Self Supervised Learning Methods for Imaging

Part 1: Introduction

Mike Davies, University of Edinburgh

Julián Tachella, CNRS, École Normale Supérieure de Lyon

Tutorial Schedule

PART I: Introduction to Imaging Inverse Problems

Inverse problem framework; ill-posed problems; dimensionality; noise; deep learning solutions; supervised versus unsupervised; learning vs inductive bias.

PART II: Unsupervised methods for Invertible Forward Operator

Blind Denoising; SURE; Noise2X; denoising autoencoders; measurement splitting; GSURE.

PART III: Learning from multiple operators

The impossibility of learning from an incomplete measurement operator; learning from multiple operators; Noise2Noise revisited; measurement splitting revisited; multi-operator consistency; handling noise.

PART IV: Unsupervised methods for ill-conditioned inverse Problems with a single operator

Exploiting Invariance and symmetries; equivariant and non-equivariant operators; enforcing equivariance; handling noise

PART V: Identifiability Theory

Identifiability and dimension; learning with noise; learning from incomplete measurements; Cramer-Wold theorem; generic identifiability

PART VI: Summary and Future Perspectives

The Inverse problem

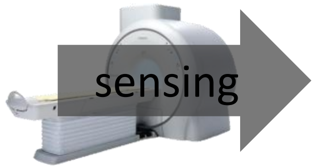
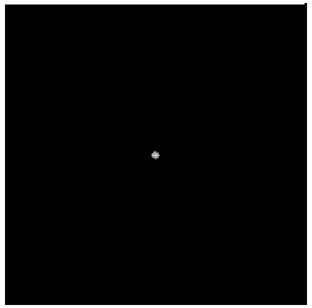
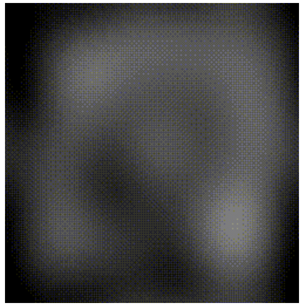
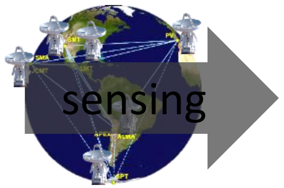
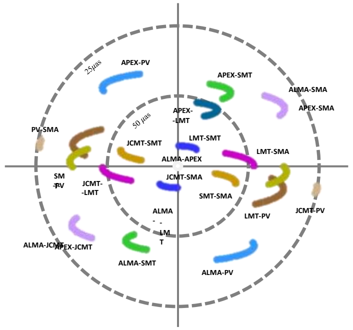
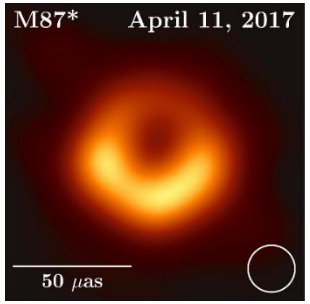
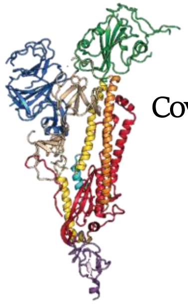
Goal: estimate signal $\boldsymbol{x}$ from $\boldsymbol{y}$

signal $\in \mathbb{R}^n$

$$\underset{\substack{\text{measurements} \\ \in \mathbb{R}^m}}{} \rightarrow \boldsymbol{y} = \underbrace{A}_{\text{Physics}}(\boldsymbol{x}) + \boldsymbol{\epsilon} \leftarrow \text{noise/error}$$

We will focus on linear problems where the forward operator A is a matrix

Examples

	x	A	y		reconstruction	
Magnetic Resonance Imaging (MRI) A: undersampled Fourier models	 ?	 sensing		recover		Source: Brian Hargreaves
Black Hole Imaging A: spatial-frequency e.g. Event Horizon Telescope (EHT)	?	 sensing		recover		The Astrophysical Journal Letters, vol. 875, no. L1, 2019.
Cryogenic electron microscopy (Cryo-EM) A: 2D projections of protein particles	 ?	 sensing		recover		Covid-19 virus' structure D. Wrapp et al. *Science*, vol. 367, no. 6483, 2020.

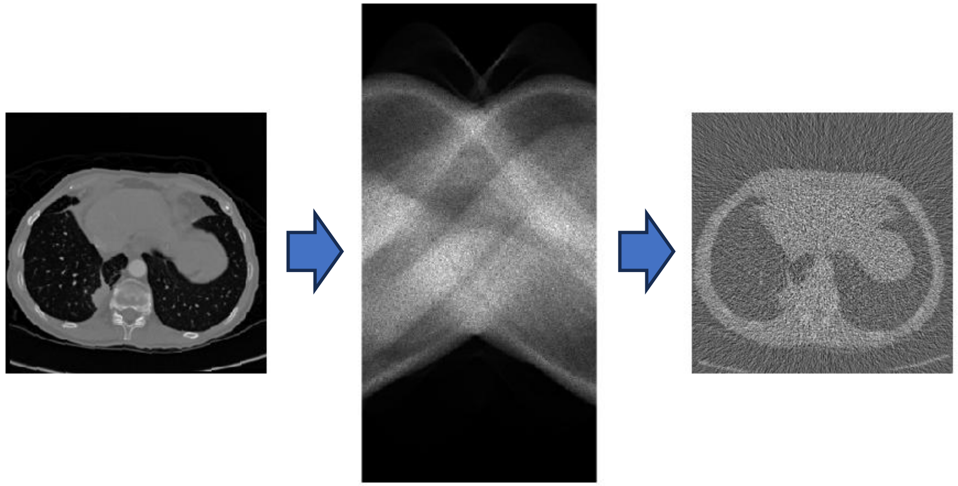
Why it is hard to invert?

Measurements are usually corrupted by noise, e.g.

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{\epsilon}$$

Can be additive, as above, or more complex, e.g. Poisson.

- Often, we do not know the exact noise distribution
- The forward operator may be poorly conditioned



with noise

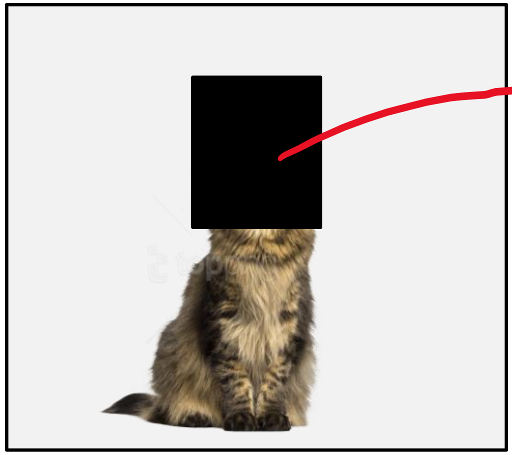
Why it is hard to invert?

Even in the absence of noise, A may not be invertible, giving infinitely many $\hat{\boldsymbol{x}}$ consistent with $\boldsymbol{y}$:

$$\hat{\boldsymbol{x}} = A^{\dagger}\boldsymbol{y} + \boldsymbol{v}$$

where $A^{\dagger}$ is the pseudo-inverse of A and $\boldsymbol{v}$ is any vector in nullspace of A

Unique solution only possible if set of signals $\mathcal{X}$ is low-dimensional



NULLSPACE OF A

reconstruct

Low Dimensional Signal Models

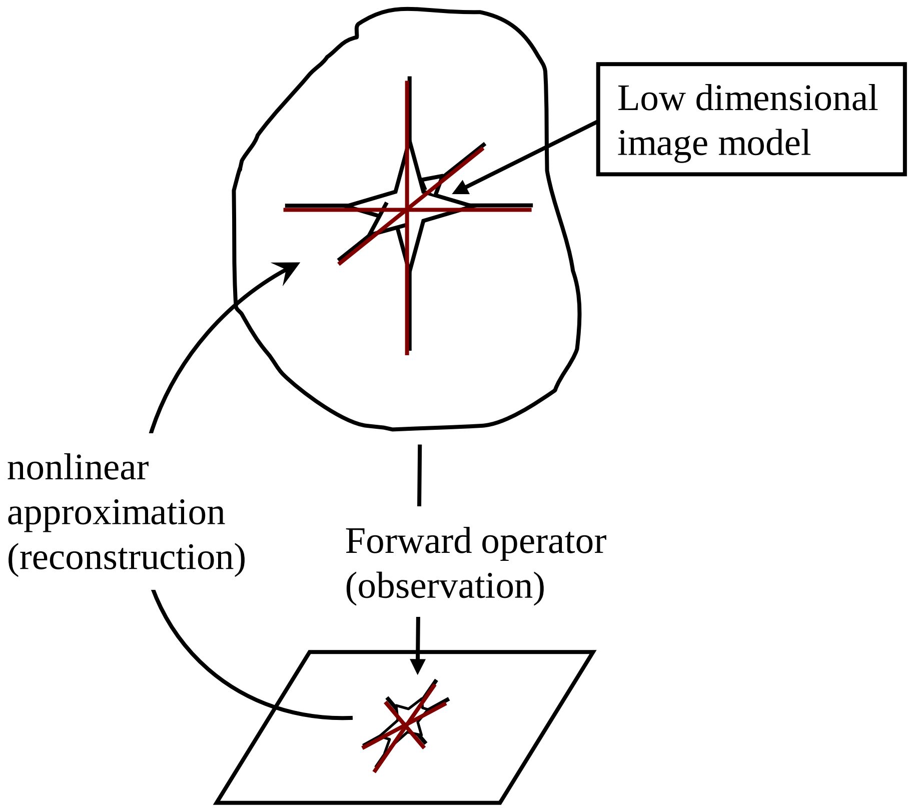
Idea: assume approximate low dimensional image model:

$$\dim \mathcal{X} = k \ll n$$

Examples: sparsity, low-rank, manifolds

Signal Embedding: if $\mathrm{m} \geq \mathcal{O}(k)$ then the problem is approximately one-to-one and (nonlinearly) invertible

This is the principle behind compressed sensing, but is implicit in most inverse imaging problems



Regularised reconstruction

Idea: define a loss $\rho(\boldsymbol{x})$ that promotes plausible reconstructions

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \; ||\boldsymbol{y} - A\boldsymbol{x}||^2 + \rho(\boldsymbol{x})$$

Examples: total-variation, sparsity, etc.

Disadvantages: hard to define a good $\rho(\boldsymbol{x})$ in real world problems, loose with respect to the true signal distribution

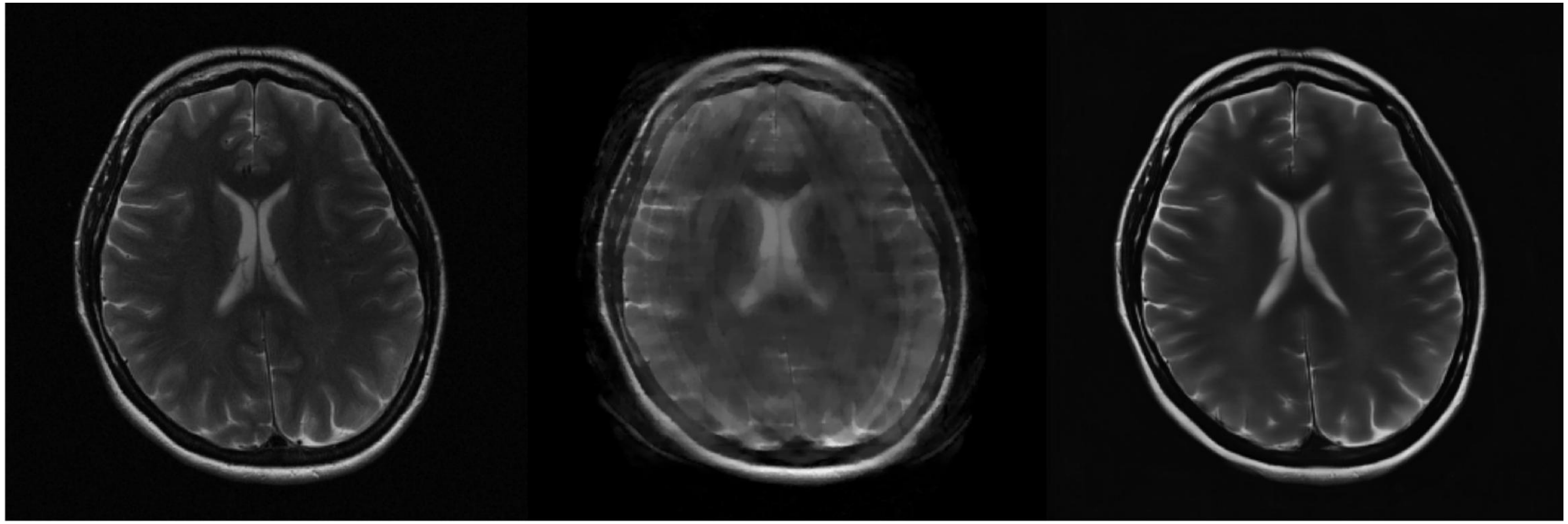
Learning approach

Advantages:

- State-of-the-art reconstructions
- Once trained, f_θ is easy to evaluate

fastMRI

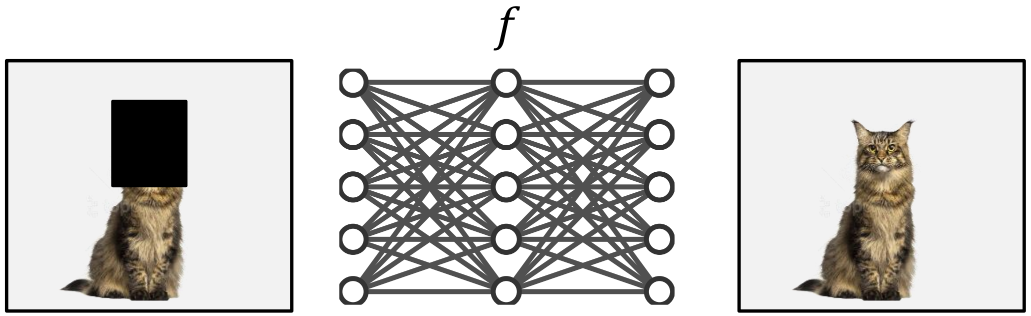
Accelerating MR Imaging with AI



x8 accelerated MRI [Zbontar et al., 2019]

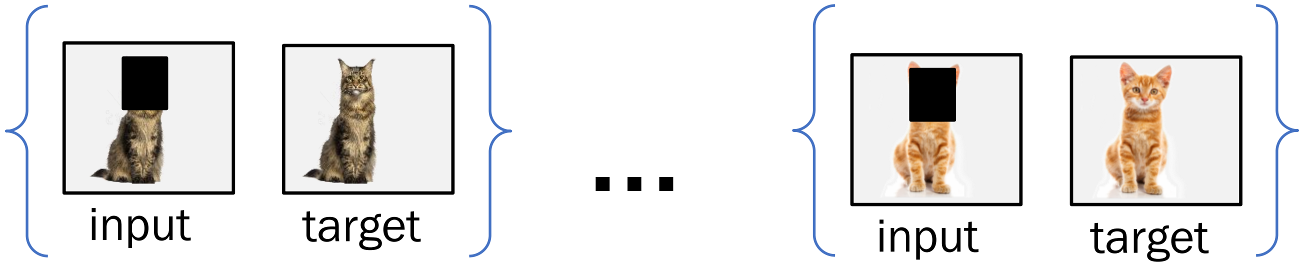
Learning approach

Idea: use training pairs of signals and measurements to directly learn the inversion function



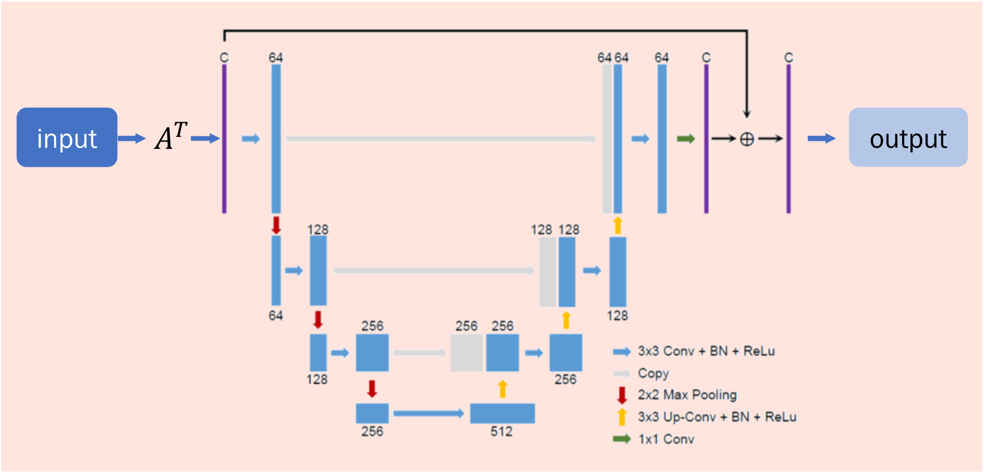
$$\underset{f}{\operatorname{argmin}}\ \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{x} - f(\boldsymbol{y})\|^2$$

supervised dataset

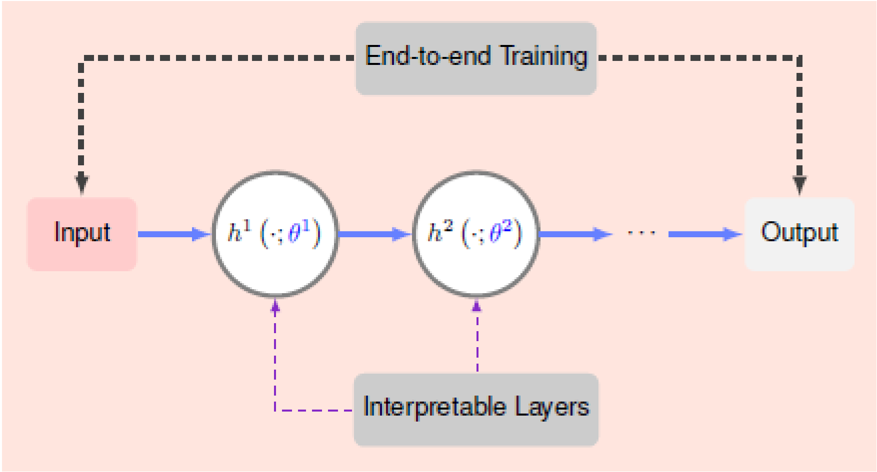


Learning approach

Many DNN architecture choices, e.g.



Back projected U-Net: $\hat{\boldsymbol{x}} = f(A^\top \boldsymbol{y})$, e.g. [Jin, 2017]



Unrolled networks: $\hat{\boldsymbol{x}} = f(\boldsymbol{y}, A)$, e.g. [Monga, 2020]

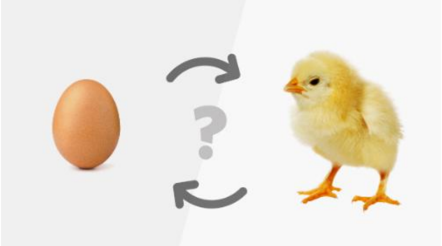
But also DnCNNs, DRUNet, SCUNet, DEQ, restormer, SwinIR, DiffPIR…

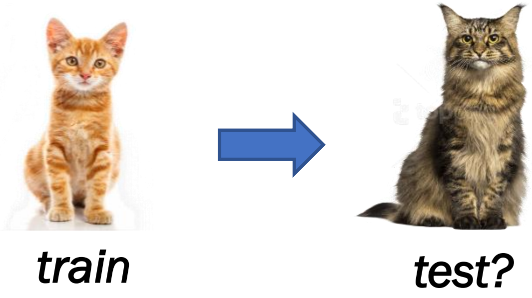
Here our focus will be on learning that is typically **architecture agnostic**

Learning approach

Main disadvantage: reference data can be expensive or impossible to get.

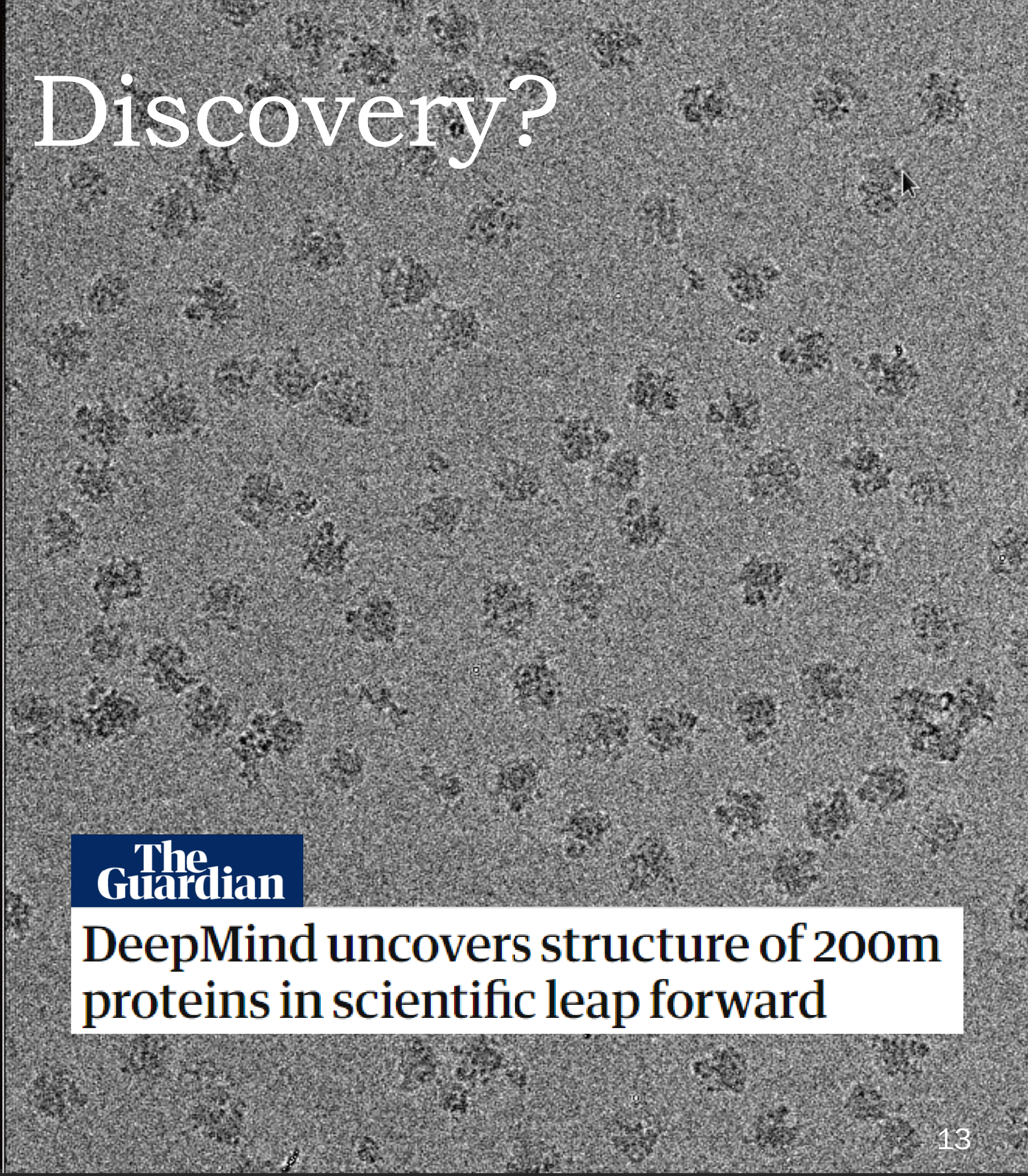
- Medical and scientific imaging
- Problems which we already 'solved'
- Distribution shift





- **Raises the question**:
  Can AI be used for data-driven knowledge discovery in imaging?

AI for Knowledge Discovery?



Learning vs Inductive Bias

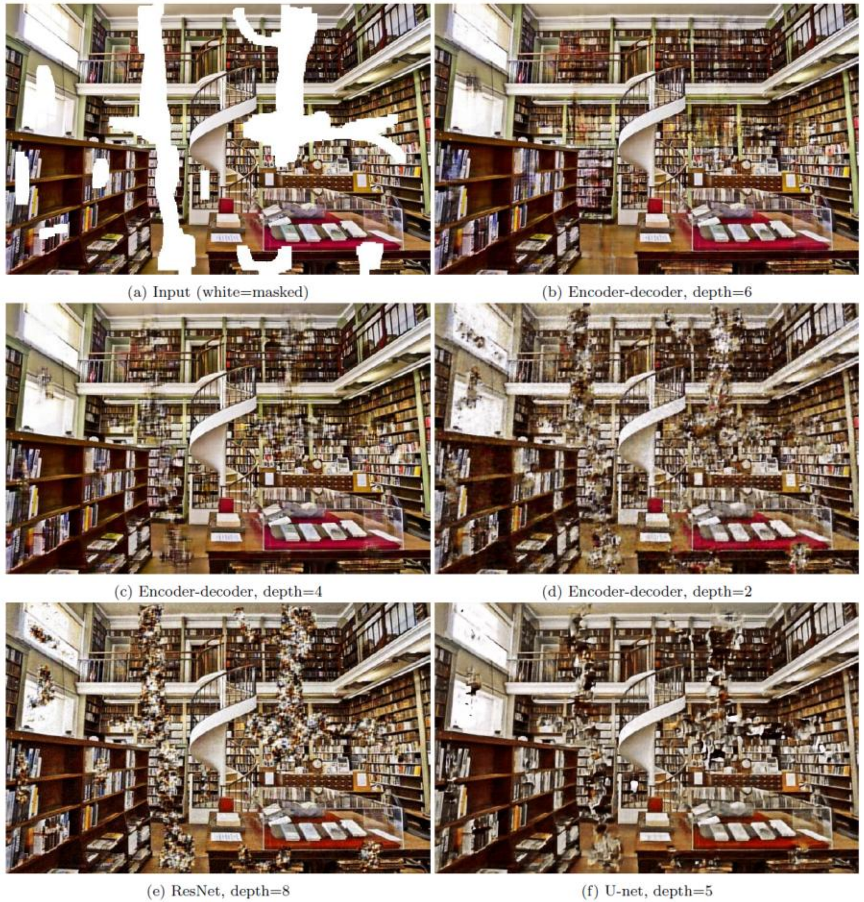
Inductive bias: all networks carry inductive bias

Deep Image Prior [Ulyanov 2018]

- Architecture *implicitly* encodes "preferred" images
- Minimise data consistency with early stopping

$$\underset{f}{\operatorname{argmin}} \|\boldsymbol{y} - Af(\boldsymbol{y})\|^2$$

Hard to categorise:
- Inductive bias?
- Learning non-local structure [Tachella 2021]?



Deep Image Prior [Ulyanov 2018]

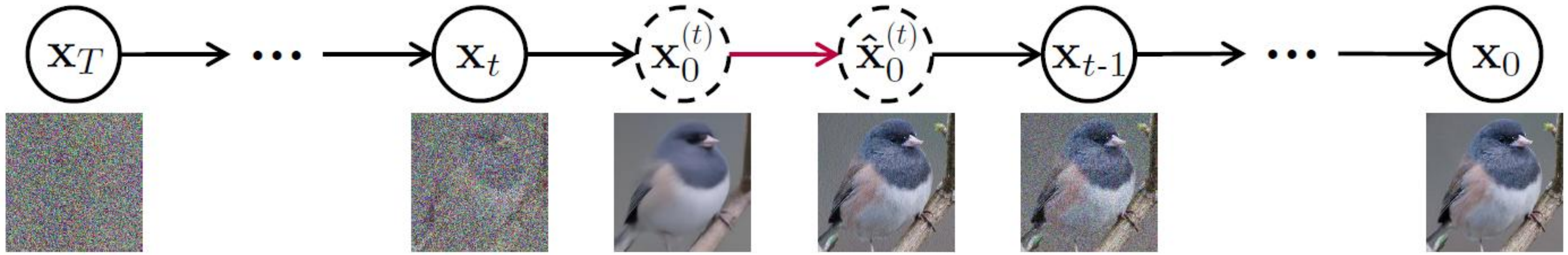
Learning vs Pre-Trained

Exploiting Deep Pre-Trained Denoisers: Exploit powerful pretrained DNN denoisers, $\mathrm{D}(\boldsymbol{u}, \sigma)$, e.g., [Zhu, 2023]

Plug and Play methods: e.g. PnP proximal gradient descent

$$\boldsymbol{u}_{k+1} = \mathrm{D}(\boldsymbol{x}_k - \gamma A^{\mathrm{T}}(A\boldsymbol{x}_k - \boldsymbol{y}), \sigma)$$

Conditional Diffusion models: use pre-trained denoiser in reverse SDE to attempt to sample conditional distribution



Generally pre-trained but can leverage self-supervised denoisers (see part III)

References

The full reference list for this tutorial can be found here:

https://tachella.github.io/projects/selfsuptutorial/