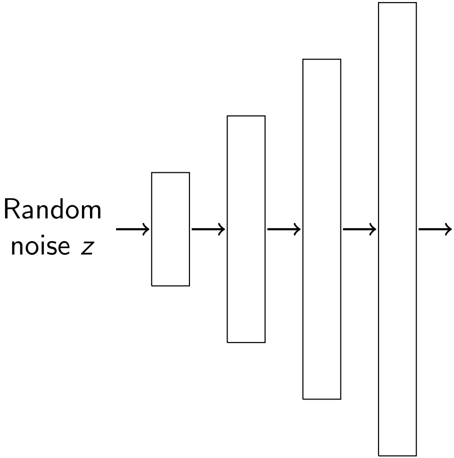
$$m \approx \frac{\text{(information in image)}}{\text{(new info. per measurement)}}$$

- 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 2019?
  - Deep convolutional neural networks.
  - In particular: *generative models*.

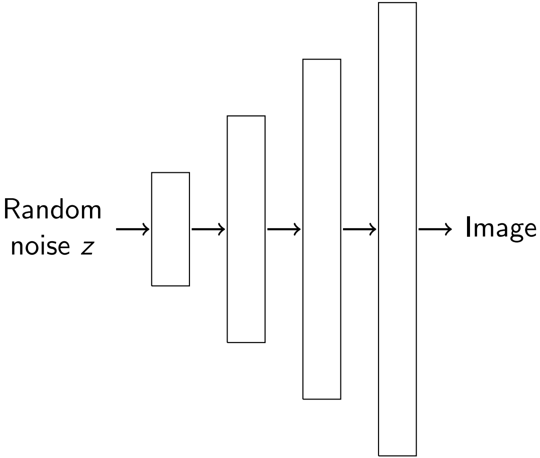
Generative Models

Random
noise z

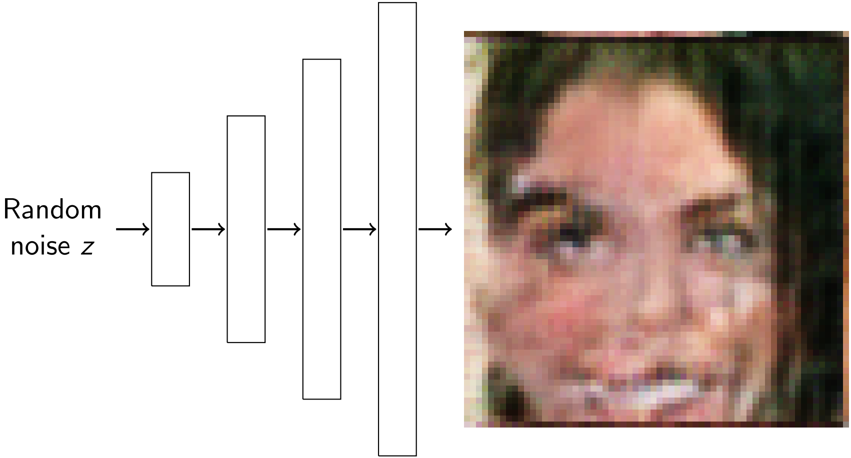
Generative Models



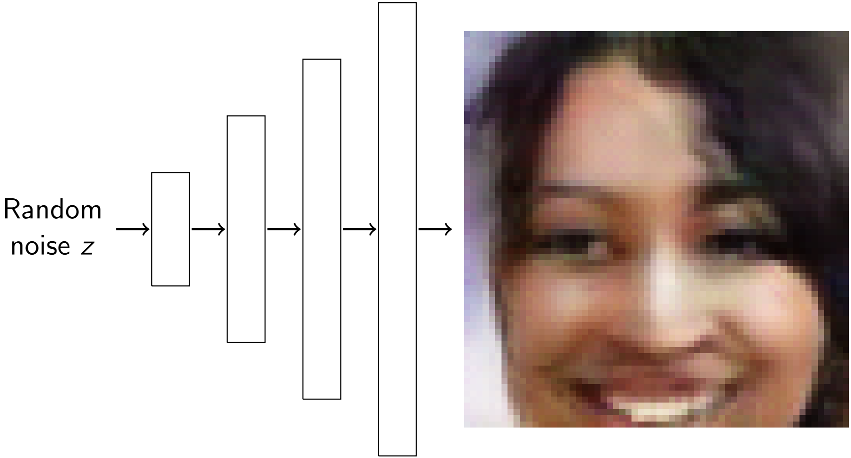
Generative Models



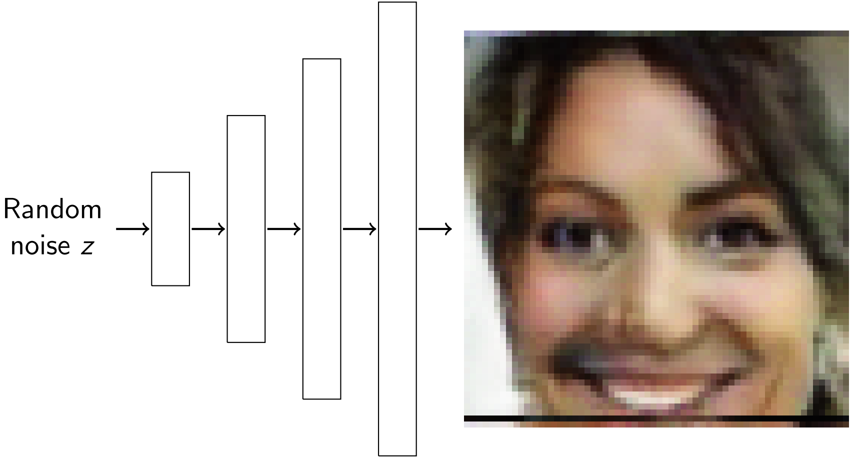
Training Generative Models



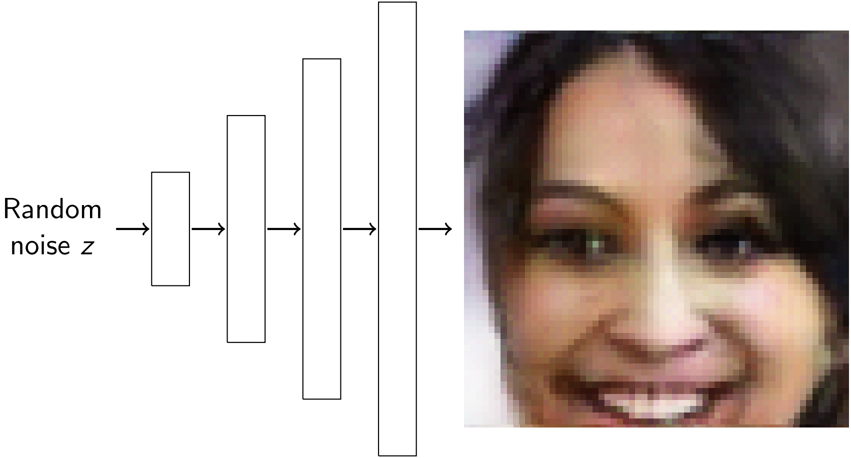
Training Generative Models



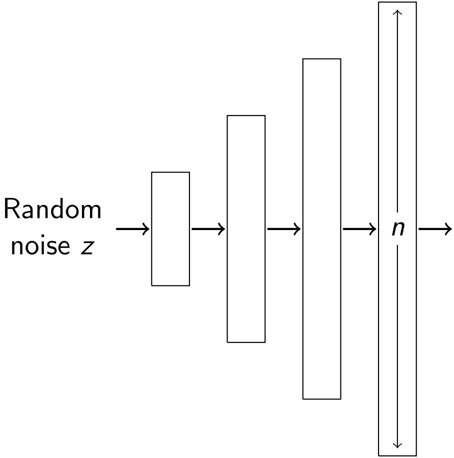
Training Generative Models



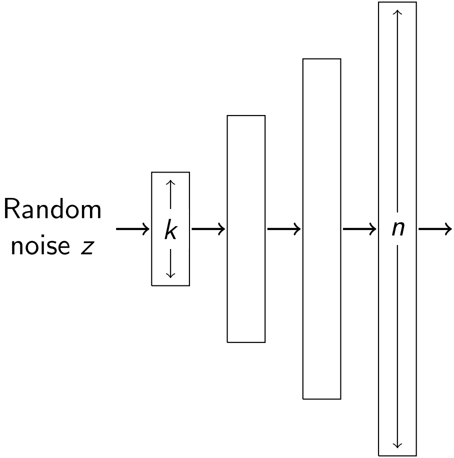
Training Generative Models



Training Generative Models



Training 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 $G(z) \sim \mathcal{D}$.
- 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 $G(z) \sim \mathcal{D}$.
- Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]:

Faces

Karras et al., 2018

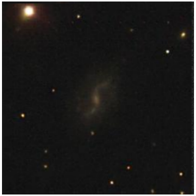
Generative Models

- 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]:

Faces

Karras et al., 2018

Astronomy



Schawinski et al., 2017

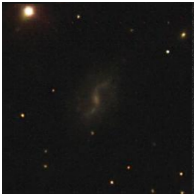
Generative Models

- 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]:

Faces

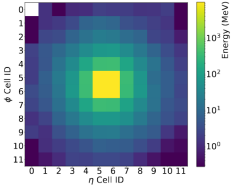
Karras et al., 2018

Astronomy



Schawinski et al., 2017

Particle Physics



Paganini et al., 2017

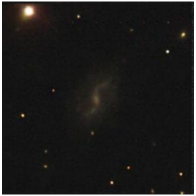
Generative Models

- 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]:

Faces

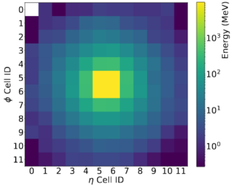
Karras et al., 2018

Astronomy



Schawinski et al., 2017

Particle Physics



Paganini et al., 2017

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

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]:

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].

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 $A \in \mathbb{R}^{m \times n}$.
- Goal: $\widehat{x}$ with

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

Our Results

"Compressible" = "near range(G)"

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

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

Our Results

"Compressible" = "near range(G)"

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

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

- 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: $\widehat{x}$ with

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

- 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)"

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

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

- 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)”

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

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

- 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: $\widehat{x}$ with

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

- 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: $\widehat{x}$ with

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

- 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: $\widehat{x}$ with

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

- 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: $\widehat{x}$ with

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

- $m = O(kd \log n)$ 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: $\widehat{x}$ with

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

- $m = O(kd \log n)$ 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: $\widehat{x}$ with

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

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

Our Results (II)

"Compressible" = "near range(G)"

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

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

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

Our Results (II)

"Compressible" = "near range(G)"

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

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

- $m = O(kd \log n)$ suffices for d-layer G.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - k here can be much smaller
- Find $\widehat{x} = G(\widehat{z})$ by gradient descent on $\|y - AG(\widehat{z})\|_2$.
  - 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: $\widehat{x}$ with

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

- $m = O(kd \log n)$ suffices for d-layer G.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - k here can be much smaller
- Find $\widehat{x} = G(\widehat{z})$ by gradient descent on $\|y - AG(\widehat{z})\|_2$.
  - 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: $\widehat{x}$ with

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

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

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

- $m = O(kd \log n)$ suffices for d-layer G.
  - Compared to $O(k \log n)$ for sparsity-based methods.
  - k here can be much smaller
- Find $\widehat{x} = G(\widehat{z})$ by gradient descent on $\|y - AG(\widehat{z})\|_2$.
  - Just like for training, no proof this converges
  - Approximate solution approximately gives (3)
  - Can check that $\|\widehat{x} - x\|_2$ is small.
  - In practice, optimization error seems negligible.

Related Work

- Projections on manifolds (Baraniuk-Wakin '09, Eftekhari-Wakin '15)

Related Work

- Projections on manifolds (Baraniuk-Wakin '09, Eftekhari-Wakin '15)
  - Conditions on manifold for which recovery is possible.

Related Work

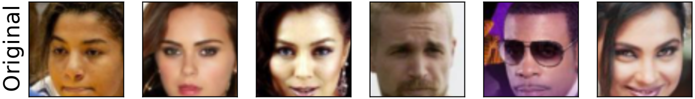
- Projections on manifolds (Baraniuk-Wakin '09, Eftekhari-Wakin '15)
  - Conditions on manifold for which recovery is possible.
- Deep network models (Mousavi-Dasarathy-Baraniuk '17, Chang et al '17)

Related Work

- 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.

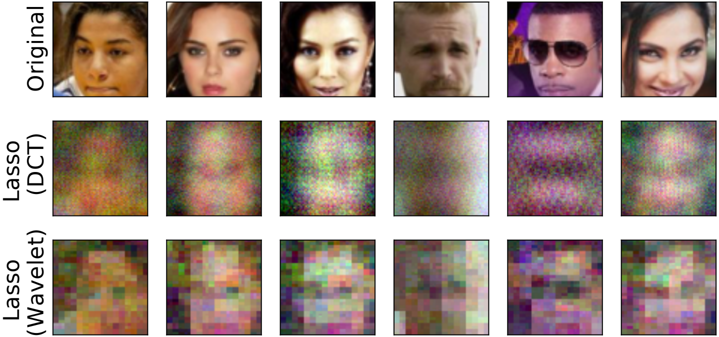
Experimental Results

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



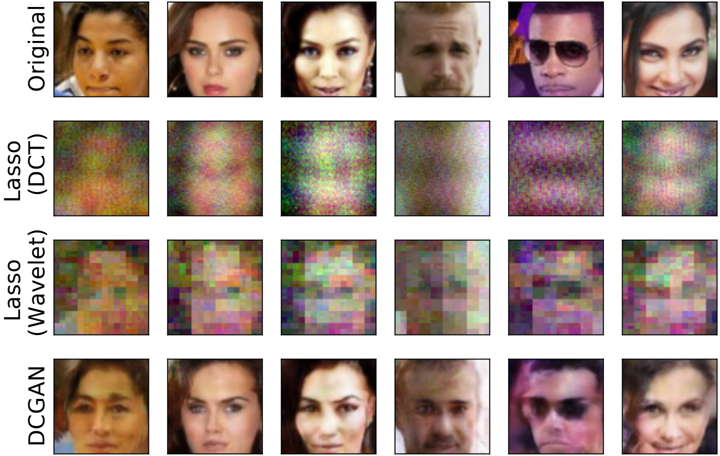
Experimental Results

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



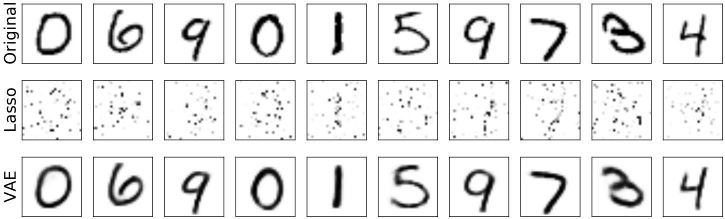
Experimental Results

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

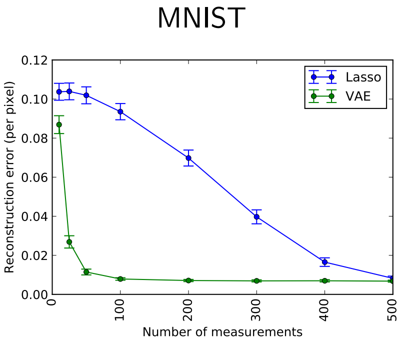


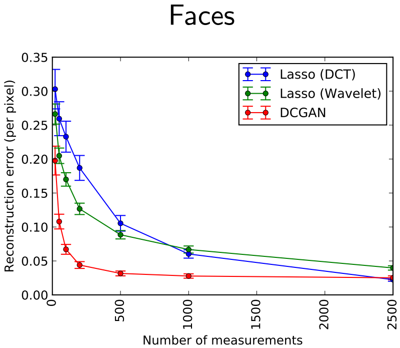
Experimental Results

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



Experimental Results





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)

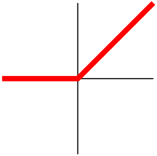
- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:

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 $z \to \mathrm{ReLU}(A_i z)$.

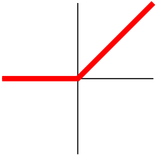
Proof Outline (ReLU-based networks)

- 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 $z \to \text{ReLU}(A_i z)$.



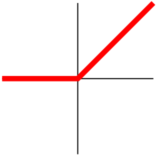
Proof Outline (ReLU-based networks)

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



Proof Outline (ReLU-based networks)

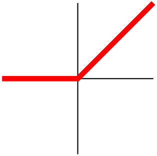
- Show range(G) lies within union of n^{dk} k-dimensional hyperplane.
  - Then analogous to proof for sparsity: $\binom{n}{k} \leq 2^{k \log(n/k)}$ hyperplanes.
  - So $dk \log n$ Gaussian measurements suffice.
- ReLU-based network:
  - Each layer is $z \to \mathrm{ReLU}(A_i z)$.
  - $\mathrm{ReLU}(y)_i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



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



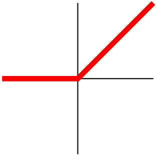
- Input to layer 1: single k-dimensional hyperplane.

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.

- 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.
- $\text{ReLU}(A_1 z)$ is linear, within any constant region of $\text{sign}(A_1 z)$.
- How many different patterns can $\text{sign}(A_1 z)$ take?

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 $\text{sign}(A_1 z)$.
- How many different patterns can $\text{sign}(A_1 z)$ take?
- $k = 2$ version

Proof of Lemma

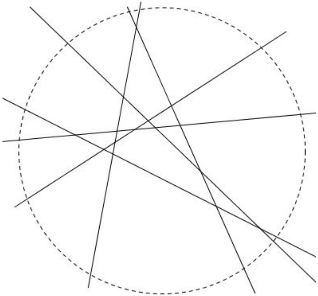
Layer 1's output lies within a union of n^k k-dimensional hyperplanes.

- z is k-dimensional.
- $\mathrm{ReLU}(A_1 z)$ 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.

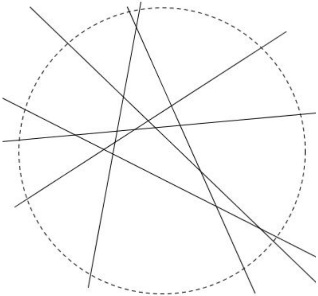
- z is k-dimensional.
- $\text{ReLU}(A_1 z)$ is linear, within any constant region of $\text{sign}(A_1 z)$.
- How many different patterns can $\text{sign}(A_1 z)$ take?
- $k = 2$ version: how many regions can n lines partition plane into?



Proof of Lemma

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

- z is k-dimensional.
- $\mathrm{ReLU}(A_1 z)$ 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?

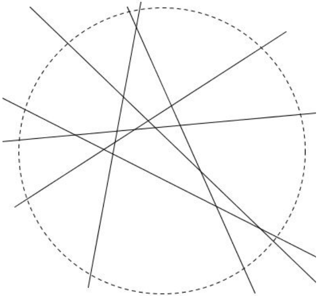


  - $1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2}$.

Proof of Lemma

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

- z is k-dimensional.
- $\mathrm{ReLU}(A_1 z)$ 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?

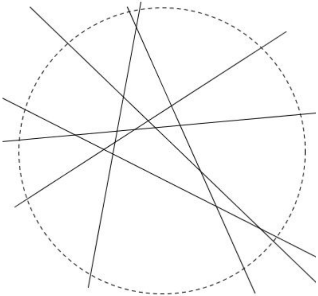


  - $1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2}$.
  - n half-spaces divide $\mathbb{R}^k$ into less than n^k regions.

Proof of Lemma

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

- z is k-dimensional.
- 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?

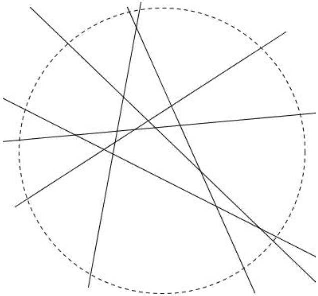


  - $1 + (1 + 2 + \ldots + n) = \frac{n^2+n+2}{2}$.
  - n half-spaces divide $\mathbb{R}^k$ into less than n^k regions. ∎

Proof of Lemma

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

- z is k-dimensional.
- 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. ■
- Therefore d-layer network has n^{dk} regions.

Summary of Compressed Sensing with Generative Models

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

Summary of Compressed Sensing with Generative Models

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

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

Summary of Compressed Sensing with Generative Models

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

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

Summary of Compressed Sensing with Generative Models

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

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

Summary of Compressed Sensing with Generative Models

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

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

Summary of Compressed Sensing with Generative Models

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

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

Summary of Compressed Sensing with Generative Models

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

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

Follow-up work

Theorem

For any L-Lipschitz $G : \mathbb{R}^k \to \mathbb{R}^n$, *recoving* $\widehat{x}$ *from* Ax *satisfying*

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

requires $m = O(k \log \frac{rL}{\delta})$ *linear measurements.*

Follow-up work

Theorem

For any L-Lipschitz $G : \mathbb{R}^k \to \mathbb{R}^n$*, recoving* $\widehat{x}$ *from* Ax *satisfying*

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

requires $m = O(k \log \frac{rL}{\delta})$ *linear measurements.*

- Matching lower bounds: [Liu-Scarlett] (Poster outside!) and [Kamath-Karmalkar-P]

Follow-up work

Theorem

For any L-Lipschitz $G : \mathbb{R}^k \to \mathbb{R}^n$, *recoving* $\widehat{x}$ *from* Ax *satisfying*

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

requires $m = O(k \log \frac{rL}{\delta})$ *linear measurements.*

- Matching lower bounds: [Liu-Scarlett] (Poster outside!) and [Kamath-Karmalkar-P]
- Better results with better G: (Asim-Ahmed-Hand '19)

Follow-up work

Theorem

For any L-Lipschitz $G : \mathbb{R}^k \to \mathbb{R}^n$, recoving $\widehat{x}$ from Ax satisfying

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

requires $m = O(k \log \frac{rL}{\delta})$ linear measurements.

- Matching lower bounds: [Liu-Scarlett] (Poster outside!) and [Kamath-Karmalkar-P]
- Better results with better G: (Asim-Ahmed-Hand '19)
- Provably fast for *random* networks (Hand-Voroninski '18)

Extensions

- Inpainting:

Extensions

- Inpainting:

Extensions

- Inpainting:

  - A is diagonal, zeros and ones.

Extensions

- Inpainting:

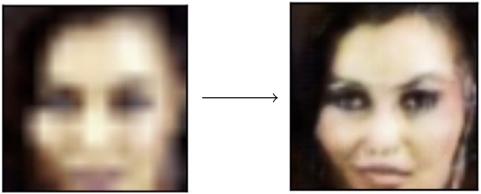
  - A is diagonal, zeros and ones.
- Deblurring:

Extensions

- Inpainting:

  - A is diagonal, zeros and ones.

- Deblurring:

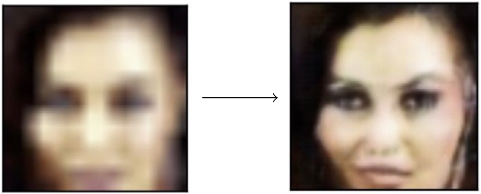


Extensions

- Inpainting:

  - A is diagonal, zeros and ones.

- Deblurring:



- Can apply even to nonlinear—but differentiable—measurements.

Talk Outline

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

Where does the generative model come from?

Where does the generative model come from?

Training from lots of data.

Where does the generative model come from?

Training from lots of data.

Problem

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

Where does the generative model come from?

Training from lots of data.

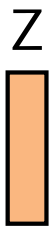
Problem

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

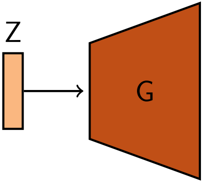
Question

Can we learn a GAN from incomplete, noisy measurements?

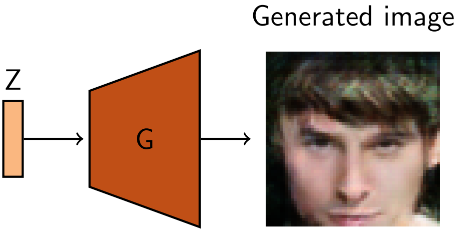
GAN Architecture



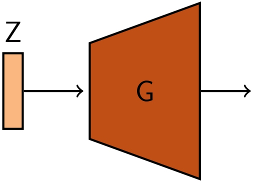
GAN Architecture



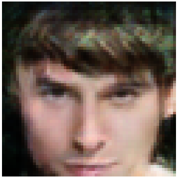
GAN Architecture



GAN Architecture

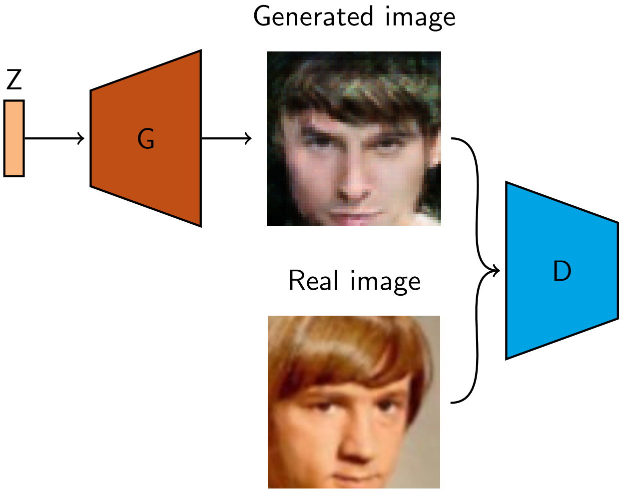


Generated image

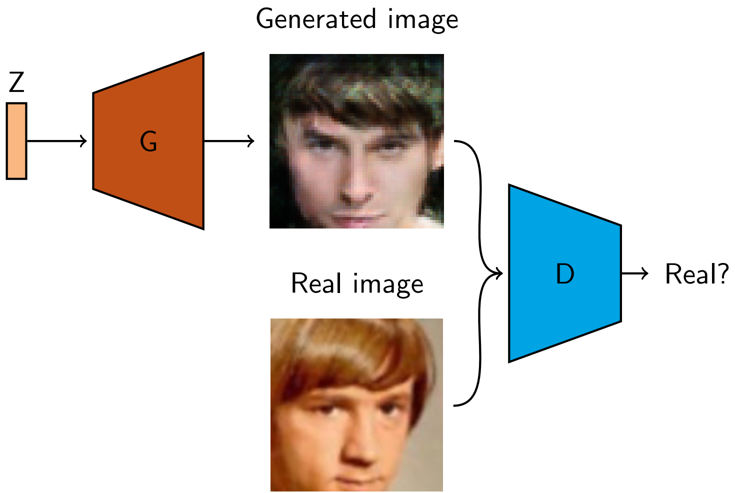


Real image

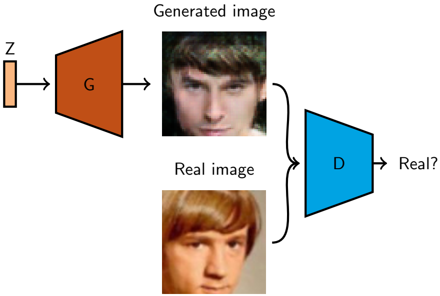
GAN Architecture



GAN Architecture

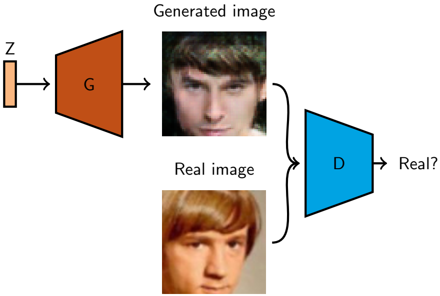


GAN Architecture



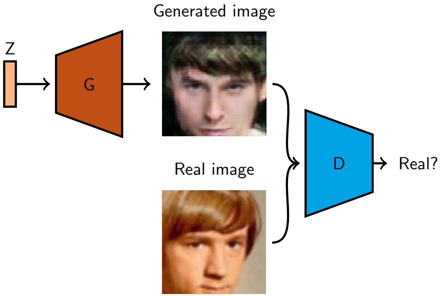
- Generator G wants to fool the discriminator D.

GAN Architecture



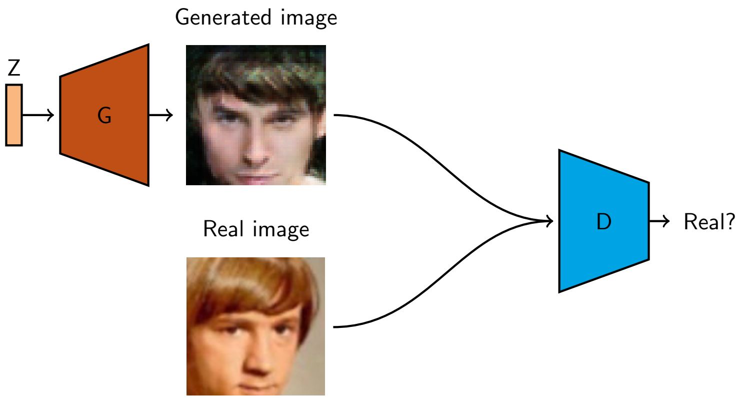
- Generator G wants to fool the discriminator D.
- If G, D infinitely powerful: only pure Nash equilibrium when $G(Z)$ equals true distribution.

GAN Architecture

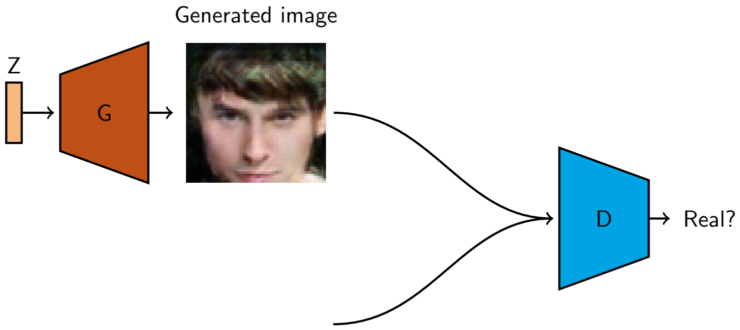


- Generator G wants to fool the discriminator D.
- If G, D infinitely powerful: only pure Nash equilibrium when $G(Z)$ equals true distribution.
- Empirically works for G, D being convolutional neural nets.

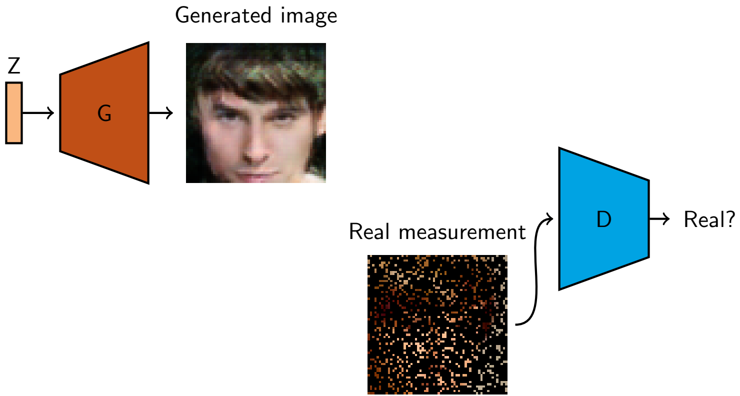
GAN training



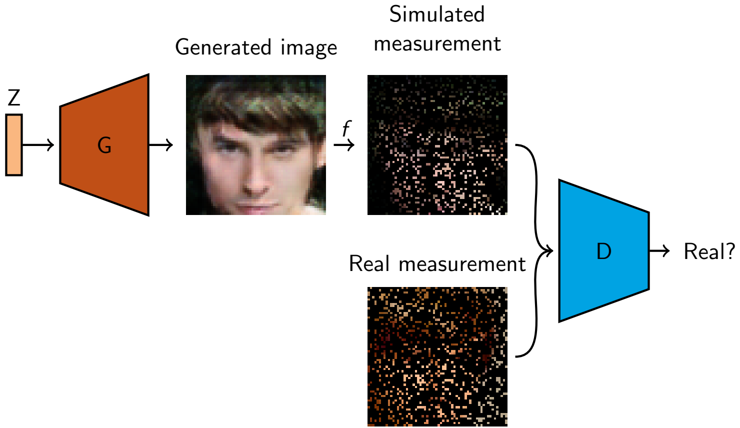
GAN training



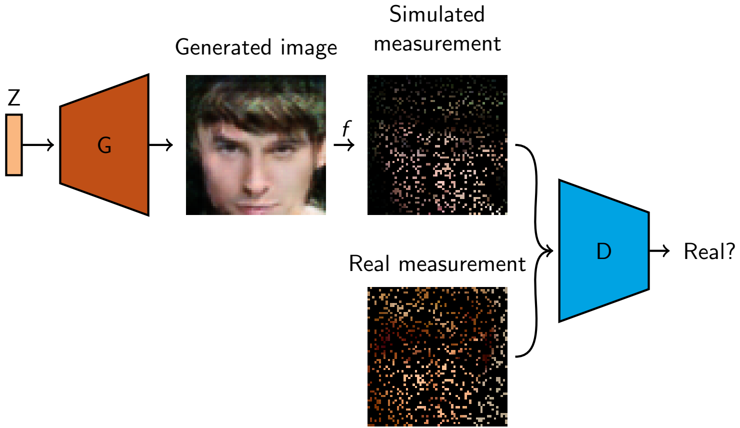
AmbientGAN training



AmbientGAN training

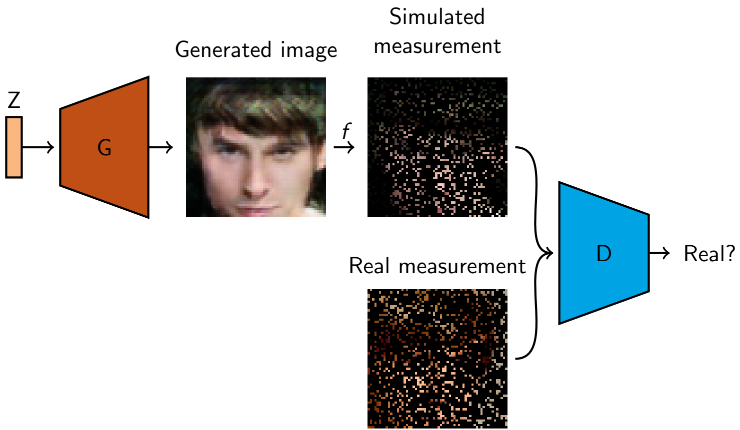


AmbientGAN training



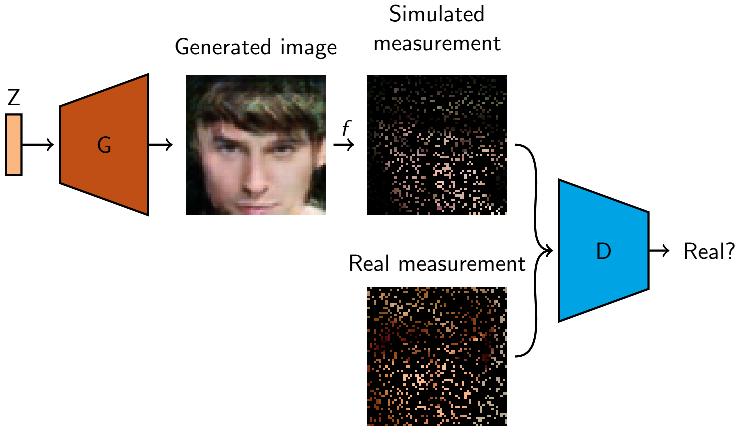
- Discriminator must distinguish *real measurements* from *simulated measurements of fake images*

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.

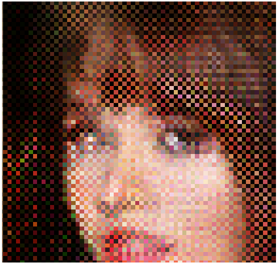
Measurement: Gaussian blur + Gaussian noise

Measured

Wiener Baseline

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

Measurement: Gaussian blur + Gaussian noise

Measured	Wiener Baseline	AmbientGAN
		

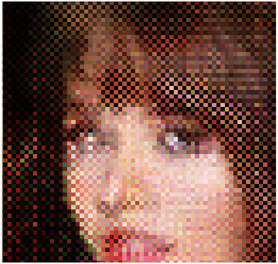
- Gaussian blur + additive Gaussian noise attenuates high-frequency components.
- Wiener baseline: deconvolve before learning GAN.
- AmbientGAN better preserves high-frequency components.

Measurement: Gaussian blur + Gaussian noise

Measured

Wiener Baseline

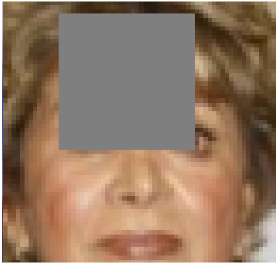
AmbientGAN



- Gaussian blur + additive Gaussian noise attenuates high-frequency components.
- Wiener baseline: deconvolve before learning GAN.
- AmbientGAN better preserves high-frequency components.
- Theorem: in the limit of dataset size and G, D capacity $\to \infty$, Nash equilibrium of AmbientGAN is the true distribution.

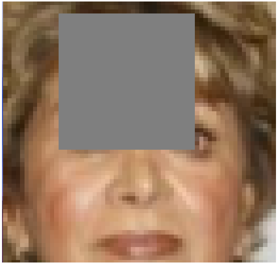
Measurement: Obscured Square

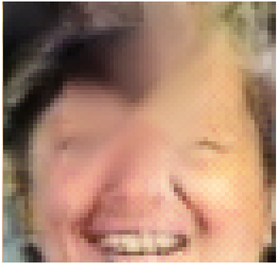
Measured



- Obscure a random square containing 25% of the image.

Measurement: Obscured Square

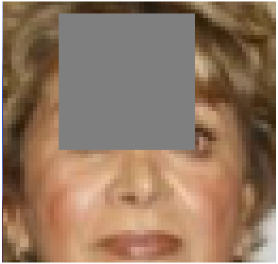




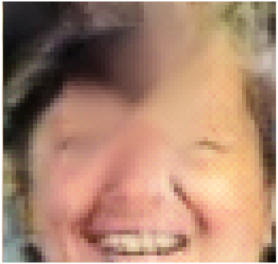
- Obscure a random square containing 25% of the image.
- Inpainting followed by GAN training reproduces inpainting artifacts.

Measurement: Obscured Square

Measured



Inpainting Baseline

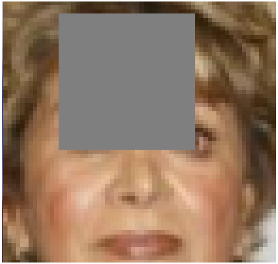


AmbientGAN

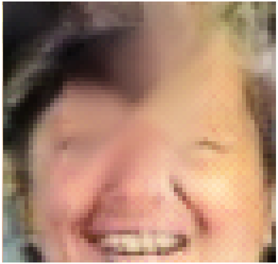
- Obscure a random square containing 25% of the image.
- Inpainting followed by GAN training reproduces inpainting artifacts.
- AmbientGAN gives much smaller artifacts.

Measurement: Obscured Square

Measured



Inpainting Baseline



AmbientGAN

- 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.

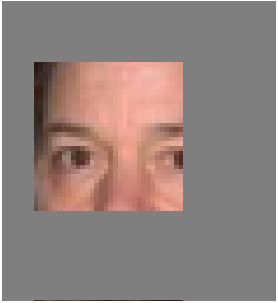
Measurement: Limited View

- 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

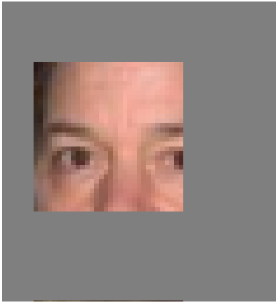


- Reveal a random square containing 25% of the image.

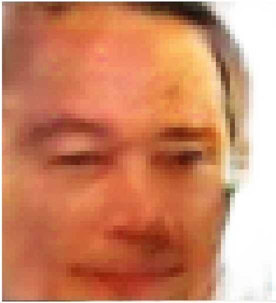
Measurement: Limited View

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

Measured



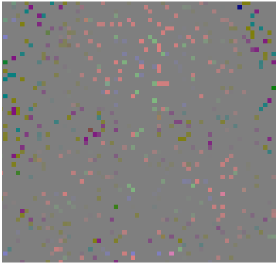
AmbientGAN



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

Measurement: Dropout

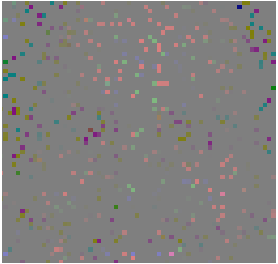
Measured



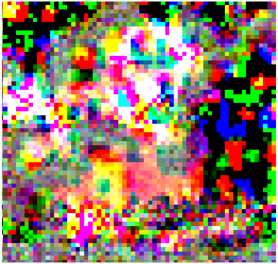
- Drop each pixel independently with probability $p = 95\%$.

Measurement: Dropout

Measured



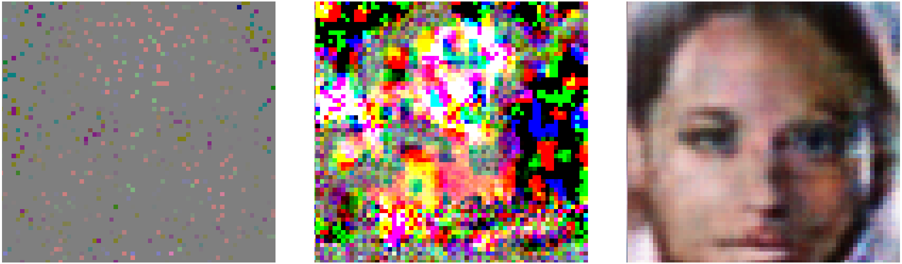
Blurring Baseline



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

Measurement: Dropout

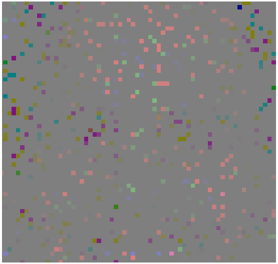
Measured | Blurring Baseline | AmbientGAN



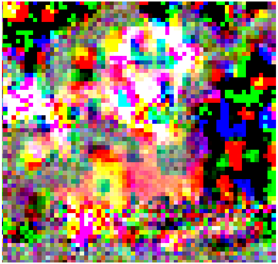
- Drop each pixel independently with probability $p = 95\%$.
- Simple baseline does terribly.
- AmbientGAN can still learn faces.

Measurement: Dropout

Measured



Blurring Baseline



AmbientGAN

- 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.

1D Projections

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

1D Projections

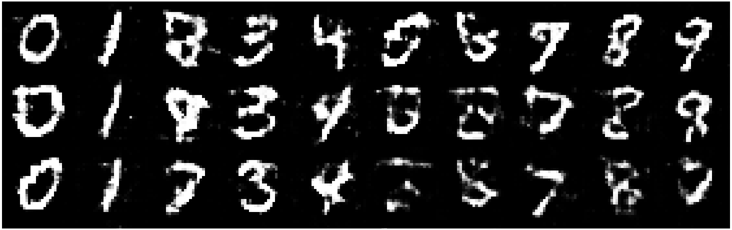
- So far, measurements have all looked like images themselves.
- What if we turn a 2D image into a 1D image?
- Motivation: X-ray scans project 3D into 2D.

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:

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:



Robustness to model mismatch

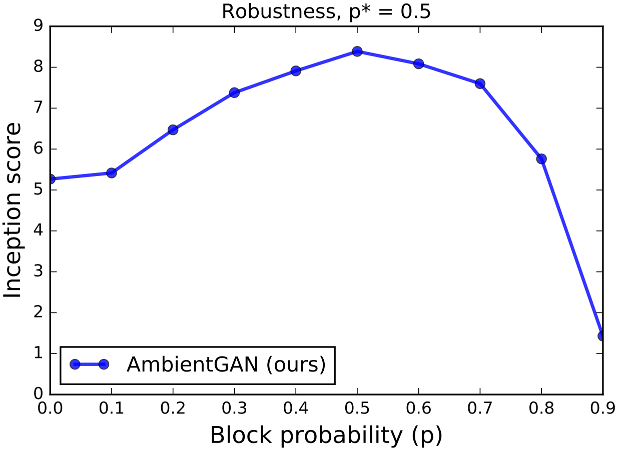
- We assume we know the true measurement process.

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:



Compressed sensing

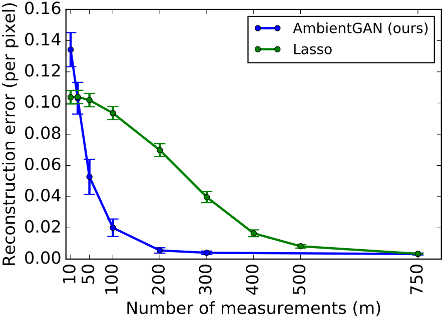
- Compressed sensing: learn an image x from low-dimensional linear projection Ax.

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 $\{(A_i, A_i x_i)\}$.

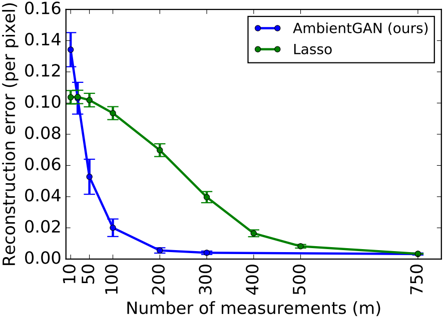
Compressed sensing

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



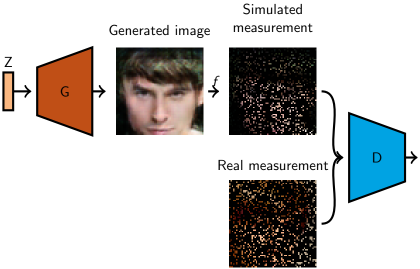
Compressed sensing

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

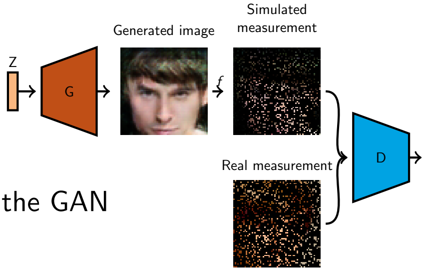


- Theorem about unique Nash equilibrium in the limit.

AmbientGAN

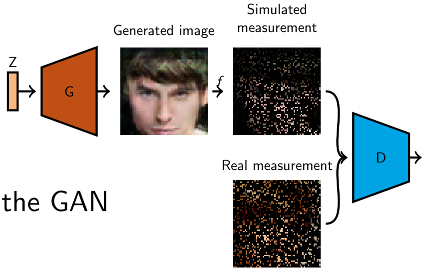

Simulated measurement
Generated image
Z
G
f
D
Real measurement

AmbientGAN



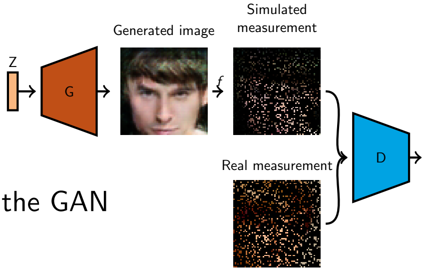
- Plug the measurement process into the GAN architecture of your choice.

AmbientGAN



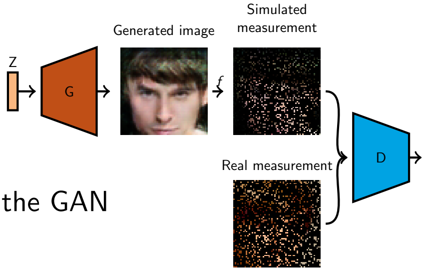
- Plug the measurement process into the GAN architecture of your choice.
- The generator learns the pre-measurement ground truth better than if you denoise before training.

AmbientGAN



- 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.

AmbientGAN



- 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.
  - Can use a generative model to reconstruct from lossy measurements.

Conclusion and open questions

- Main results:
  - Can use lossy measurements to learn a generative model.
  - 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.
  - Can use a generative model to reconstruct from lossy measurements.
- Finite-sample theorems for learning the generative model?
- Better measure than Lipschitzness of generative model complexity?

Conclusion and open questions

- Main results:
  - Can use lossy measurements to learn a generative model.
  - Can use a generative model to reconstruct from lossy measurements.
- Finite-sample theorems for learning the generative model?
- Better measure than Lipschitzness of generative model complexity?
- More uses of differentiable compression?

Conclusion and open questions

- Main results:
  - Can use lossy measurements to learn a generative model.
  - Can use a generative model to reconstruct from lossy measurements.
- Finite-sample theorems for learning the generative model?
- Better measure than Lipschitzness of generative model complexity?
- More uses of differentiable compression?

Thank You