Inverse Problems, Sparsity and Neural Networks Priors

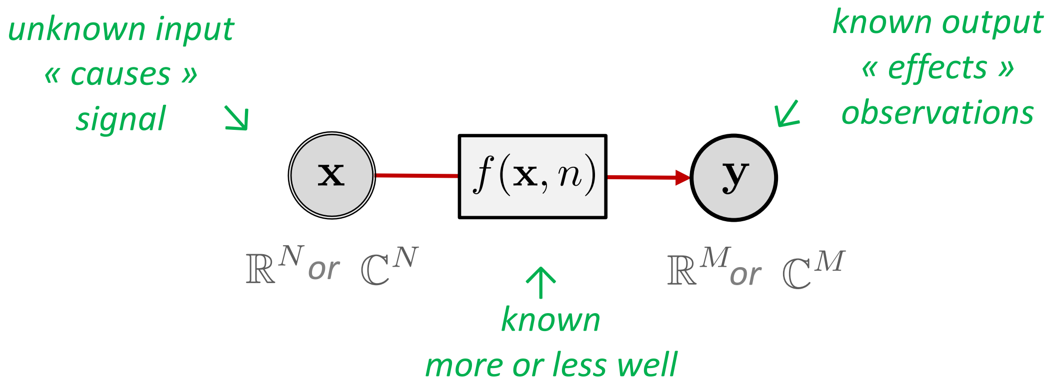
October 20th 2020

FWAM

Marylou Gabrié
(FI CCM & Center for Data Science NYU)

General setting: Inverse problems

"An inverse problem in science is the process of calculating from a set of observations the causal factors that produced them." (Wikipedia)



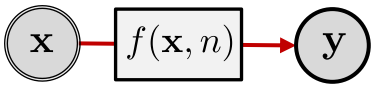
▷ **Forward problem : get y from x**

▷ **Inverse problem : get x from y**
- ▷ Perfect "recovery", invertible function
- ▷ Imperfect/partial recovery if information lost by forward model

Today: Incorporate prior knowledge on the signal to help reconstruction

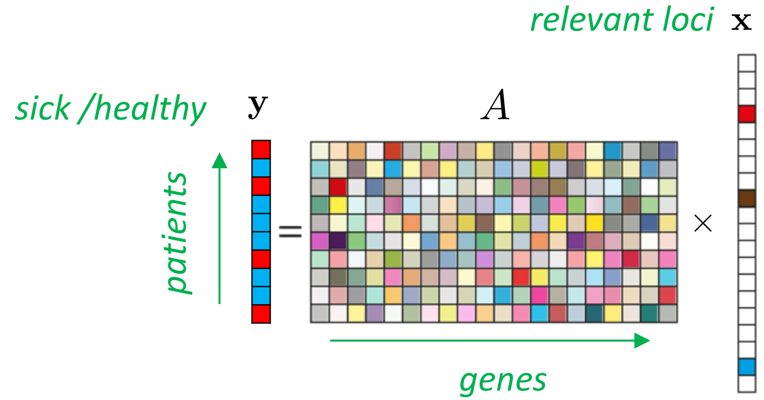
1. Sparsity
2. Neural networks

Examples: Linear Inverse problems



Mathematical formulation:

$$\mathbf{y} = A\mathbf{x} + n$$



Challenges:

- **poor signal-to-noise ratio (SNR)**
- **overdetermined / underdetermined**
- **non-invertible / badly conditioned *A***

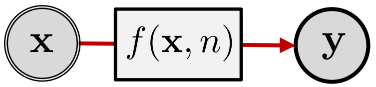
▷ **Genomics**:

- i-th row of *A* gene sequence of an individual
- y_i indicator of healthy / sick patient
- **x** indicator of relevant loci in the genome to the disease

▷ **Image processing: deblurring**

- *A* convolution with a translation inv. Gaussian kernel
- **y** blurred image
- **x** original image

Examples: Non-Linear Inverse problems



Mathematical formulation:

$$\mathbf{y} = f(\mathbf{x}, n)$$

PDE solution, quadratic system etc…

Challenges:

- **"well-posedness"**
- **non-convexity**
- **…**

▷ **Seismology**

- $\mathbf{x}$ density profile
- perturbation + wave propagation
- $\mathbf{y}$ waves reflected at the surface

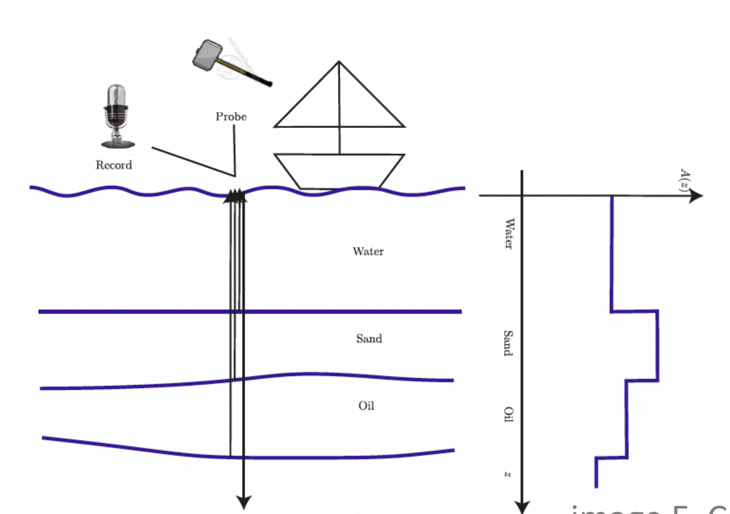


image E. Candès

▷ **Phase retrieval for instance in imaging (coherence diffraction, astronomy)**

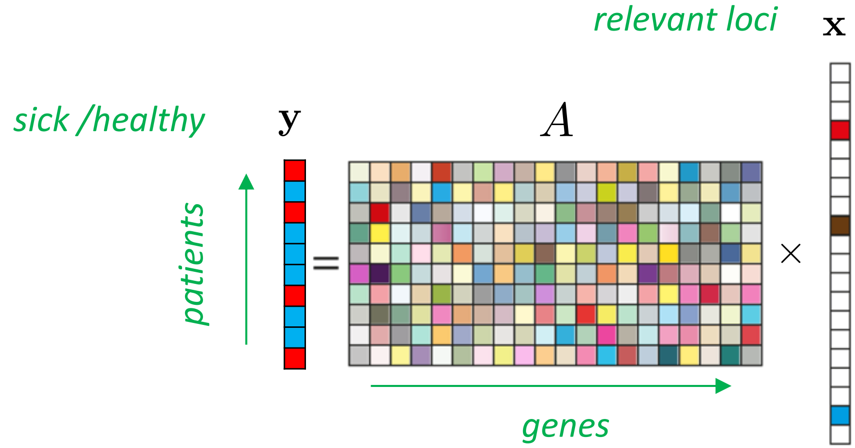
- A Fourier operator (oversampled)
- $\mathbf{y}$ noisy CCD measurements
- $\mathbf{x}$ specimen of interest

$$\mathbf{y} = |A\mathbf{x}| + n$$

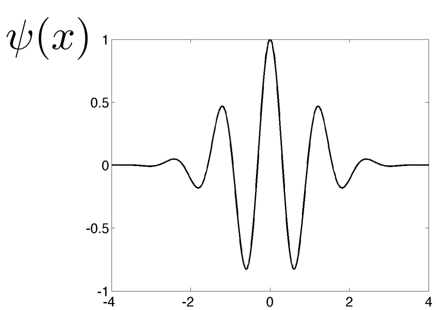
SPARSITY

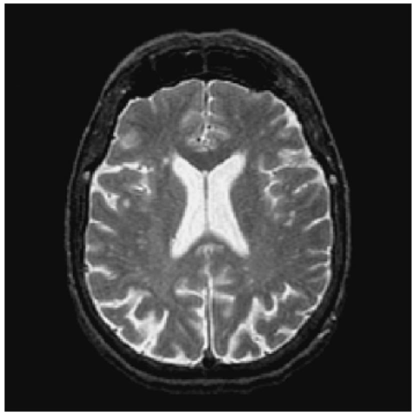
Sparse representation

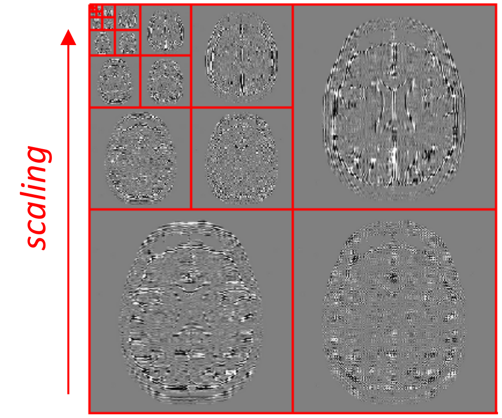
… with respect to a basis



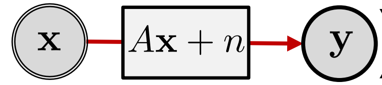
- ▷ **Directly in natural basis of the problem:**
  - ▷ Genomics example
  - ▷ Interfaces in the seismology example
- ▷ **In a specifically chosen basis:**
  - ▷ Fourier analysis
  - ▷ Orthonormal wavelet
    - ▻ mother wavelet $\psi(x)$
    - ▻ translations *m*, scaling factor ℓ $\psi(x)^{\ell}_{m} = 2^{-\ell/2}\psi(2^{-\ell}x - m)$, rotations







[Mallat 1989, Image from G. Peyré]

Leveraging sparsity: Constrained optimization (Linear inverse problems 

)

"Sparse regression"

assumed sparsity

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \{\|\mathbf{y} - A\mathbf{x}\|_2 \,;\, \|\mathbf{x}\|_0 \leq K\}$$

observation fidelity term

ℓ_0-norm counting non-zero coefficients

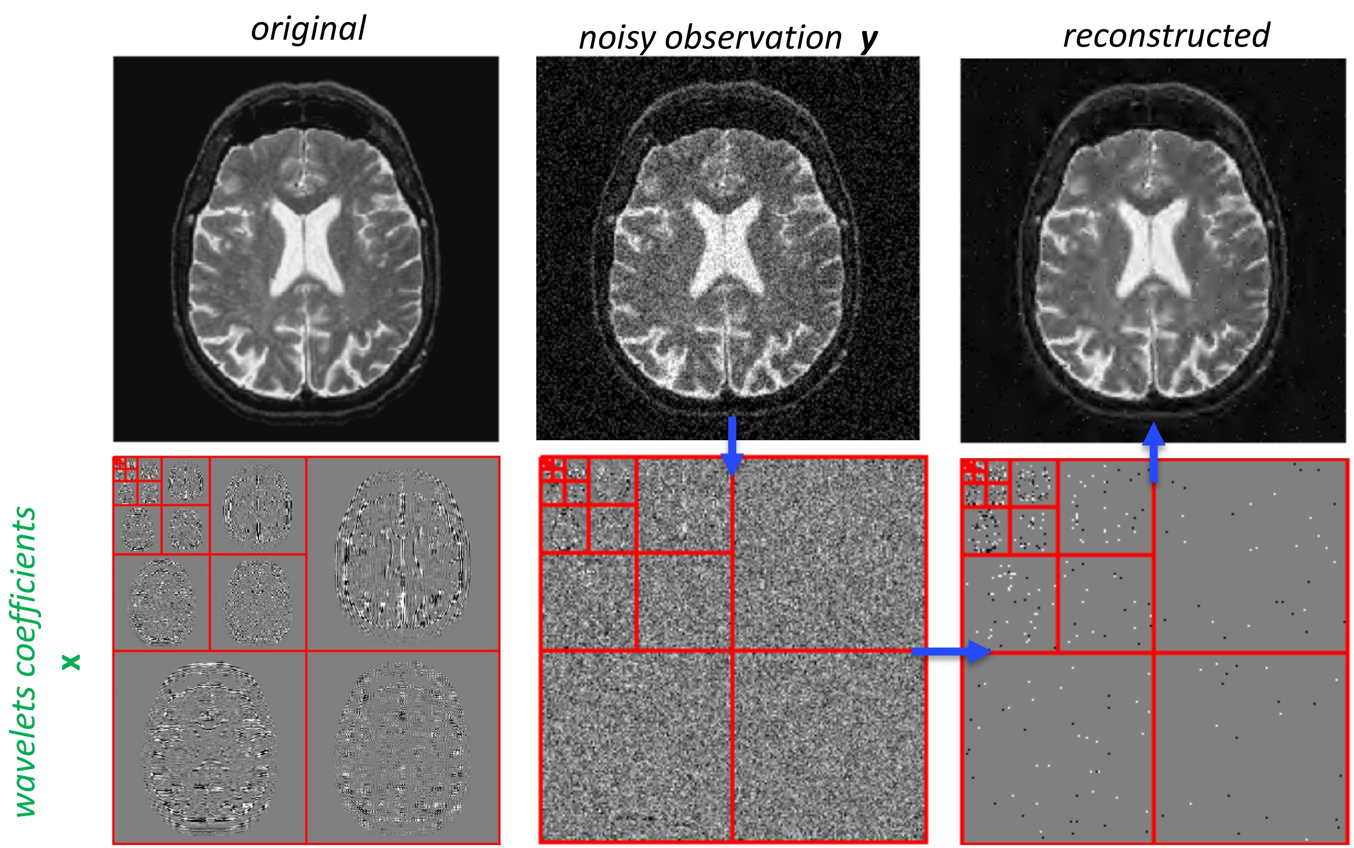
Leveraging sparsity: Constrained optimization (Linear inverse problems)

Example: Image denoising with wavelets $\mathbf{y} = A\mathbf{x} + n$

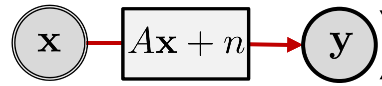
$A =$ *wavelet orthonormal basis*

$\mathbf{x} =$ *wavelet components coef.*

Algorithm:
- Decompose **y** over the wavelet basis
- Keep the *K* wavelets with largest coefficients



[Image from G. Peyré]

Leveraging sparsity: Constrained optimization (Linear inverse problems )

"Sparse regression"

assumed sparsity

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \{\|\mathbf{y} - A\mathbf{x}\|_2 \,;\, \|\mathbf{x}\|_0 \leq K\}$$

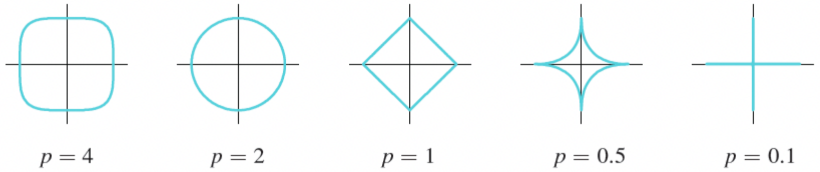
observation fidelity term

ℓ_0-norm counting non-zero coefficients

NP hard problem – No efficient (polynomial time) algorithm in general!

Very active research topic in signal processing in general, many different methods

- Greedy algorithms
- Bayesian methods
- Convex relaxation methods ← *let's focus here*

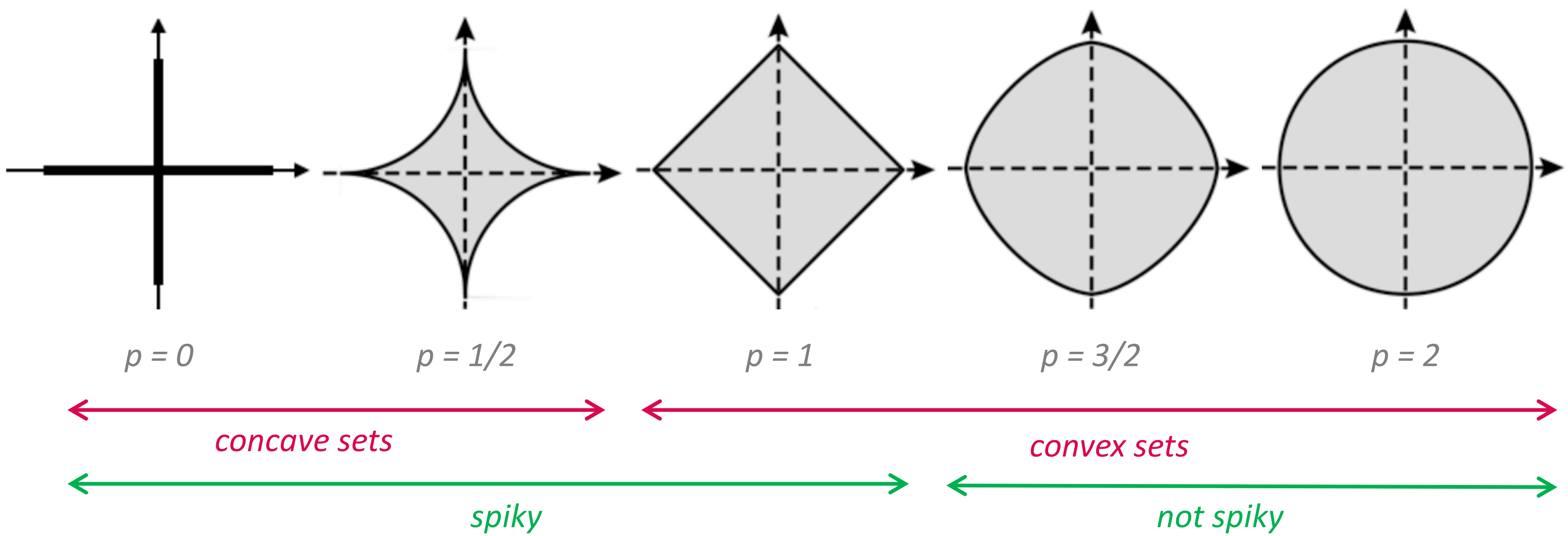


Definition:

For a vector $\mathbf{x} \in \mathbb{R}^N$, $$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{N} x_i^p \right)^{1/p}$$

examples: $\|\mathbf{x}\|_0 =$ *# of non-zero* , $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$, $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{N} x_i^2}$

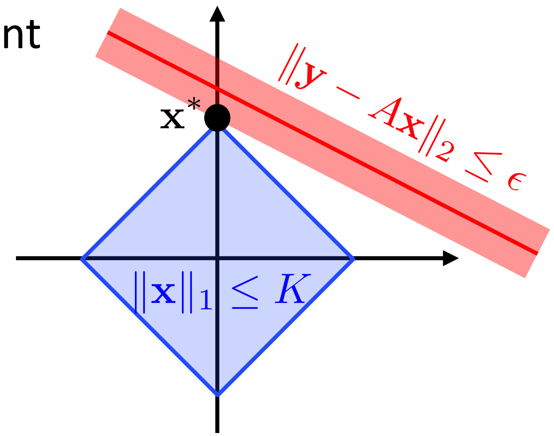
1-balls for the different norms: $\|\mathbf{x}\|_p \leq 1$



ℓ_1-constraint or ℓ_1-regularization

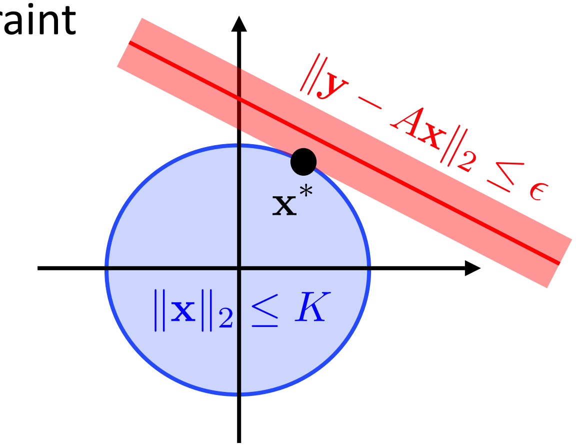
Convex sets intersections

ℓ_1-constraint



$$\min_{\mathbf{x}} \{\|\mathbf{y} - A\mathbf{x}\|_2\} \text{ s.t. } \|\mathbf{x}\|_1 \leq K$$

ℓ_2-constraint



$$\min_{\mathbf{x}} \{\|\mathbf{y} - A\mathbf{x}\|_2\} \text{ s.t. } \|\mathbf{x}\|_2 \leq K$$

LASSO (Tibshirani '96), Basis pursuit (Chen et al. '98)

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \{\|\mathbf{y} - A\mathbf{x}\|_2\,;\, \|\mathbf{x}\|_1 \leq K\} \quad \text{or} \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \{\|\mathbf{y} - A\mathbf{x}\|_2 + \lambda\, \|\mathbf{x}\|_1\}$$

- under structural assumptions on A returns the same as ℓ_0 norm
- still harder than ℓ_2 because ℓ_1 non-differentiable
- but a lot easier than ℓ_0
- finding efficient algorithms very active direction of research

sklearn.linear_model.Lasso

class `sklearn.linear_model.Lasso`(*alpha=1.0, *, fit_intercept=True, normalize=False, precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

Compressive sensing

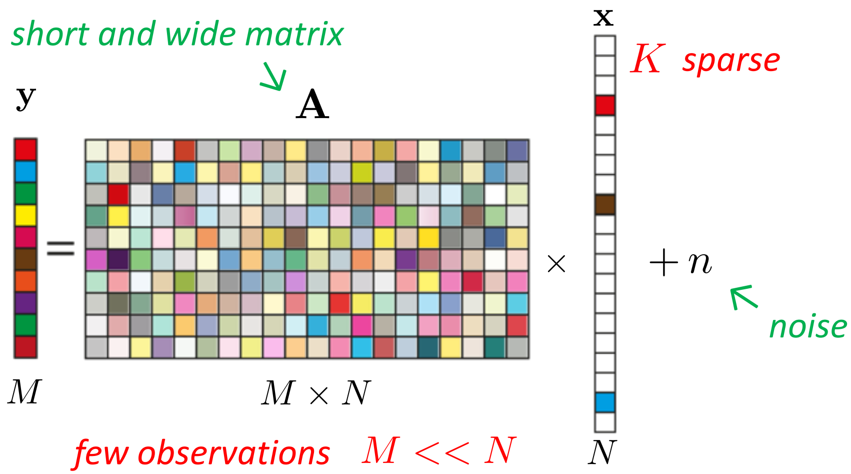
Idea: [Donoho, Candès, Romberg, and Tao in early 2000s]

▷ **Signals which admit sparse representations are compressible**

▷ **Design an acquisition of the signal already compressed**

Implementation:

- ▹ **Sub Nyquist sampling:** Number of measurements proportional to sparsity



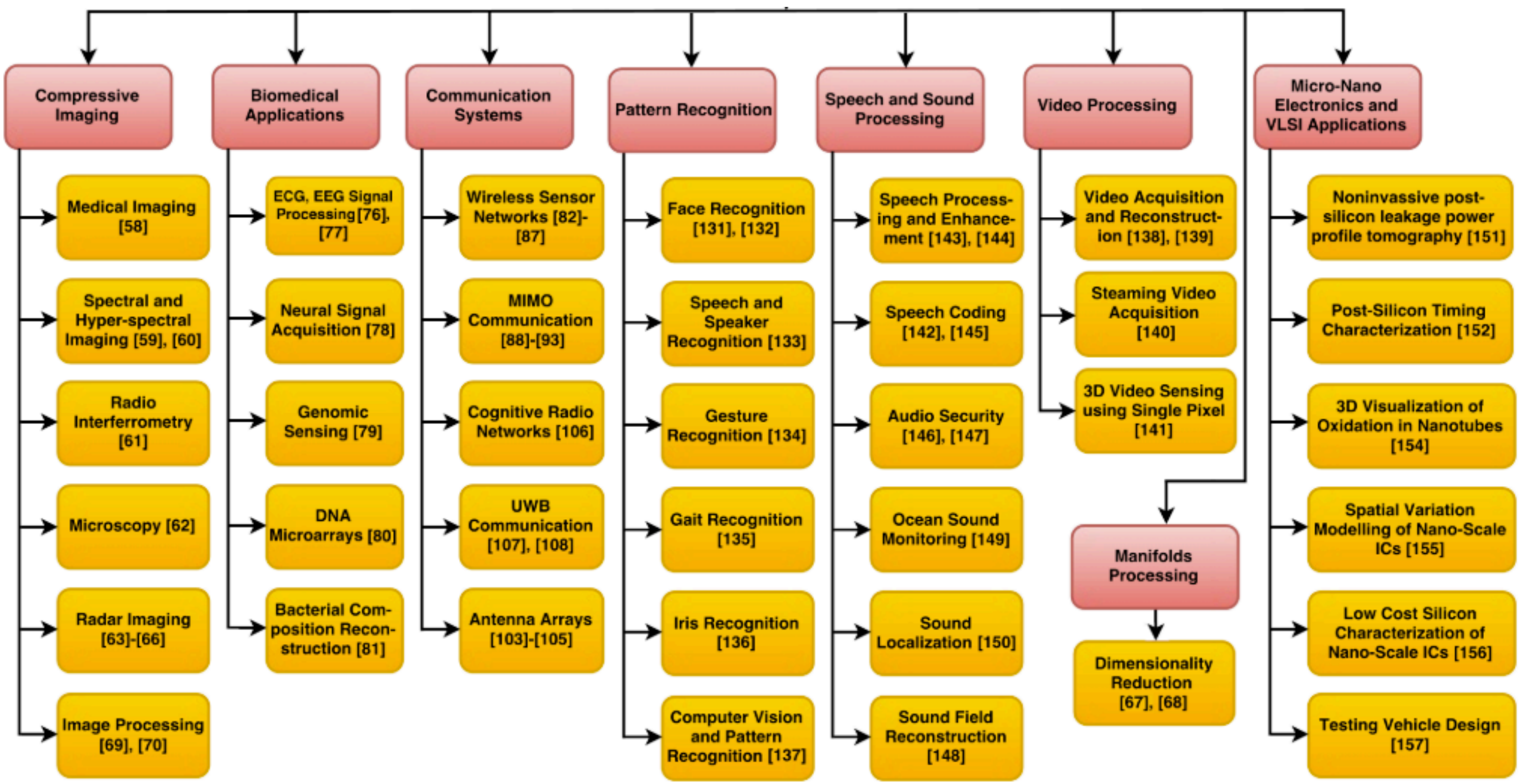
- ▹ **Reconstruction from the observations**: sparse regression (discussed above)

- ▹ **Measurement matrix design:**
  - a lot of theoretical work on guarantees for *M* vs *K* depending on properties of *A*
  - randomness particularly efficient:

    e.g. Gaussian random i.i.di, randomly subsampled Fourier

Applications of compressed sensing

Expensive and/or time-consuming measurements



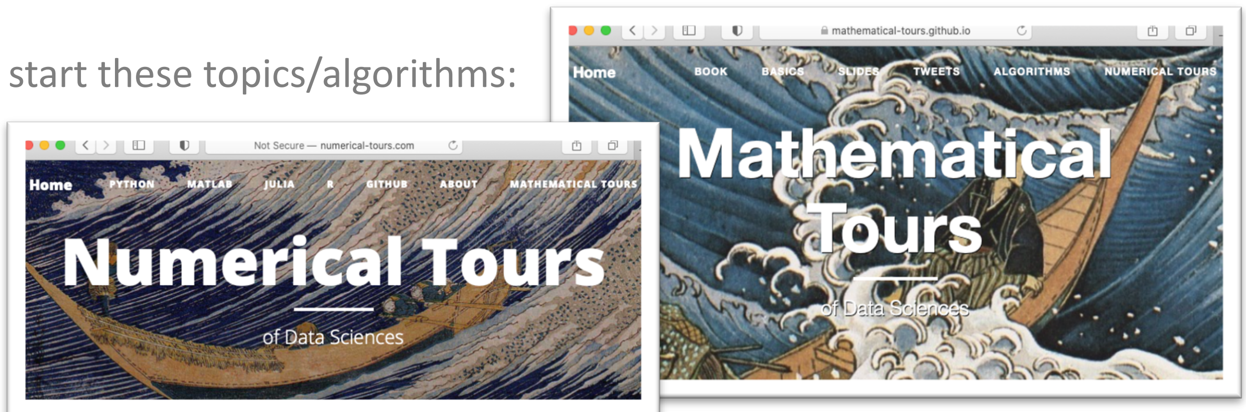
[A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications, Rani et al 2018]

Sparsity and Beyond

What we have seen:

- ▷ Exploiting sparsity really had a tremendous impact
- ▷ Focused on linear inverse problems: but intuition similar for non-linear inverse problems

Nice reference to start these topics/algorithms:



[websites by G. Peyré]

Now:

- ▷ Neural networks as more sophisticated models of signals
- ▷ How to use them in inverse problems

NEURAL NETWORK PRIORS

Learning Data representation with Generative models

Idea:

Use expressivity of neural networks to model non-trivial high dimensional data distributions

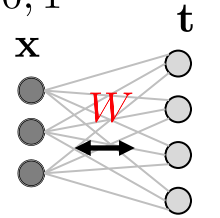
- ▷ **Sampling - Architectures**
  - ▷ Restricted Boltzmann Machines
  - ▷ Deep generative model
- ▷ **Unsupervised learning – Training procedures**
  - ▷ Maximum likelihood
  - ▷ Adversarial training

Restricted Boltzmann Machine (RBM)

Definition: RBM are energy based models

pairwise interactions input-hidden units ↓

$x_i = 0, 1$



$t_\alpha = 0, 1$

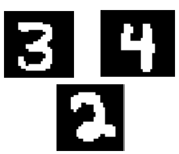
$$p(\mathbf{x}, \mathbf{t}) = \frac{1}{\mathcal{Z}} e^{\sum_{i=1}^{N} b_{x,i} x_i + \sum_{\alpha=1}^{M} b_{t,\alpha} t_\alpha + \sum_{\alpha,i} W_{i\alpha} x_i t_\alpha}$$

$$p(\mathbf{x}) = \int \mathrm{d}\mathbf{t}\; p(\mathbf{x}, \mathbf{t})$$

→ *effective interactions **all orders***

Unsupervised learning:

$$\mathcal{D} = \{\mathbf{x}^{(k)}\}_{k=1}^{P}$$

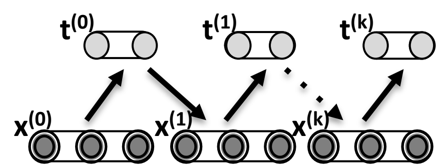


MNIST

$$\max_{W, \mathbf{b}_x, \mathbf{b}_t} \prod_{k=1}^{P} p(\mathbf{x}^{(k)})$$

maximum likelihood learning →

$$\theta = \{\mathbf{b}_x,\; \mathbf{b}_t,\; W\}$$

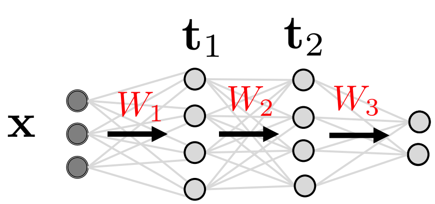


Markov Chain sampling →

$\sim p(\mathbf{x})$

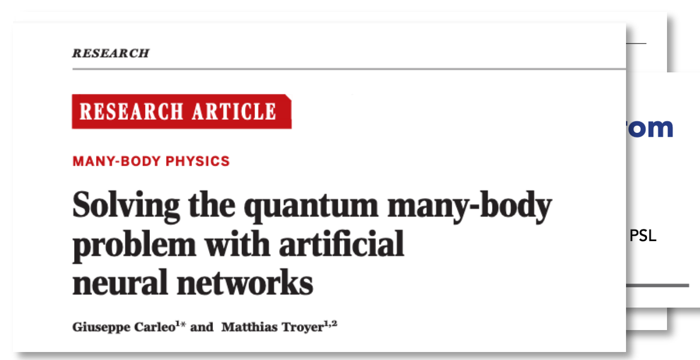
Applications:

Pretraining of deep networks



Biophysics models

Quantum physics



RESEARCH

RESEARCH ARTICLE

MANY-BODY PHYSICS

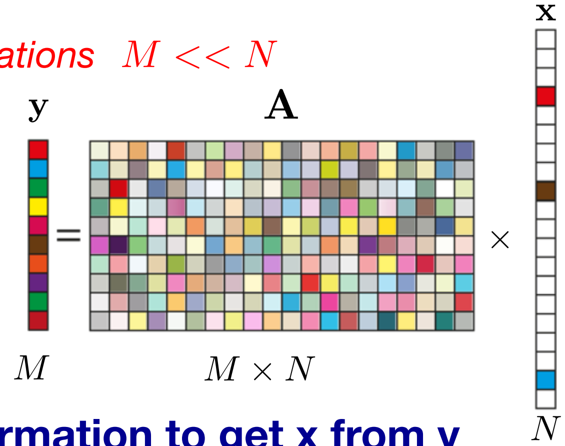
Solving the quantum many-body problem with artificial neural networks

Giuseppe Carleo[1*] and Matthias Troyer[1,2]

Generative priors for inverse problems: First example Compressed Sensing

few observations $M << N$

- ▹ **Recall**

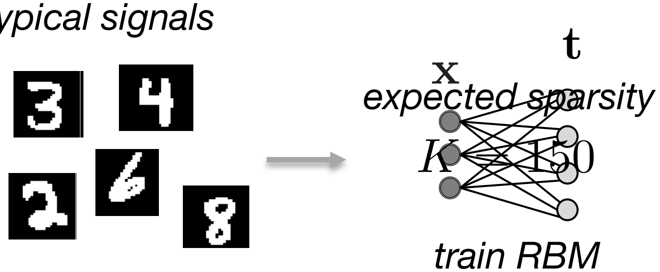


sparse signal

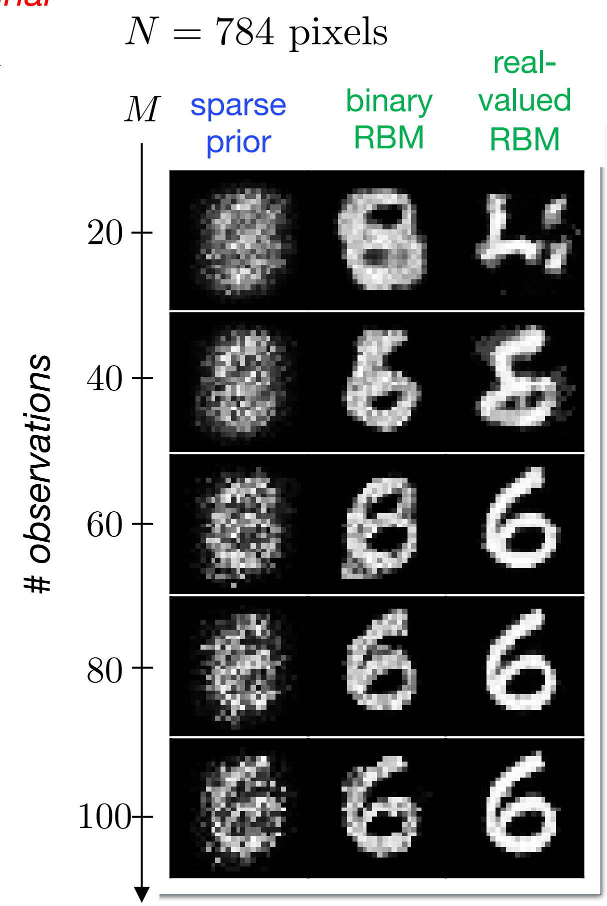
$K << N$

- ▹ **Exploit prior information to get x from y**

typical signals



observations y
\+
CS algorithm: AMP
\+
RBM AMP



- ▹ **RBM learns spatial correlations greatly improves reconstruction**

- ▹ **« Neural networks are the new sparsity ? »**

[E. Tramel et al. JSTAT 2016, E. Tramel, A. Manoel, F Caltagirone, M.Gabrié, & F. Krzakala ITW 2016]

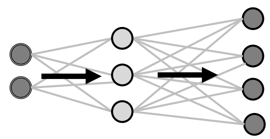
Deep Generative Models

Definition:

latent distribution

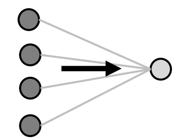
$\mathbf{z} \sim p_z(\mathbf{z})$

Generator



$\mathbf{x} = G_\theta(\mathbf{z})$

Discriminator



$D_\phi(\mathbf{x}) \in [0, 1]$

Unsupervised learning: $\mathcal{D} = \{\mathbf{x}^{(k)}\}_{k=1}^{P}$

▷ Minimum KL /maximum log-likelihood $\min_\theta \mathrm{KL}(p_d(\mathbf{x}) || p_\theta(\mathbf{x}))$ or $\max_\theta \sum_{k=1}^{P} \ln p_\theta(\mathbf{x}^{(k)})$

Variational auto-encoders (VAEs)

▷ Adversarial training $\min_\theta \max_\phi \left[\mathbb{E}_{p_d}\left[\ln D_\phi(\mathbf{x})\right] + \mathbb{E}_{p_z}\left[\ln(1 - D_\phi(G_\theta(\mathbf{z})))\right]\right]$

Generative Adversarial networks (GANs)

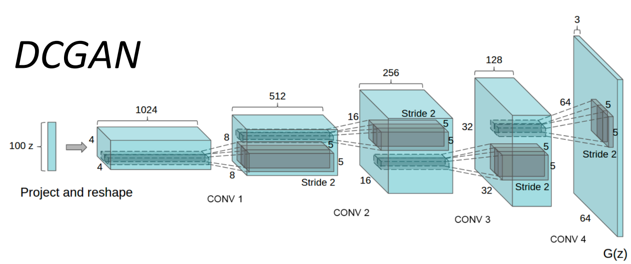
prob of being genuine

prob of being fake

Applications:

- Can include some convolutions
- First generative models able to generate sharp images of great complexity

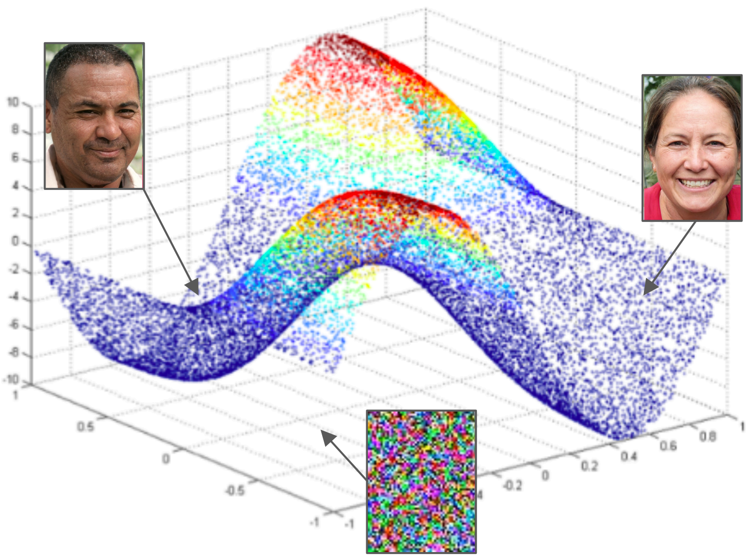
DCGAN



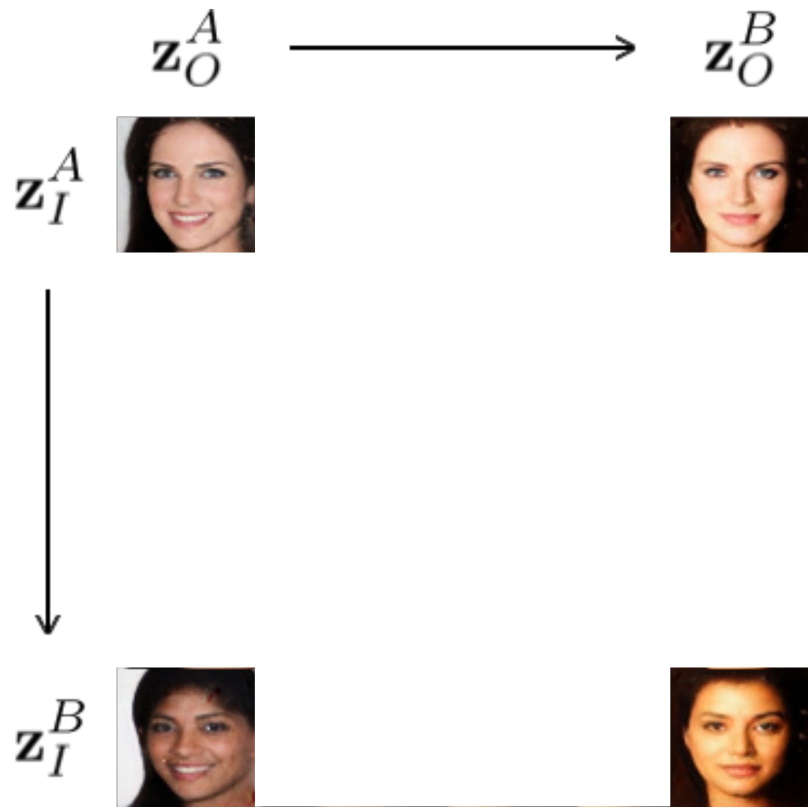
[Goodfellow et al, Neurips 2014; Kigma & Welling Neurips 2014; Radford et al ICLR 2016; Karras et al CVPR 2019]

Intuition of the smaller dimensional manifold

Low dimensional manifold



Interpolating in the latent space

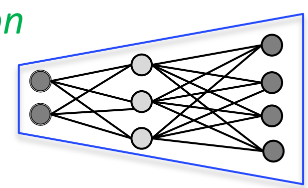


→ learning low dimensional embedding

latent distribution

$\mathbf{z} \sim p_z(\mathbf{z})$

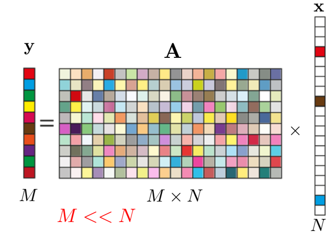
Generator



$\mathbf{x} = G_\theta(\mathbf{z})$

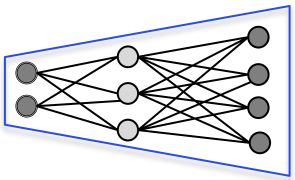
[Donahue et al, ICLR 2018]

Compressed sensing on faces images



- ▷ **Find image in the range of the generator**
- ▷ **In accordance with the observations**

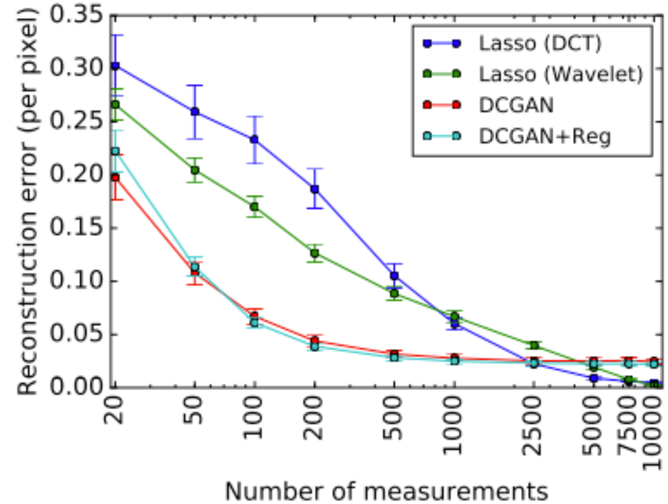
Generator

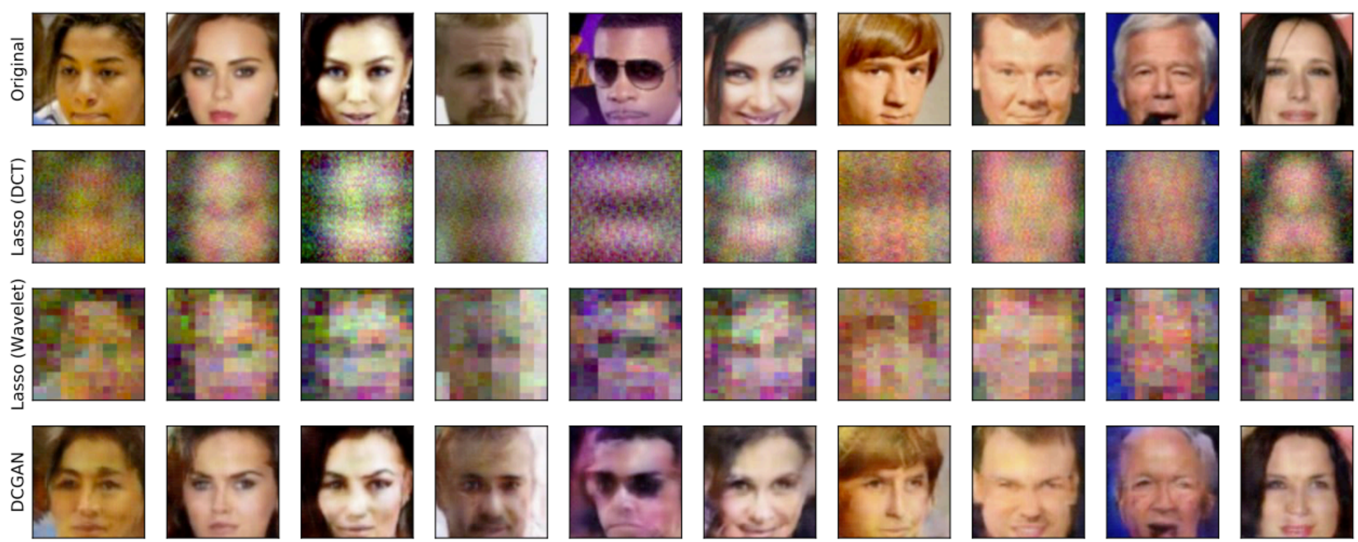


$$\mathbf{x} = G_\theta(\mathbf{z}^*)$$

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \|\mathbf{y} - AG_\theta(\mathbf{z})\|_2$$

↖ *Gradient descent (Pytorch or Tensorflow)*





N = 12288 pixels, M = 2500 measures

[Bora et al, ICML 2017]

General Strategy : Inverse problem solving with Deep Generative models

- ▷ **Original problem:** $$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{y} - f(\mathbf{x}, n)\|_2$$

- ▷ **New strategy**
  - ▷ Train a generative model on typical signals $G_\theta(\mathbf{z})$
  - ▷ Solve inverse problem in the range of generative model

$$\mathbf{z}^* = \arg\min_{\mathbf{x}} \|\mathbf{y} - f(G_\theta(\mathbf{z}), n)\|_2$$

- ▷ **What can go wrong?**
  - ▷ Representation error
  - ▷ Generalization out of the training set
  - ▷ Access to a training set

Limitations of the strategy

Representation error: $\mathbf{z}^* = \arg\min_{\mathbf{z}} \|\mathbf{x}_d - G_\theta(\mathbf{z})\|_2$ $\mathbf{x} = G_\theta(\mathbf{z}^*)$

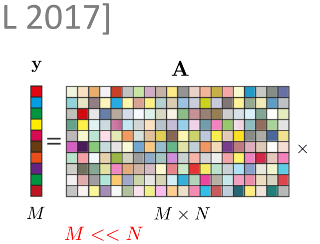
$\mathbf{x}_d$

$\mathbf{x}$

[Bora et al, ICML 2017]

Generalization out of sample:

celebA training data



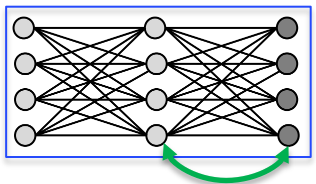
N = 12288 pixels, M = 2500 measures

[Asim et al, ICML 2019]

Another kind of generative models: Normalizing flows

▷ **Bijective networks:**

$$\mathbf{z} \sim p_z(\mathbf{z})$$



$$\mathbf{x} = G_\theta(\mathbf{z})$$

$$\mathbf{z} = G_\theta^{-1}(\mathbf{x})$$

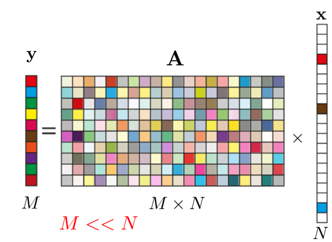
invertible transformations

▷ **Maximum likelihood training**

$$\max_\theta \sum_{k=1}^{K} \ln p_\theta(\mathbf{x}^k)$$

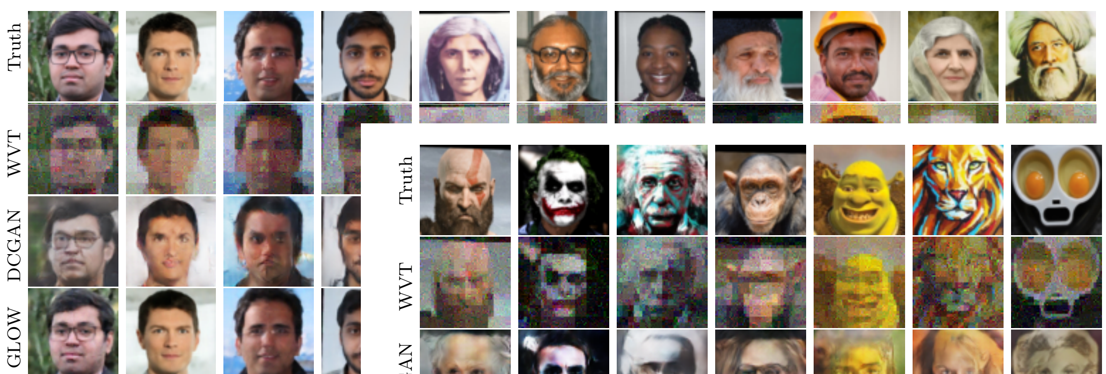
tractable expression

$$p_\theta(\mathbf{x}) = \frac{p_z(G_\theta^{-1}(\mathbf{x}))}{|\nabla_z G_\theta(G_\theta^{-1}(\mathbf{x}))|}$$

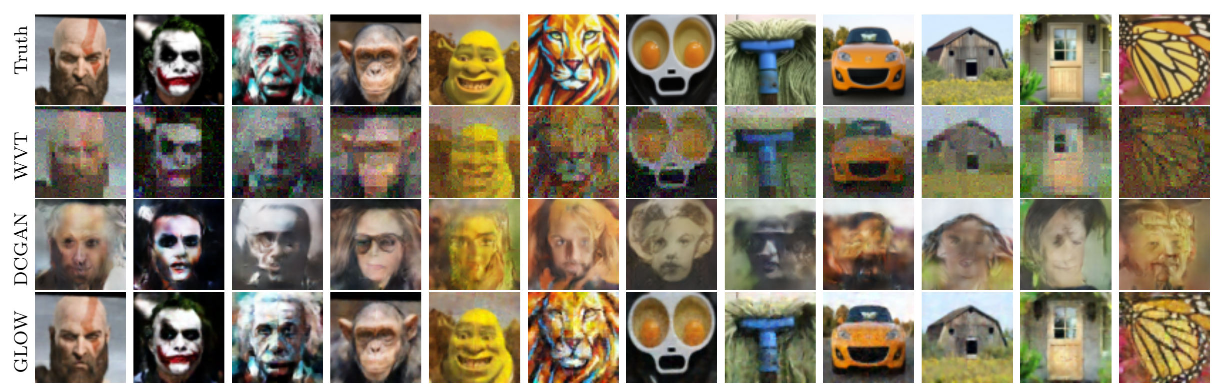


Compressing example:

▷ **Zero representation error**



▷ **Generalizes out of sample**



N = 12288 pixels, M = 2500 measures

[Asim et al, ICML 2019]

General Strategy : Inverse problem solving with Deep Generative models

- **Original problem:**

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{y} - f(\mathbf{x}, n)\|_2$$

- **New strategy**
  - Train a generative model on typical signals
  - Solve inverse problem in the range of generative model

$$\mathbf{z}^* = \arg\min_{\mathbf{x}} \|\mathbf{y} - f(G_\theta(\mathbf{z}), n)\|_2$$

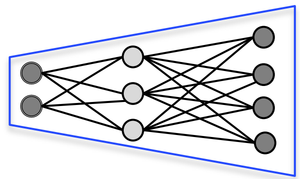
- **What can go wrong?**
  - Representation error
  - Generalization out of the training set
  - Access to a training set

Normalizing flows part of the answer

Untrained image priors

▷ **Trained model**

Generator

$\mathcal{D} = \{\mathbf{x}^{(k)}\}_{k=1}^{P}$ →  → $\mathbf{z}^* = \arg\min_{\mathbf{x}} \|\mathbf{y} - f(G_\theta(\mathbf{z}), n)\|_2$

▷ **Data-set free strategy**

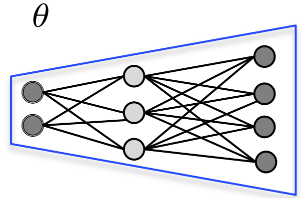
1. draw (random) ***z***

2. adjust parameters of the generator to fit one observation ***y***

3. apply adjusted G to get ***x***

$\mathbf{z} \sim p_z(\mathbf{z})$ → $\theta^* = \arg\min_{\theta} \|\mathbf{y} - f(G_\theta(\mathbf{z}), n)\|_2$ → $\mathbf{x}^* = G_{\theta^*}(\mathbf{z})$

fixed!



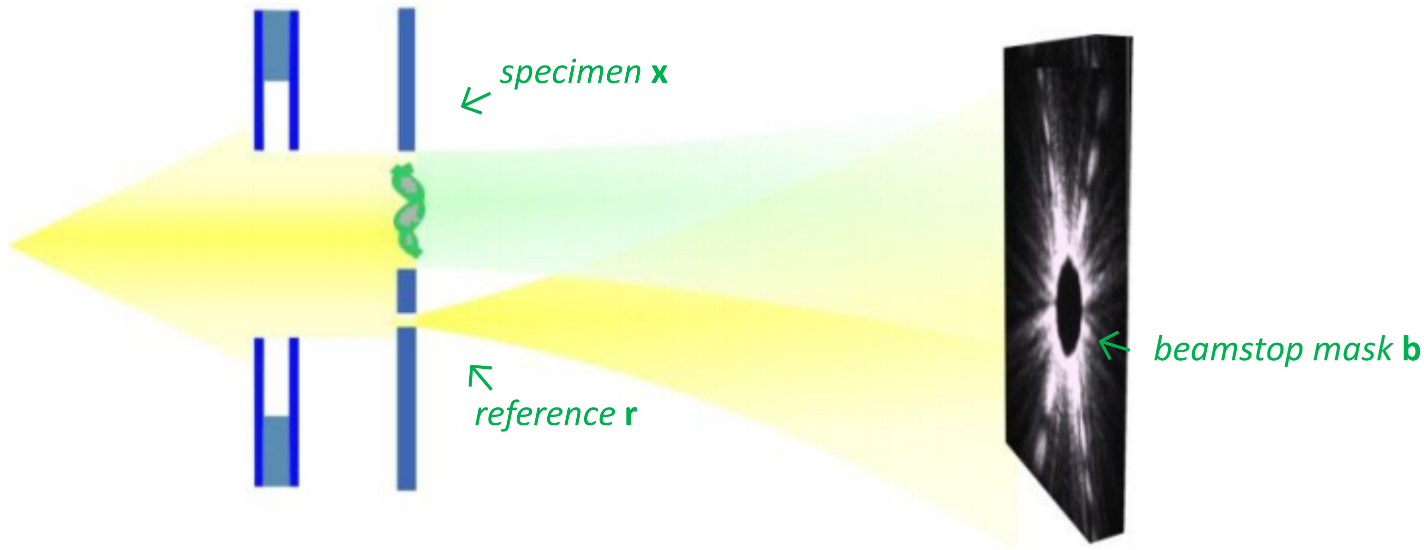
← *neural network architecture (convolutions) biases reconstruction to natural images !*

▷ **Promising for experimental applications!** → *Let's see 2 examples: CDI and MRI*

[Deep Image Prior, Lempitsky et al CVPR 2018; Deep Decoder, Heckle et al ICML 2019]

Applications example I: Coherent Diffraction Imaging (CDI)

- ▷ **CDI:** Imaging technique for nanoscale X-ray imaging
- ▷ **Holography:** Add a known reference in the beam to help the reconstruction



- ▷ **Noiseless forward model** $\mathbf{I}(\mathbf{x}) = |\mathcal{F}(\mathbf{x}+\mathbf{r}) \odot \mathbf{b}|^2$
- ▷ **Low-photon regime** $y_{ij} \sim \text{Poisson}\left(N_p \frac{I_{ij}(\mathbf{x})}{\|\mathbf{I}(\mathbf{x})\|_1}\right)$

N_p = photons/pixel

Holographic phase retrieval: Proposed strategy

forward model:

- $y_{ij} \sim \text{Poisson}\left(N_p \frac{I_{ij}(\mathbf{x})}{\|\mathbf{I}(\mathbf{x})\|_1}\right)$
- $\mathbf{I}(\mathbf{x}) = |\mathcal{F}(\mathbf{x}+\mathbf{r}) \odot \mathbf{b}|^2$

▷ **Incorporate forward model in objective**

(Poisson likelihood ≠ squared loss)

▷ **Reconstruction with an untrained prior (Deep Decoder)**

$$\theta^* = \arg\max_{\theta} \sum_{ij|\mathbf{b}_{ij}=1} y_{ij} \log I_{ij}(G_\theta(\mathbf{z})) - I_{ij}(G_\theta(\mathbf{z}))$$

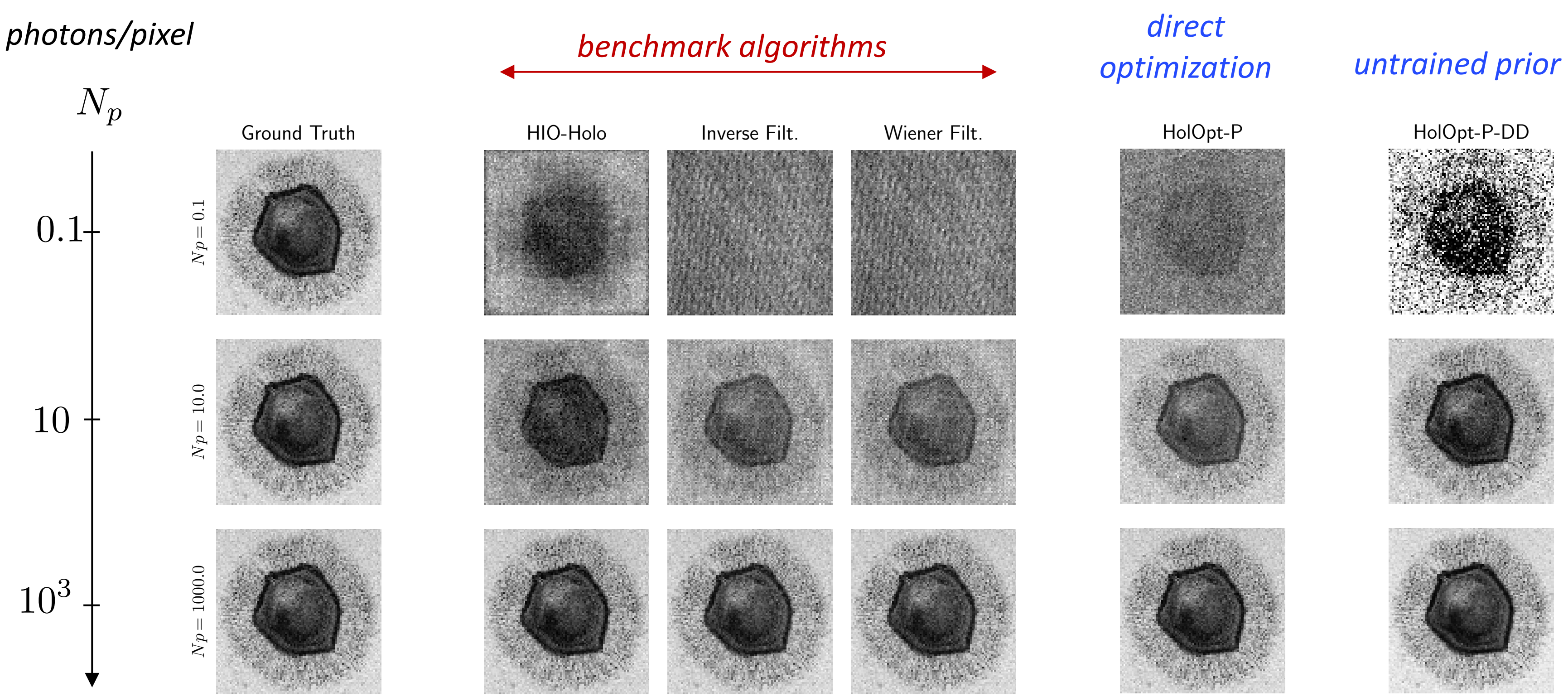
$$\mathbf{x}^* = G_{\theta^*}(\mathbf{z})$$

▷ **Optimization using deep learning packages such as PyTorch**

→ package for Fourier Phase Retrieval on 2D images to be released

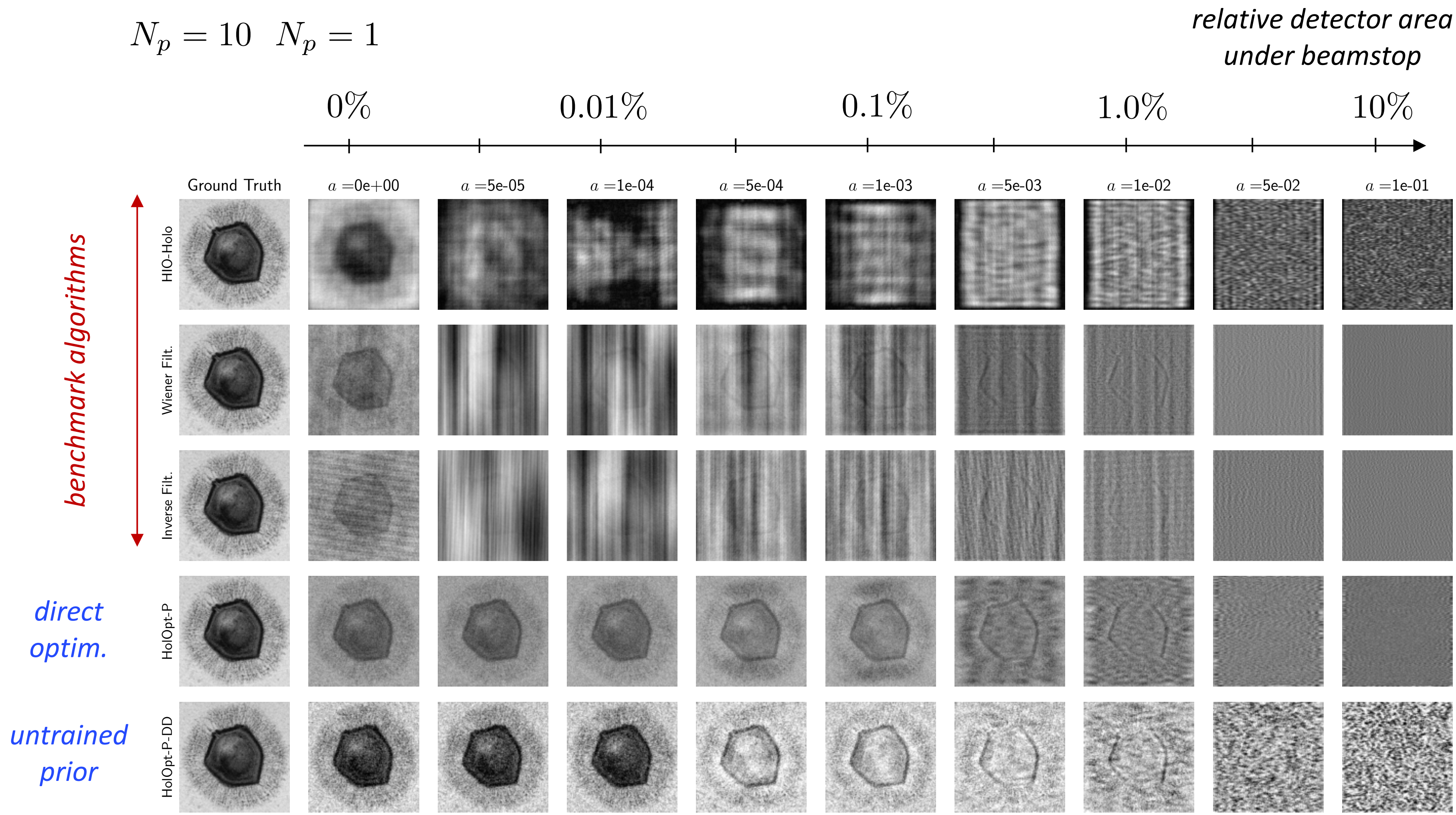
[H. Lawrence, D. Bramherzig, H. Li, M. Eickenberg, M. Gabrié (submitted)]

Holographic CDI: Robustness to noise



[H. Lawrence, D. Bramherzig, H. Li, M. Eickenberg, M. Gabrié (submitted)]

Holographic CDI: Compensating for beamstop

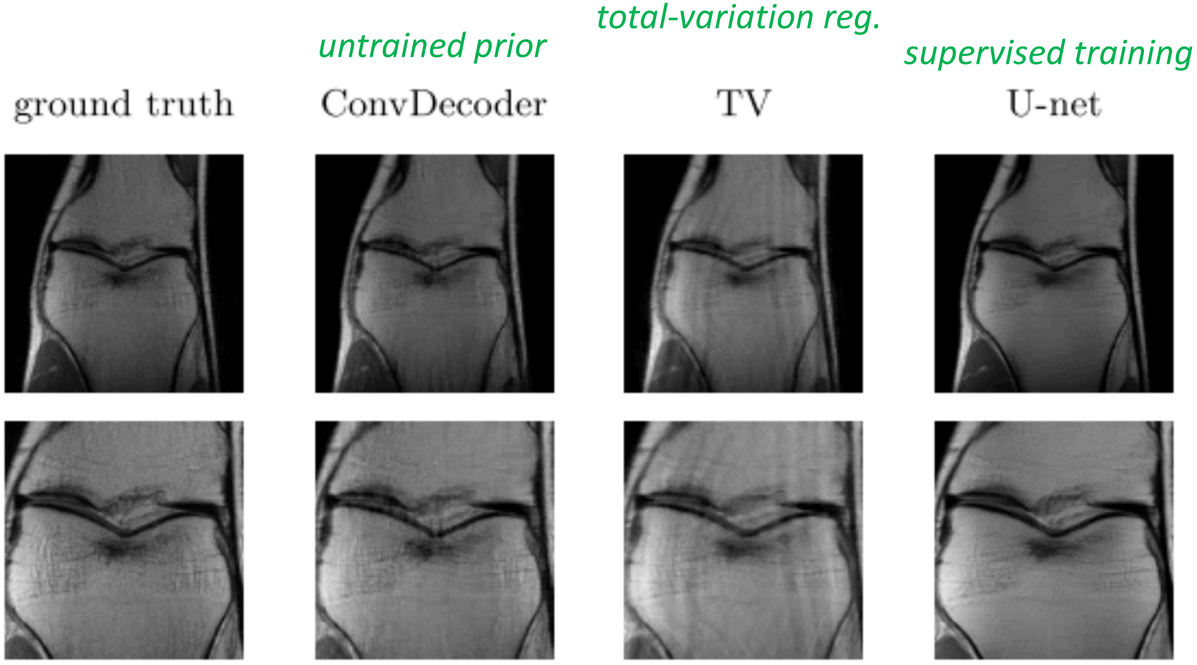


▷ **Practical tool under challenging experimental conditions**

[H. Lawrence, D. Bramherzig, H. Li, M. Eickenberg, M. Gabrié (submitted)]

Application example II: Accelerated Magnetic Resonance Imaging (MRI)

- **FastMRI dataset (https://fastmri.org/)**
  - Experimental data
  - Entire dataset with "ground truth"



- **Untrained prior here as good as trained ML methods**

[Zalbagi Darestani & Heckle, 2020]

Summary: A set of tools for inverse problems

▷ **Sparsity**
- A wide literature on using sparse representation in inverse problems

▷ **Compressive sensing**
- Designing measurements to acquire compressed signal

▷ **Generative neural nets**
- A more sophisticated way to characterize typical signals

▷ **For images: Dataset free - Untrained image priors**
- A dataset free strategy exploiting neural networks architectures

More ways to incorporate machine learning in inverse problems

- ▷ **Learn directly the inverse map y to x**
- ▷ **Use denoising neural networks within other iterative algorithms**

training data available

- ▷ **Many other variants:**

information about forward model

	Supervised with matched (x, y) pairs	**Train from unpaired x's and y's (Unpaired ground truths and Measurements)**	**Train from x's only (Ground truth only)**	**Train from y's only (Measurements only)**
$\mathcal{A}$ fully known during training and testing (§4.1)	§4.1.1: Denoising auto-encoders [16], U-Net [78], Deep convolutional framelets [79] Unrolled optimization [80–83], Neumann networks [84]	*amounts to training from (x, y) pairs*	*amounts to training from (x, y) pairs*	§4.1.2: SURE LDAMP [85, 86], Deep Basis Pursuit [87]
$\mathcal{A}$ known only at test time (§4.2)	§4.2.2	§4.2.2	§4.2.1: CSGM [25], LDAMP [88], OneNet [22], Plug-and-play [89], RED [90]	§4.2.2
$\mathcal{A}$ partially known (§4.3)	§4.3.1	§4.3.2: CycleGAN [91]	§4.3.3: Blind deconvolution with GAN's [92–94]	§4.3.4: AmbientGAN [76], Noise2Noise [95], UAIR [96]
$\mathcal{A}$ unknown (§4.4)	§4.4.1: AUTOMAP [97]	§4.4.2	§4.4.2	§4.4.2

[Deep Learning Techniques for Inverse Problems in Imaging, Ongie et al 2020]

Summary: A set of tools for inverse problems

▷ **Sparsity**
- A wide literature on using sparse representation in inverse problems

▷ **Compressive sensing**
- Designing measurements to acquire compressed signal

▷ **Generative neural nets**
- A more sophisticated way to characterize typical signals

▷ **For images: Dataset free - Untrained image priors**
- A dataset free strategy exploiting neural networks architectures