Self Supervised Learning Methods for Imaging

Escuela de Primavera de Deep Learning, Buenos Aires, Argentina

Julián Tachella, CNRS, École Normale Supérieure de Lyon

The Inverse problem

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

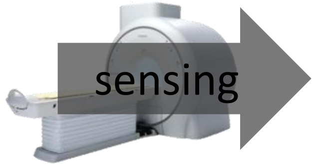
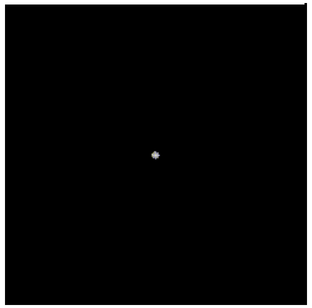
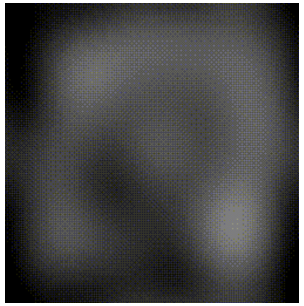
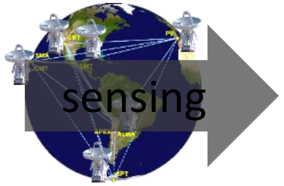
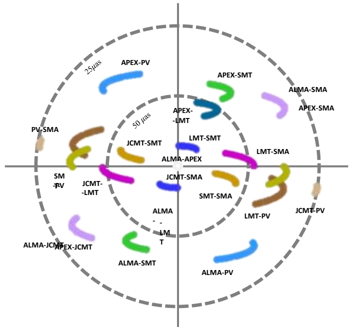
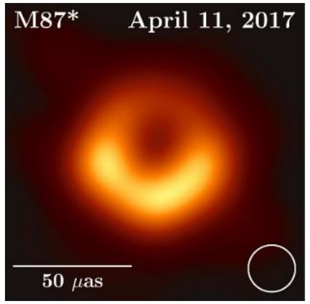
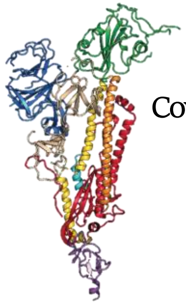
signal $\in \mathbb{R}^n$

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

Physics

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

Examples

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

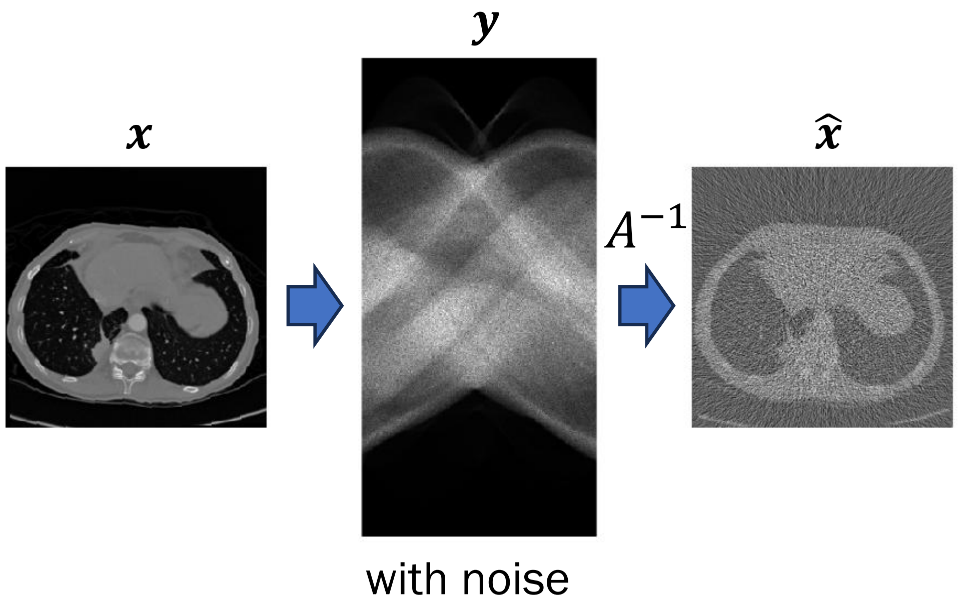
Why it is hard to invert?

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



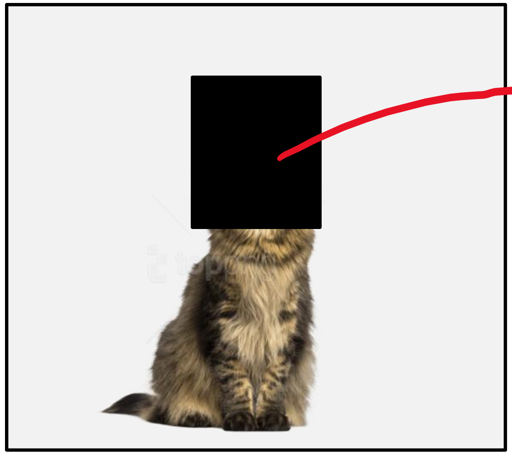
Why it is hard to invert?

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

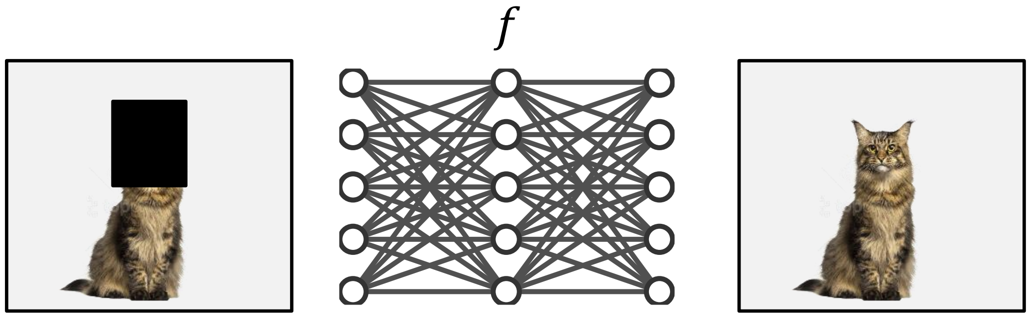


NULLSPACE OF A

reconstruct

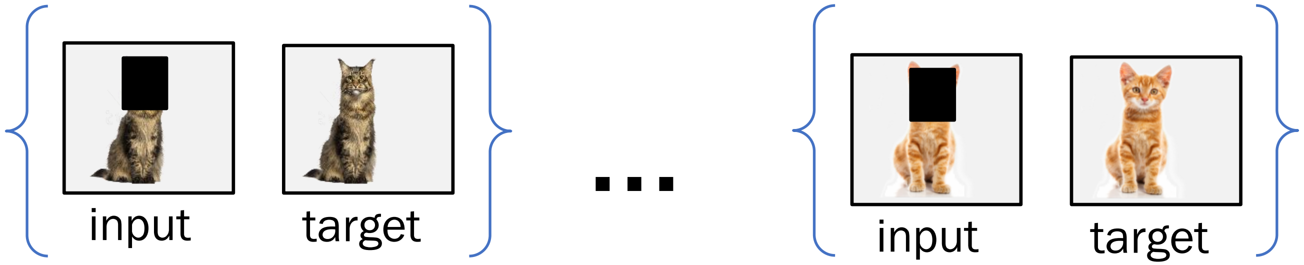
Learning approach

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



$$\underset{f}{\operatorname{argmin}} \sum_{i=1}^{N} || f(\boldsymbol{y}_i) - \boldsymbol{x}_i ||^2$$

supervised dataset



Learning approach

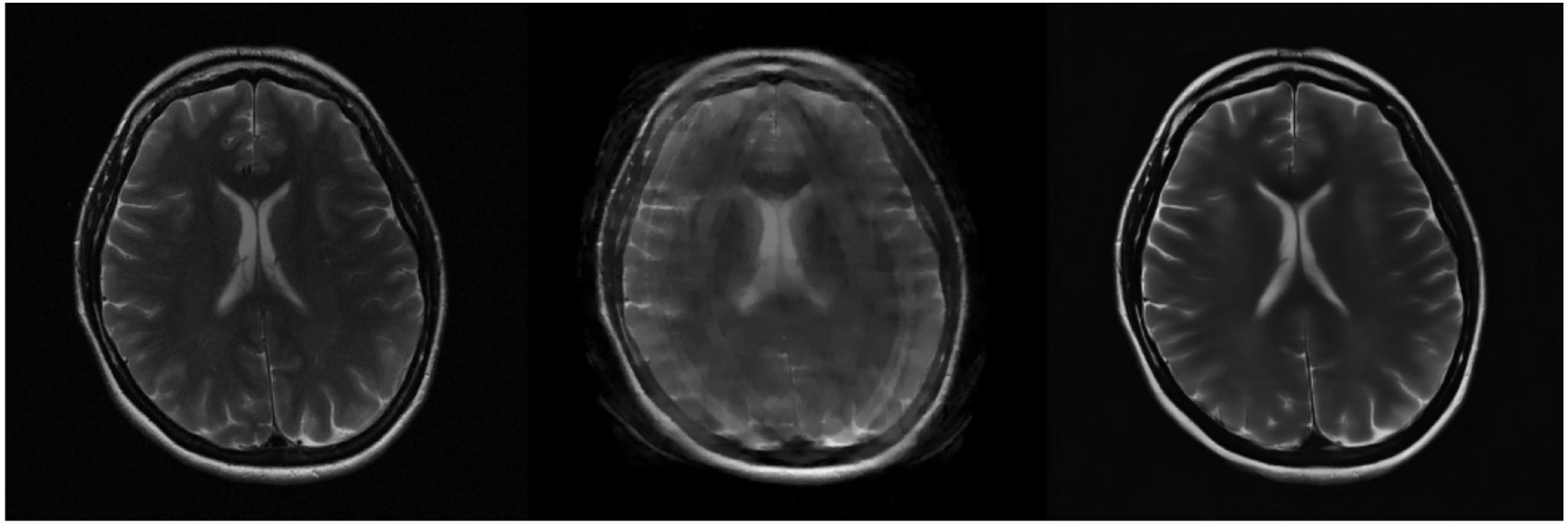
Advantages:

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

fastMRI

Accelerating MR Imaging with AI

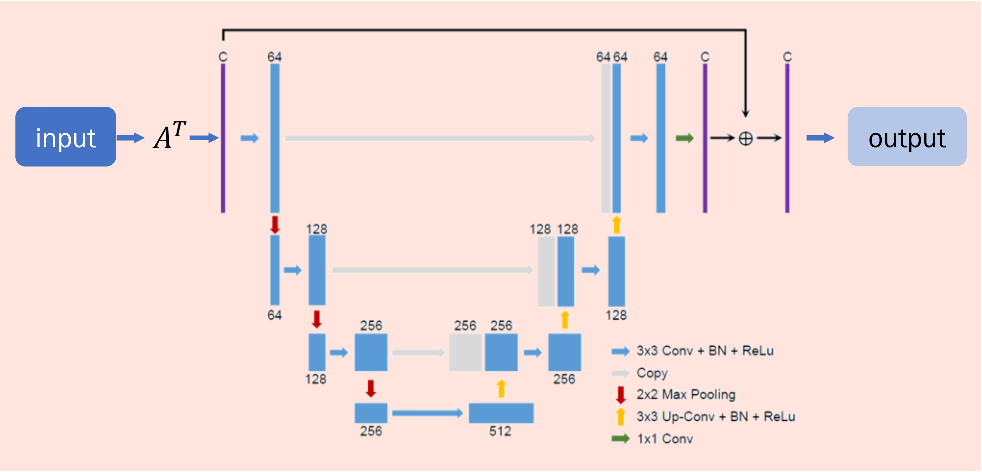
Ground-truth | Total variation (28.2 dB) | Deep network (**34.5 dB**)



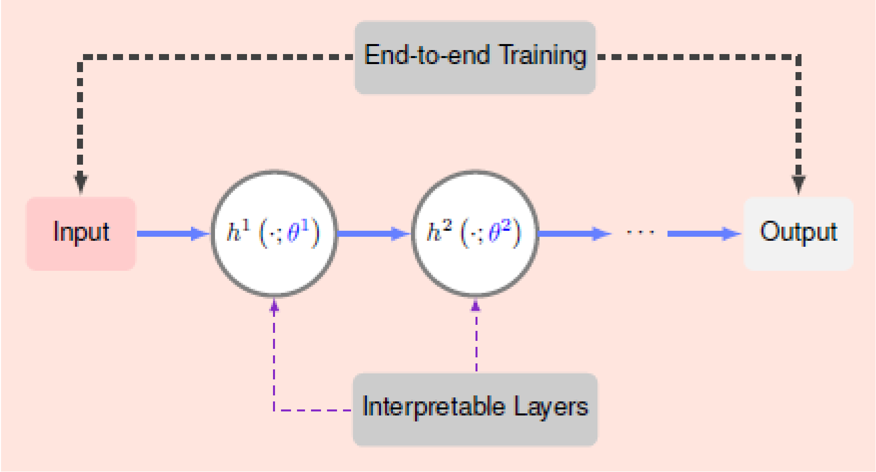
x8 accelerated MRI [Zbontar et al., 2019]

Learning approach

Many 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]

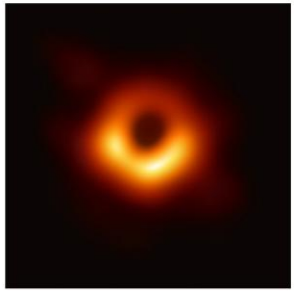
But also DnCNNs, DRUNet, SCUNet, DEQ, restormer, SwinIR, DiffPIR…

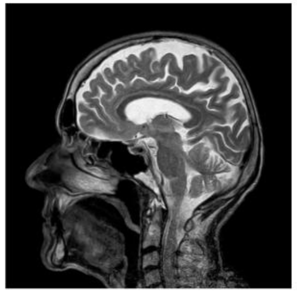
Here our focus will be on learning that is typically **architecture agnostic**

Learning approach

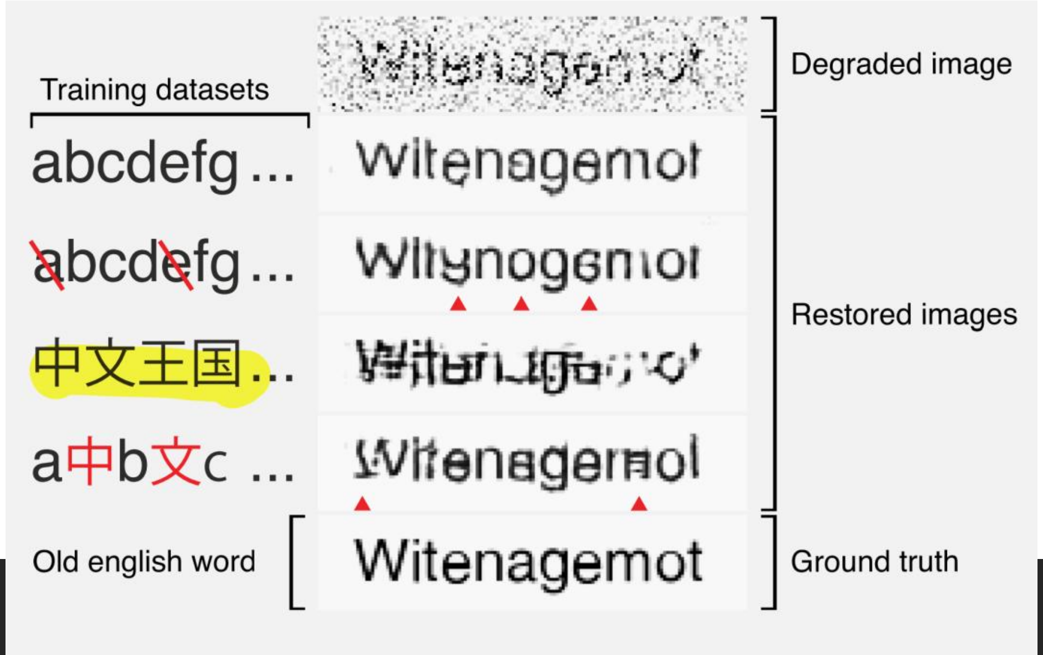
Main disadvantage: Obtaining training signals x_i can be expensive or impossible.

- Medical and scientific imaging

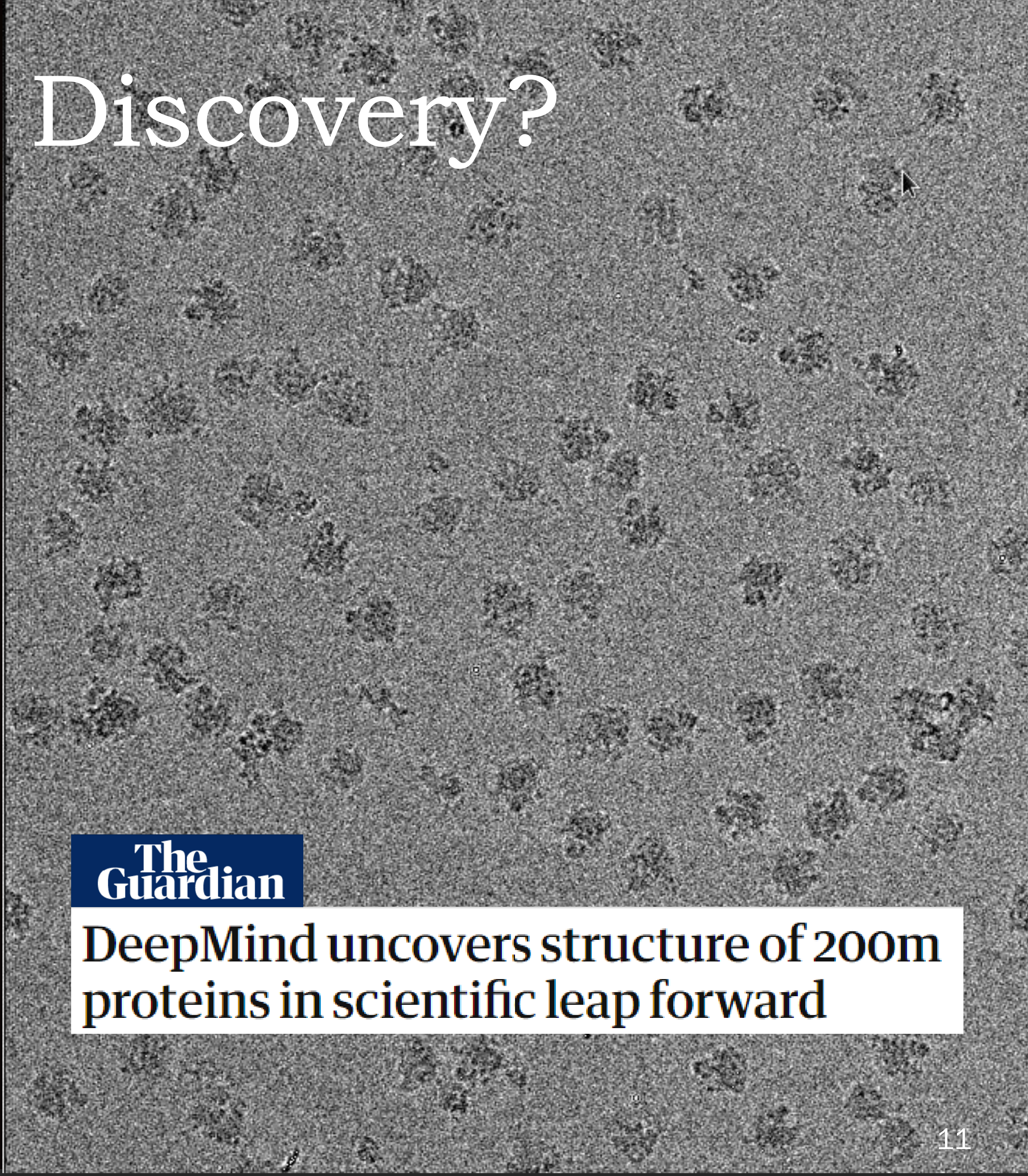




- Distribution shift [Belthangady & Royer, 2019]



AI for Knowledge Discovery?



What this talk is **not** about

Autoregressive models: LLM pretraining on an autoregressive task.

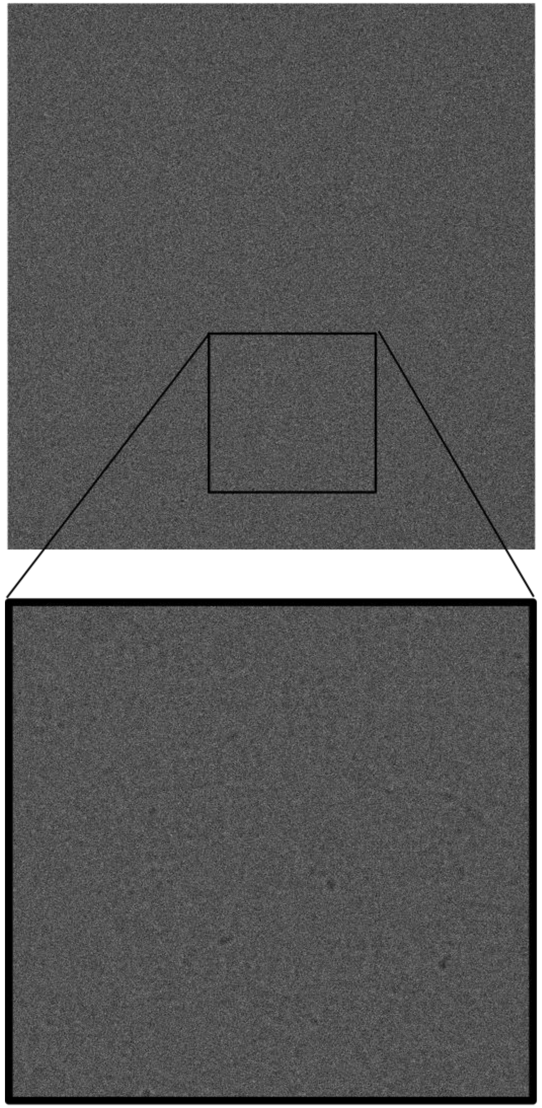
Self-supervised learning for feature learning: Sim2Sim, masked autoencoders, DINOv2,3, etc. Focus on learning features for downstream tasks.

Diffusion models and PnP: require pre-trained denoiser/scores with ground-truth data

- The methods presented can be used to train denoisers without ground-truth!

Purpose of this talk

How can we learn f from measurement $\{\boldsymbol{y}_i\}_{i=1}^{N}$ data alone?



1. Noisy: $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon}$

2. Incomplete and noisy: $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{\epsilon}$

$$\boxed{\underset{f}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(y_i, f)}$$

Best we can expect

We focus on ℓ_2 loss and minimum mean squared error estimators (MMSE)

$$f^* = \arg\min_f \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \, ||\boldsymbol{x} - f(\boldsymbol{y})||^2$$

$$\Longrightarrow \quad f^*(\boldsymbol{y}) = \mathbb{E}\{\boldsymbol{x}|\boldsymbol{y}\}$$

- Other estimators might be preferred, eg. perceptual [Blau and Michaeli, 2018]

Self-supervised learning

Approximating the supervised loss:

1. Unbiased estimator

$$\mathbb{E}_{\boldsymbol{y}}\, \mathcal{L}(\boldsymbol{y}, f) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} || f(\boldsymbol{y}) - \boldsymbol{x}||^2$$

2. Same minimizer

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

$$f^*(\boldsymbol{y}) = \mathbb{E}\{\boldsymbol{x}|\boldsymbol{y}\}$$

3. Unbiased estimator under constraints

$\mathbb{E}_{\boldsymbol{y}}\, \mathcal{L}(\boldsymbol{y}, f) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} || f(\boldsymbol{y}) - \boldsymbol{x}||^2$ for *certain* $f \neq \mathbb{E}\{\boldsymbol{x}|\boldsymbol{y}\}$

Part 2: Learning from noisy data

Denoising problems

In this part, we will focus on ‘denoising’ problems

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

where $A \in \mathbb{R}^{m \times n}$ is invertible (and thus $m \geq n$).

- We focus on $\boldsymbol{A} = \boldsymbol{I}$ for simplicity.
- All methods in this part can be extended to any invertible $\boldsymbol{A}$.

Self-supervised risk estimators

Supervised loss

$$\mathcal{L}_{\text{sup}}(\boldsymbol{x},\boldsymbol{y},f) = ||\boldsymbol{x}-f(\boldsymbol{y})||^2 = \underbrace{||\boldsymbol{y}-f(\boldsymbol{y})||^2}_{\text{Measurement consistency}} + \underbrace{2f(\boldsymbol{y})^\top(\boldsymbol{y}-\boldsymbol{x})}_{\substack{\text{key term to approximate!} \\ = \boldsymbol{f}(\boldsymbol{y})^\top\boldsymbol{\epsilon}}} + \text{const.}$$

Naïve loss doesn't work!

$$\mathcal{L}_{\text{MC}}(\boldsymbol{y},f) = ||\boldsymbol{y}-f(\boldsymbol{y})||^2$$

$$\Rightarrow f^*(\boldsymbol{y}) = \boldsymbol{y}$$

Noise2Noise

Mallows $\boldsymbol{C_p}$ [Mallows, 1973], **Noise2Noise** [Lehtinen, 2018]

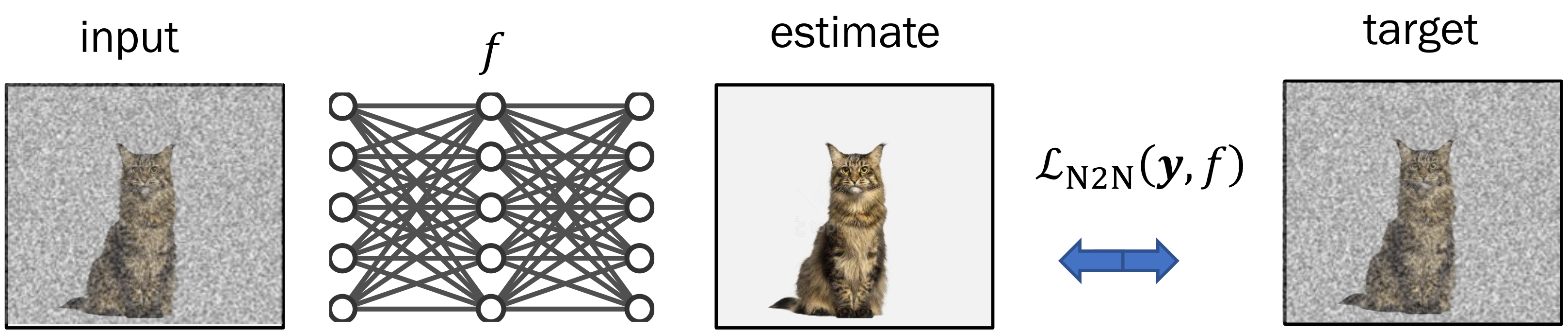
- **Independent** pairs $\boldsymbol{y}_a = \boldsymbol{x} + \boldsymbol{\epsilon}_a$ and $\boldsymbol{y}_b = \boldsymbol{x} + \boldsymbol{\epsilon}_b$ with $\boldsymbol{\epsilon}_a, \boldsymbol{\epsilon}_b$ **independent**
- $\mathbb{E}_{\boldsymbol{\epsilon_b}|\boldsymbol{x}}\boldsymbol{\epsilon_b} = \boldsymbol{0}$

$$\boxed{\mathcal{L}_{\text{Noise2Noise}}(\boldsymbol{y}_a, \boldsymbol{y}_b, f) = ||\, \boldsymbol{y}_b - f(\boldsymbol{y}_a)||^2}$$

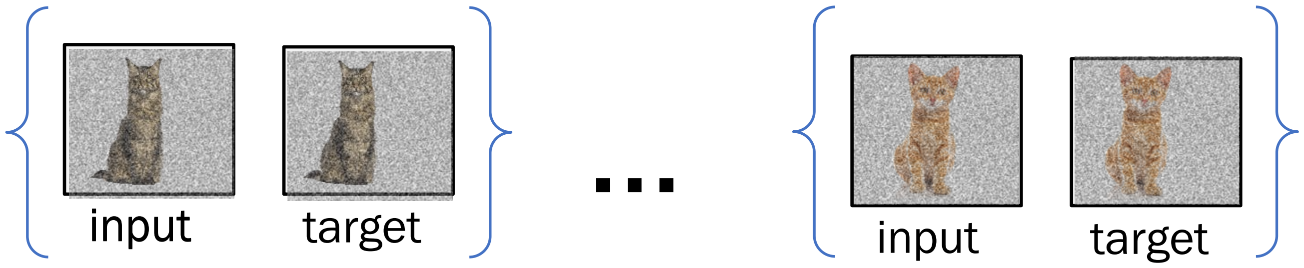
$$\mathbb{E}_{\boldsymbol{y_b}|\boldsymbol{x}}\, f(\boldsymbol{y}_a)^\top(\boldsymbol{y_b} - \boldsymbol{x}) = f(\boldsymbol{x} + \boldsymbol{\epsilon}_a)\, \underbrace{\mathbb{E}\, \boldsymbol{\epsilon}_b}_{=0} = 0$$

- Also works for any noise distribution with $\mathbb{E}_{\boldsymbol{y_b}|\boldsymbol{x}}\, \boldsymbol{y_b} = \boldsymbol{x}$

Noise2Noise



Dataset of noisy pairs



Not useful for the microscopy example!

Recorrupted2Recorrupted

Recorrupted2Recorrupted [Pang et al., 2021], **Coupled Bootstrap** [Oliveira et al., 2022], **Noisier2Noise** [Moran et al., 2020].

Proposition: Let $\boldsymbol{y} \sim N\left(\boldsymbol{x}, I\sigma^2\right)$ and define

$$\boldsymbol{y}_a = \boldsymbol{y} + \sqrt{\frac{1-\alpha}{\alpha}}\boldsymbol{\omega}$$

$$\boldsymbol{y}_b = \frac{\boldsymbol{y}}{\alpha} - \frac{\boldsymbol{y}_a(1-\alpha)}{\alpha}$$

where $\boldsymbol{\omega} \sim N\left(\boldsymbol{0}, I\sigma^2\right)$ and $\alpha \in \mathbb{R}$, then $\boldsymbol{y}_a$ and $\boldsymbol{y}_b$ are **independent** random variables (fixed $\boldsymbol{x}$).

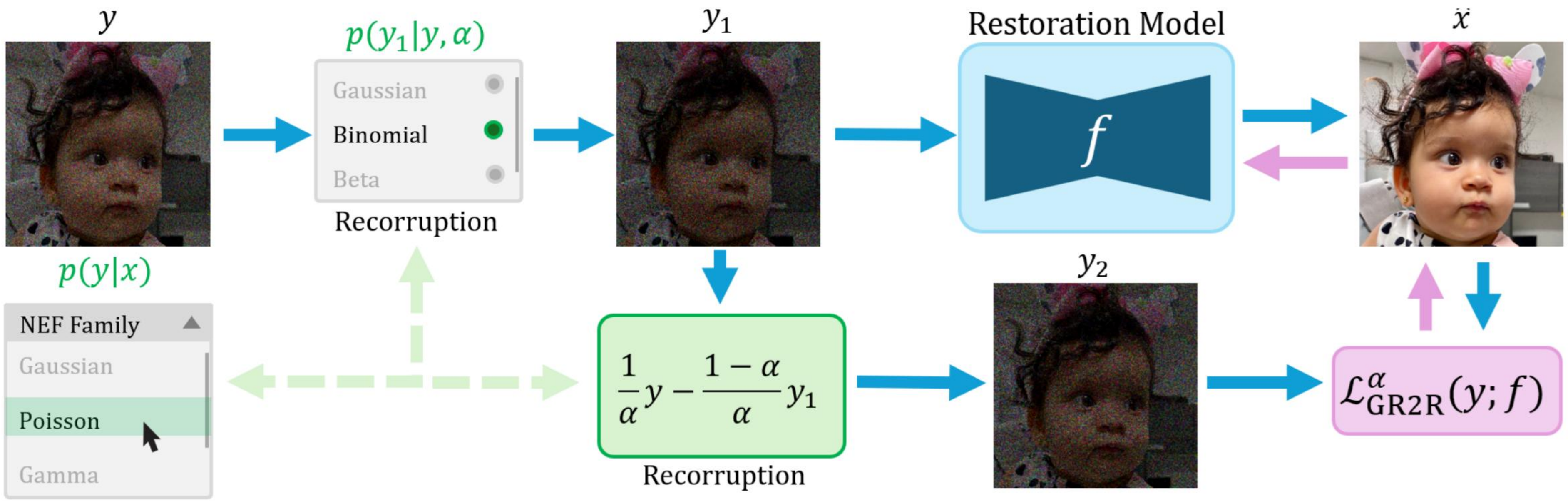
$$\boxed{\mathcal{L}_{R2R}(\boldsymbol{y}, f) = \mathbb{E}_{\boldsymbol{\omega}} \|\, \boldsymbol{y}_b - f(\boldsymbol{y}_a)\, \|^2}$$

- Price to pay: SNR($\boldsymbol{y}_a$) < SNR($\boldsymbol{y}$)
- At **test time**, $f^{\text{test}}(\boldsymbol{y}) = \frac{1}{N}\sum_i f\left(\boldsymbol{y} + \sqrt{\frac{1-\alpha}{\alpha}}\boldsymbol{\omega}_i\right)$ with $\boldsymbol{\omega}_i \sim \mathcal{N}(\boldsymbol{0}, I\sigma^2)$

Recorrupted2Recorrupted

- Can be extended to other noise distributions [Monroy, Bacca and Tachella, CVPR 2025]

Model	**Gaussian** $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\Sigma})$	**Poisson** $\boldsymbol{z} \sim \mathcal{P}(\boldsymbol{x}/\gamma), \boldsymbol{y} = \gamma \boldsymbol{z}$	**Gamma** $\boldsymbol{y} \sim \mathcal{G}(\ell, \ell/\boldsymbol{x})$
$\boldsymbol{y}_1$	$\boldsymbol{y}_1 = \boldsymbol{y} + \sqrt{\frac{\alpha}{1-\alpha}}\boldsymbol{\omega}$, $\boldsymbol{\omega} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$	$\boldsymbol{y}_1 = \frac{\boldsymbol{y} - \gamma\boldsymbol{\omega}}{1-\alpha}$, $\boldsymbol{\omega} \sim \mathrm{Bin}(\boldsymbol{z}, \alpha)$	$\boldsymbol{y}_1 = \boldsymbol{y} \circ (\boldsymbol{1} - \boldsymbol{\omega})/(1-\alpha)$, $\boldsymbol{\omega} \sim \mathrm{Beta}(\ell\alpha, \ell(1-\alpha))$



Stein's Unbiased Risk Estimator

- **Stein's lemma** [Stein 1974] **:** Let $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}, I\sigma^2)$, f be weakly differentiable, then

$$\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}}\, (\boldsymbol{y}-\boldsymbol{x})^\top f(\boldsymbol{y}) = \mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}}\, \sigma^2 \sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y})$$

$$\boxed{\mathcal{L}_{\text{SURE}}(\boldsymbol{y}, f) = \underbrace{||\, \boldsymbol{y} - f(\boldsymbol{y})||^2}_{\text{Measurement consistency}} + \underbrace{2\sigma^2 \sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y})}_{\text{Degrees of freedom [Efron, 2004]}}}$$

- **Hudson's lemma** [Hudson 1978] extends this result for the exponential family (eg. **Poisson Noise**)
- Beyond exponential family: **Poisson-Gaussian noise** [Le Montagner et al., 2014]

[Raphan and Simoncelli, 2011]

Stein's Unbiased Risk Estimator

Monte Carlo SURE [Efron 1975, Breiman 1992, Ramani et al., 2007]

SURE's divergence is generally approximated as

$$\sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) \approx \frac{\boldsymbol{\omega}^\top}{\alpha} \left(f(\boldsymbol{y}) - f(\boldsymbol{y} + \boldsymbol{\omega}\alpha)\right)$$

where $\alpha > 0$ small, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, I)$

- Noisier2Noise is equivalent to SURE when $\alpha \to 0$ [Monroy Bacca and Tachella, CVPR 2025].

Stein's Unbiased Risk Estimator

The solution to SURE is **Tweedie's Formula**

$$\arg\min_{f} \ \mathbb{E}_{\boldsymbol{y}} || \boldsymbol{y} - f(\boldsymbol{y}) ||^2 + 2\sigma^2 \sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y})$$

Integration by parts

$$\arg\min_{f} \ \mathbb{E}_{\boldsymbol{y}} || \boldsymbol{y} - f(\boldsymbol{y}) ||^2 - 2\sigma^2 \sum_i f(\boldsymbol{y}) \frac{\delta \log p_{\boldsymbol{y}}(\boldsymbol{y})}{\delta y_i}$$

Complete squares

$$\arg\min_{f} \ \mathbb{E}_{\boldsymbol{y}} || f(\boldsymbol{y}) - \boldsymbol{y} - \sigma^2 \nabla \log p_{\boldsymbol{y}}(\boldsymbol{y}) ||^2$$

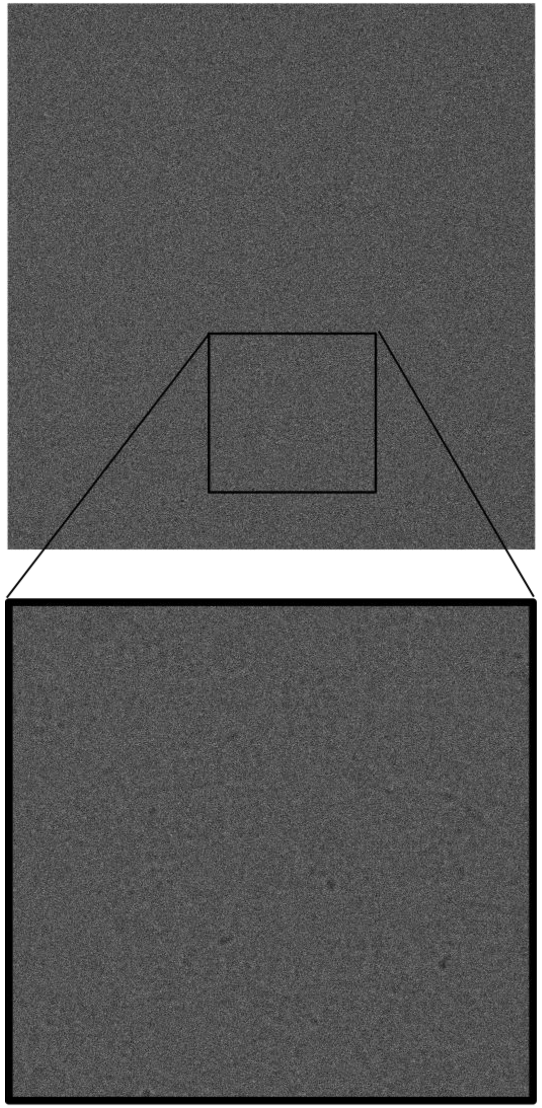
$$\Rightarrow \quad f(\boldsymbol{y}) = \boldsymbol{y} + \sigma^2 \nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})$$

- **Noise2Score** [Kim and Ye, 2021] learns $\nabla \log p_{\boldsymbol{y}}(\boldsymbol{y})$ from noisy data + denoises with Tweedie.
- Key formula behind diffusion models, which can be trained self-supervised [Daras et al., 2024]

Summary So Far

	Train Eval	Test Eval	Single y	MMSE optimal	Unknown noise
Noise2Noise	1	1		✓	✓
R2R	1	>1	✓	✓	
SURE	2	1	✓	✓	

If we have a single y and don't know the noise distribution?



UNSURE

Assumption: Let $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{x}, I\sigma^2\right)$, σ^2 unknown

UNSURE [Tachella et al., ICLR 2025]

$$\boxed{\mathcal{L}_{\mathrm{UNSURE}}(\boldsymbol{y}, f) = ||\,\boldsymbol{y} - f(\boldsymbol{y})||^2 \text{ subject to } \mathbb{E}_{\boldsymbol{y}} \textstyle\sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) = 0}$$

- SURE's perspective: $\mathcal{L}_{\mathrm{SURE}}(\boldsymbol{y}, f) = ||\,\boldsymbol{y} - f(\boldsymbol{y})||^2 + 2\sigma^2 \cancel{\sum_i \frac{\delta f_i}{\delta y_i}}(\boldsymbol{y})$

- Not MMSE optimal (but almost)

UNSURE

- In practice, we use Lagrange multipliers

$$\min_{f} \max_{\eta} \mathbb{E}_{\boldsymbol{y}} \, || \, \boldsymbol{y} - f(\boldsymbol{y})||^2 + 2\eta \sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y})$$

$$\Rightarrow \quad f^*(\boldsymbol{y}) = \boldsymbol{y} + \hat{\eta} \, \nabla \log p_{\boldsymbol{y}}(\boldsymbol{y}) \qquad \hat{\eta} = \left(\frac{1}{n} \mathbb{E}_{\boldsymbol{y}} \, ||\nabla \log p_{\boldsymbol{y}}(\boldsymbol{y}) \, ||^2\right)^{-1}$$

- Expected error

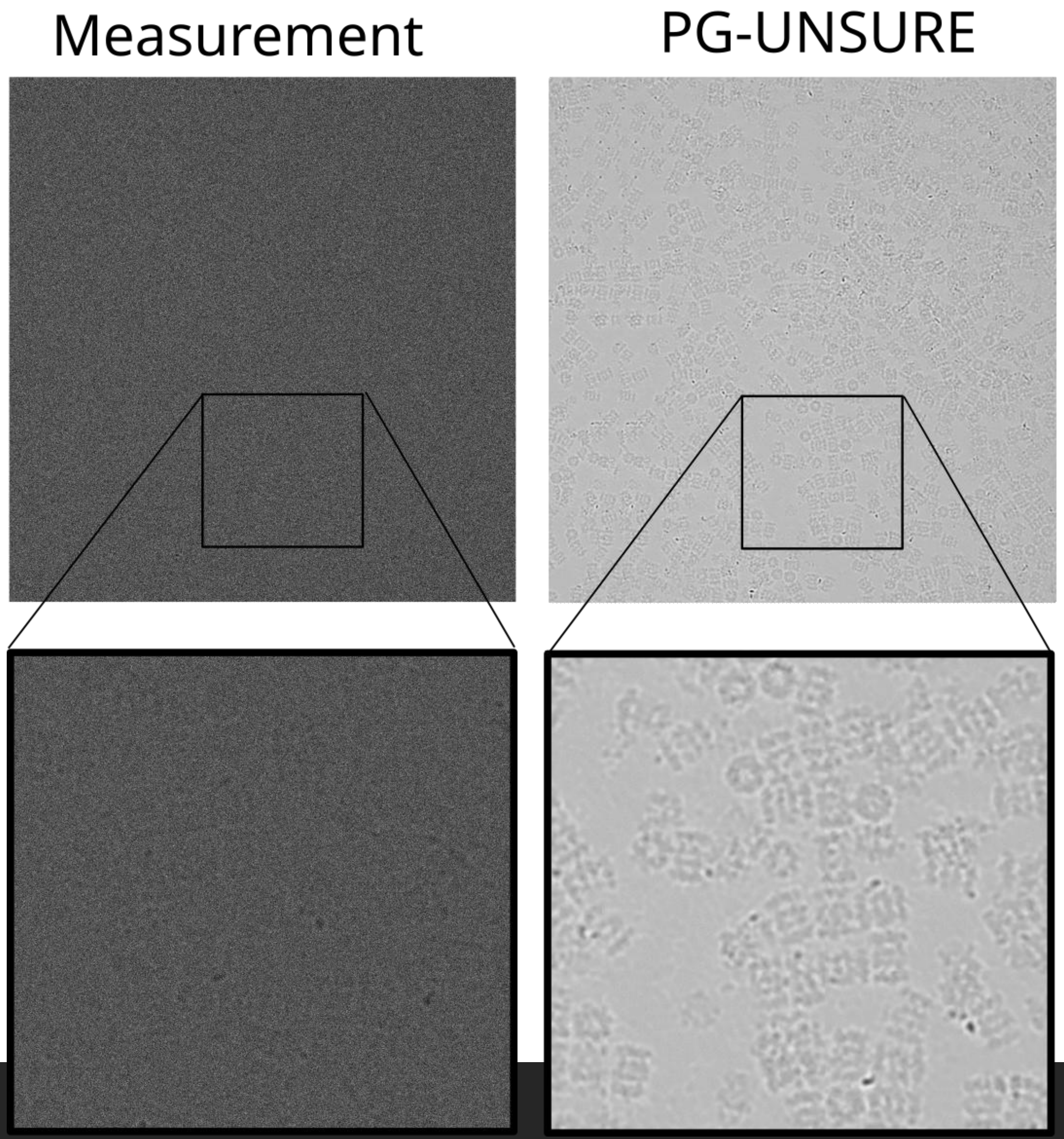
$$\frac{1}{n} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} || \, f^*(\boldsymbol{y}) - \boldsymbol{x} \, || = \sigma^2 \left(\frac{1}{1 - \frac{\mathrm{MMSE}}{\sigma^2}} - 1 \right) \approx \mathrm{MMSE} + \frac{\mathrm{MMSE}^2}{\sigma^2}$$

- UNSURE can be extended to **unknown noise covariance** and **Poisson Gaussian noise**

Experiments

Real data experiments

- Cryo electron microscopy images
- Extremely low SNR
- Approx. Poisson-Gaussian noise



Cross-Validation Methods

Assumption: f_i does not depend on y_i, that is $\frac{\delta f_i}{\delta y_i} = 0$. Decomposable noise $p(\boldsymbol{y}|\boldsymbol{x}) = \prod p(y_i|x_i)$

$$\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}} \sum_{i=1}^{n} f_i(\boldsymbol{y})(y_i - x_i) = \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{y}_{-i}|\boldsymbol{x}}\, f_i(\boldsymbol{y}_{-i})\, \overbrace{\mathbb{E}_{y_i|x_i}\,(y_i - x_i)}^{=0} = 0$$

$$\boxed{\mathcal{L}_{CV}(\boldsymbol{y}, f) = \;||\, \boldsymbol{y} - f(\boldsymbol{y})||^2 \text{ subject to } \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) = 0\ \forall i, \boldsymbol{y}}$$

- SURE's perspective: $\mathcal{L}_{SURE}(\boldsymbol{y}, f) = ||\, \boldsymbol{y} - f(\boldsymbol{y})||^2 + 2\sigma^2 \cancel{\sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y})}$

- These methods are not MMSE optimal
- How to remove dependence on y_i: training or architecture

Measurement Splitting

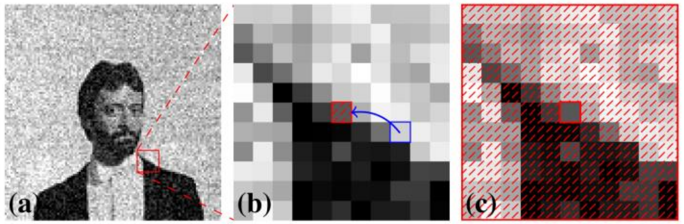
Cross-validation [Efron, 2004]: random split $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_a \\ \boldsymbol{y}_b \end{bmatrix}$ at each iteration

$$\boxed{\mathcal{L}_{N2V}(\boldsymbol{y}, f) = \mathbb{E}_{a,b} || \, \boldsymbol{y}_b - \operatorname{diag} \boldsymbol{m}_b \, f(\boldsymbol{y}_a) ||^2}$$

where $\boldsymbol{m}_b \in \{0,1\}^n$ masks out the pixels in $\boldsymbol{y}_a$.

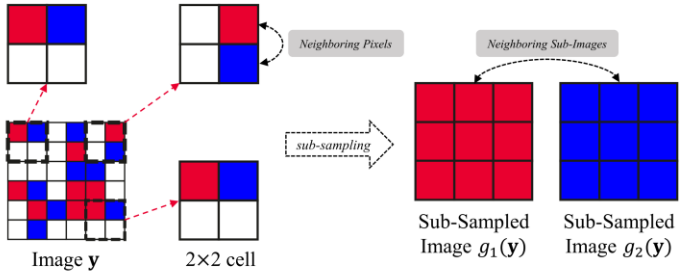
Noise2Void [Krull et al., 2019], **Noise2Self** [Batson, 2019]

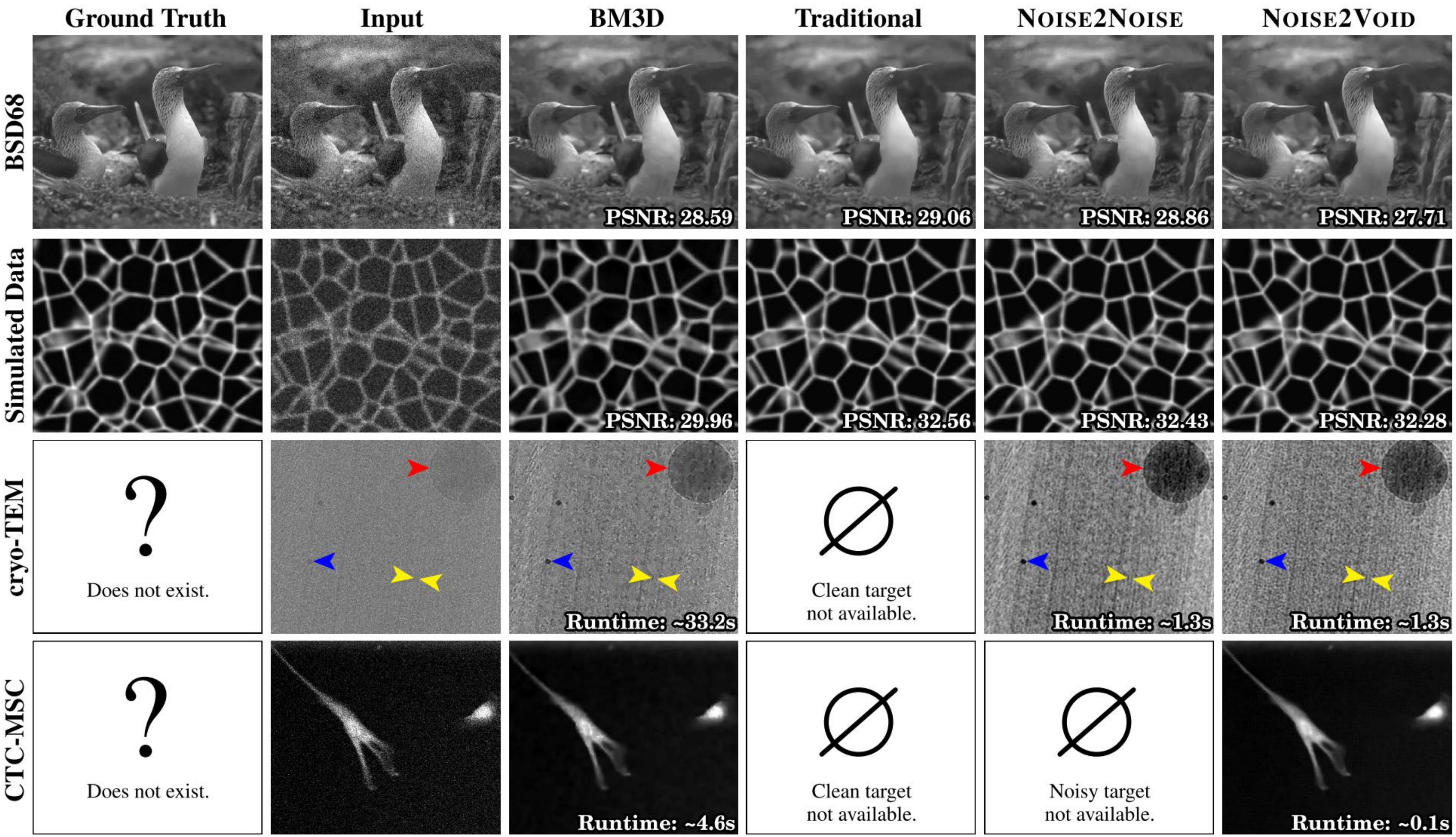
- During training flip centre pixel
- Computes loss only on flipped pixels



Neighbor2Neighbor [Huang, 2023]

- Use different subsampling as input and target
- Assumes scale invariance




Ground Truth
Input
BM3D
Traditional
NOISE2NOISE
NOISE2VOID
BSD68
PSNR: 28.59
PSNR: 29.06
PSNR: 28.86
PSNR: 27.71
Simulated Data
PSNR: 29.96
PSNR: 32.56
PSNR: 32.43
PSNR: 32.28
cryo-TEM
?
Does not exist.
Runtime: ~33.2s
Clean target not available.
Runtime: ~1.3s
Runtime: ~1.3s
CTC-MSC
?
Does not exist.
Runtime: ~4.6s
Clean target not available.
Noisy target not available.
Runtime: ~0.1s

Measurement Splitting

At **test time**, $f(\boldsymbol{y})$ is evaluated as

1. Test f as trained (expensive)

$f^{\text{test}}(\boldsymbol{y}) = \frac{1}{N}\sum_i M\, f\left(\boldsymbol{y}_{\boldsymbol{a}_i}\right)$ with $\boldsymbol{y}_{\boldsymbol{a}_i} \sim p(\boldsymbol{y}_{\boldsymbol{a}}|\boldsymbol{y})$ and $M = \left(\sum_i^N \operatorname{diag}(\boldsymbol{m}_{b,i})\right)^{-1}$

2. Assume good generalization of f (cheap)

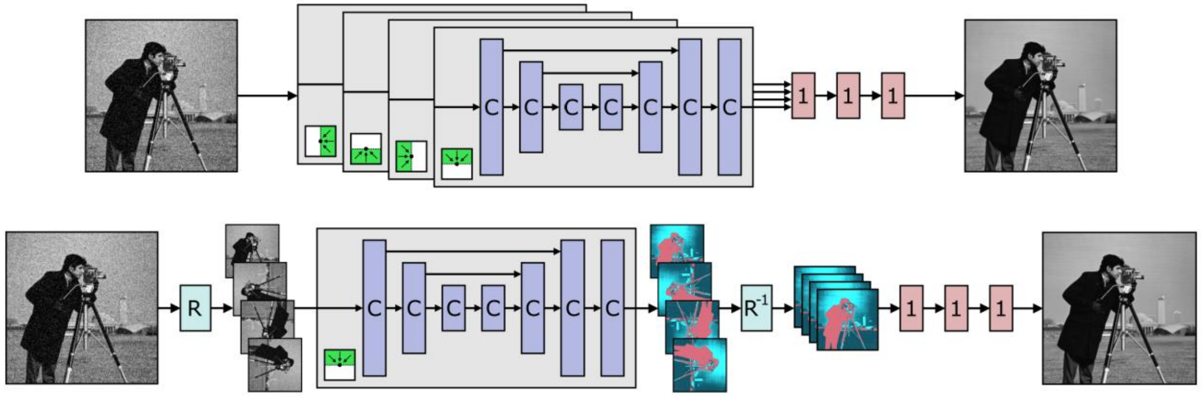
- $f^{\text{test}}(\boldsymbol{y}) = f(\boldsymbol{y}_{\boldsymbol{a}})$ with $\boldsymbol{y}_{\boldsymbol{a}} \sim p(\boldsymbol{y}_{\boldsymbol{a}}|\boldsymbol{y})$
- $f^{\text{test}}(\boldsymbol{y}) = f(\boldsymbol{y})$

Blind Spot Networks

Blind spot networks [Laine et al., 2019], [Lee et al., 2022]

- Convolutional architecture that doesn't 'see' centre pixel by construction

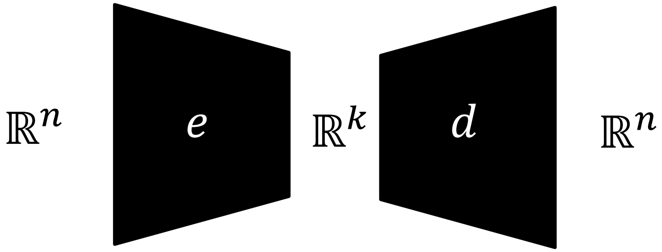
$$\boxed{\mathcal{L}_{\mathrm{BS}}(\boldsymbol{y}, f_{\mathrm{BS}}) = ||\,\boldsymbol{y} - f_{\mathrm{BS}}(\boldsymbol{y})||^2}$$



Autoencoders

Autoencoders

$f_{\mathrm{AE}} = d \cdot e$



$\sum_i \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) \approx O(k) \ll n$ is small

Assume

- f has a strong bottleneck

$$\boxed{\mathcal{L}_{\mathrm{AE}}(\boldsymbol{y}, f) = ||\, \boldsymbol{y} - f_{\mathrm{AE}}(\boldsymbol{y})||^2}$$

- Noise distribution is ‘high-dimensional’ whereas signal distribution is ‘low-dimensional’

Summary

	Train Eval	Test Eval	Single y	MMSE optimal	Unknown separable noise	Unknown coloured noise
Noise2Noise	1	1		✓	✓	✓
R2R	1	>1	✓	✓		
SURE	2	1	✓	✓		
UNSURE	2	1	✓	✓	✓	✓
Noise2Void	1	1	✓		✓	
Blind Spot	1	>1	✓		✓	
Autoencoders	1	1			✓	✓

No free lunch: less assumptions about noise = less optimal estimator

Summary

SURE, R2R

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

known σ

UNSURE

$$\sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{y}} \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) = 0$$

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

unknown σ

$$\mathbb{E}_{\boldsymbol{y}} \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) = 0$$

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon}$$

unknown $\boldsymbol{\sigma}$

$$\mathbb{E}_{\boldsymbol{y}} \frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) = 0$$

$$\mathbb{E}_{\boldsymbol{y}}\, y_i \frac{\delta f_i}{\delta y_i} = 0$$

$$\boldsymbol{y} = \gamma \mathcal{P}(\frac{\boldsymbol{x}}{\gamma}) + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon}$$

unknown $\boldsymbol{\sigma}, \gamma$

Noise2Self, Noise2Inverse, BlindSpot Net

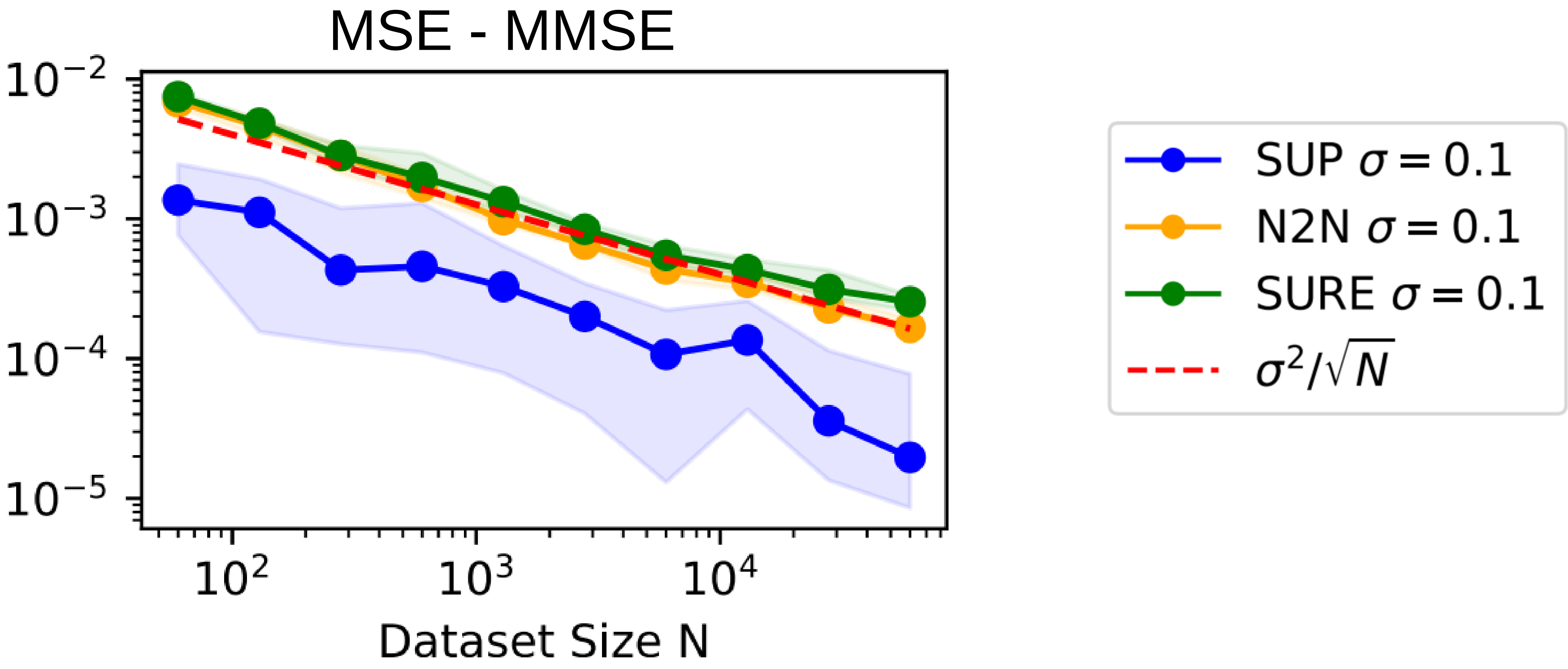
$$\frac{\delta f_i}{\delta y_i}(\boldsymbol{y}) = 0 \ \forall \boldsymbol{y}$$

$$\boldsymbol{y}|\boldsymbol{x} \sim \prod_{i=1}^{n} p_i(y_i|x_i), \ \mathbb{E}\, \boldsymbol{y}|\boldsymbol{x} = \boldsymbol{x}$$

unknown p_i

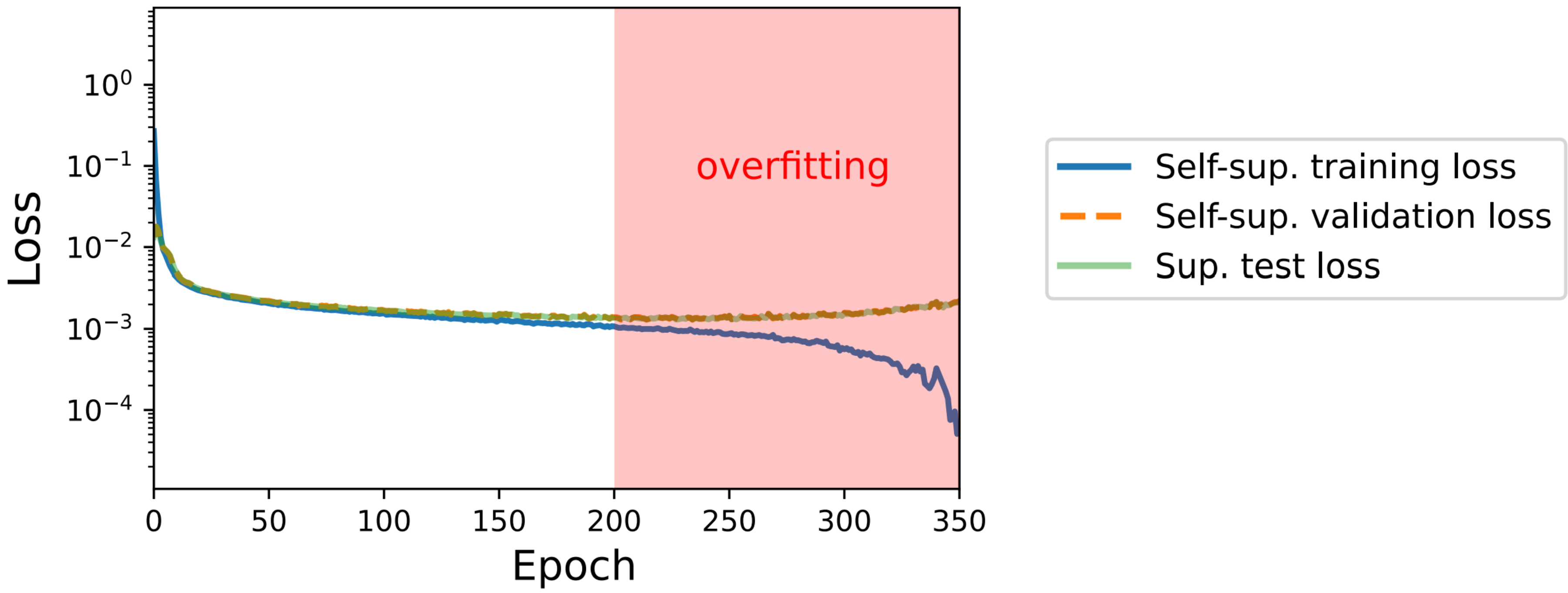
Less expressive estimators

Sample Complexity



Self-supervised validation

- Follow the same validation practices in supervised learning

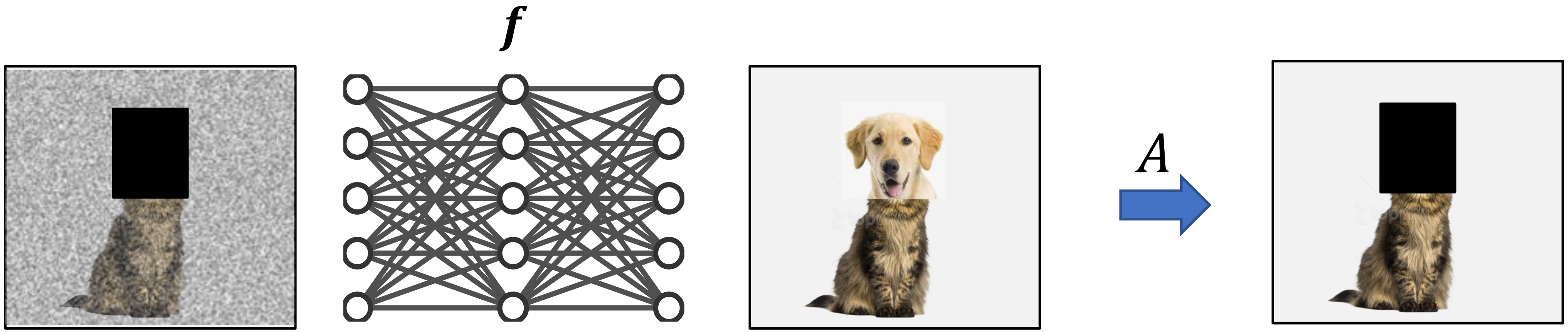


Incomplete Measurements?

For $A \neq I$, most estimators can be adapted to approximate

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

In this case, the risk does not penalise $f(\boldsymbol{y})$ in the **nullspace** of A!

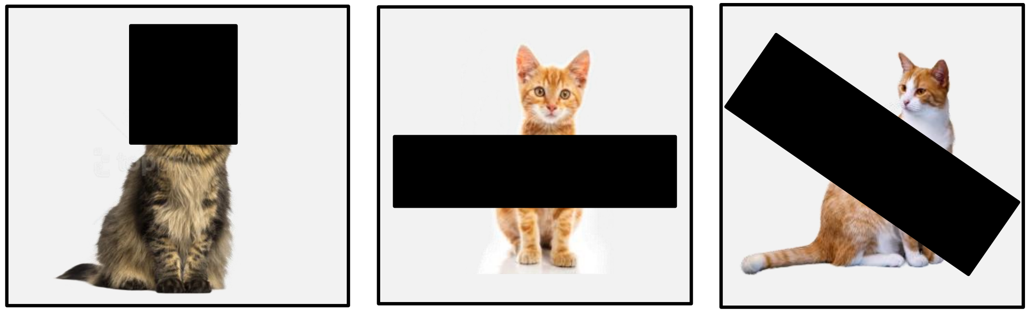


Learning from Measurements

How to learn from only $\boldsymbol{y}$?

- Access multiple operators $\boldsymbol{y}_i = A_{g_i}\boldsymbol{x}_i$ with $g \in \{1, \dots, G\}$
- Each A_g with different nullspace
- Offers the possibility for learning using multiple measurement operators

$A_1 x_1$ $A_2 x_2$ $A_3 x_3$



$A_1 x_1$ $A_2 x_2$ $A_3 x_3$

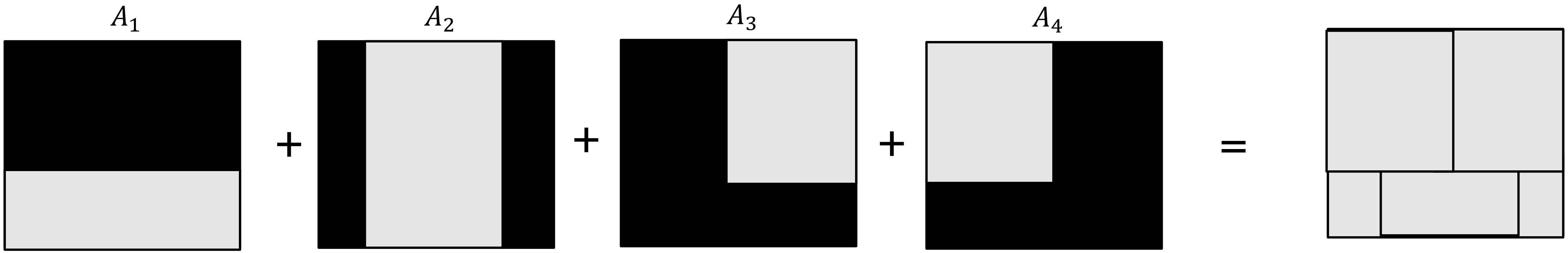
Necessary Condition

Intuition: we need that the operators $A_1, A_2, \ldots A_G$ cover the whole ambient space [T., 2022].

Proposition: Learning reconstruction mapping f from observed measurements possible only if

$$\operatorname{rank}\left(\mathbb{E}_g\, A_g^\top A_g\right) = n$$

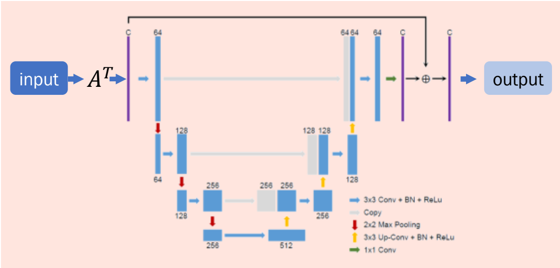
and thus, if $m \geq n/G$.

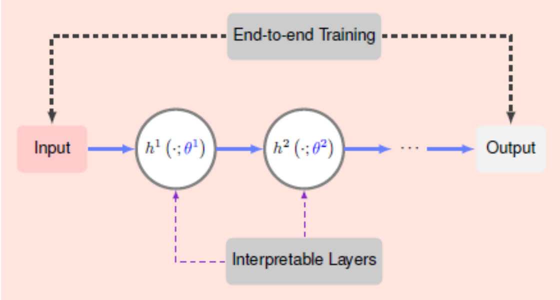


Learning Approach

We will consider networks $\hat{\boldsymbol{x}} = f(\boldsymbol{y}, A)$, where f is also a function of measurement operator e.g.,

- Filtered back projection $f(\boldsymbol{y}, A) = f(A^{\dagger}\boldsymbol{y})$
- Unrolled networks…





- Naïve **measurement consistency** loss:

$$\mathcal{L}_{\mathrm{MC}}(f) = \mathbb{E}_{\boldsymbol{y},g}\, ||\boldsymbol{y} - A_g\, f(\boldsymbol{y}, A_g)||^2$$

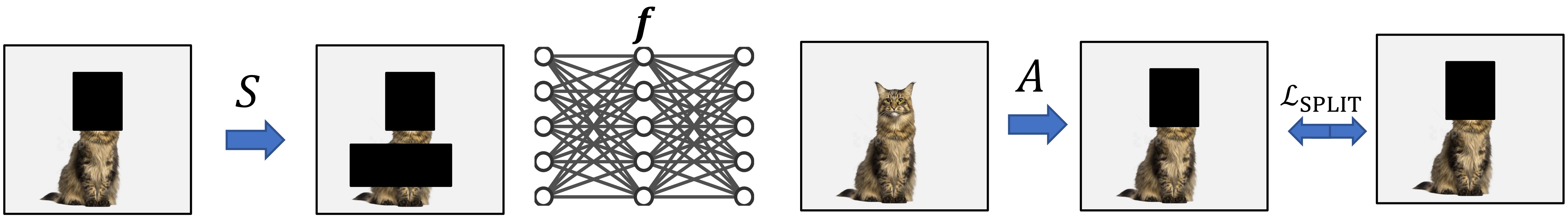
Without noise, a minimizer is the trivial solution $f(\boldsymbol{y}, A_g) = A_g^{\dagger}\boldsymbol{y}, \forall g$

Measurement Splitting Revisited

Self-supervised learning via data undersampling (SSDU) [Yaman et al., 2019]

Assume clean measurements, sample additional mask $S \sim p(S \mid A_g)$ and mask inputs

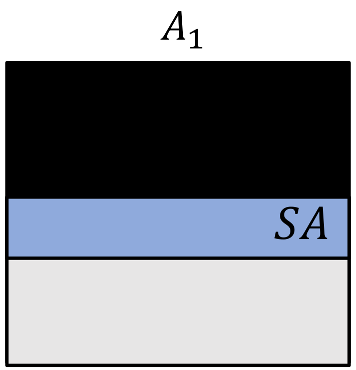
$$\mathcal{L}_{\text{SPLIT}}(\boldsymbol{y}, f) = \mathbb{E}_S || \, \boldsymbol{y} - A \, f(S\boldsymbol{y}, SA)||^2$$



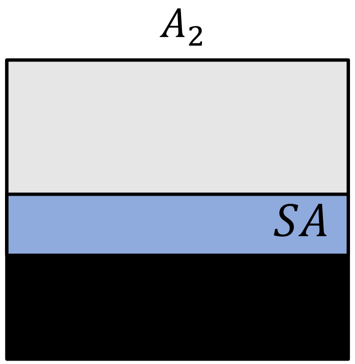
Measurement Splitting Revisited

Theorem [Daras et al. 2024]. If $\mathbb{E}\{A^\top A|SA\}$ has full rank, it has same minimizer as supervised loss, $f^*(S\boldsymbol{y}, SA) = \mathbb{E}\{\boldsymbol{x}|S\boldsymbol{y}, SA\}$

Inpainting example with $G = 2$ operators



$f_1^*(\boldsymbol{y}) = \mathbb{E}\{A_1\boldsymbol{x}|S\boldsymbol{y}, SA\}$



$f_2^*(\boldsymbol{y}) = \mathbb{E}\{A_2\boldsymbol{x}|S\boldsymbol{y}, SA\}$

$$\boxed{f^*(\boldsymbol{y}) = f_1^*(\boldsymbol{y}) + f_2^*(\boldsymbol{y}) = \mathbb{E}\{\boldsymbol{x}|S\boldsymbol{y}, SA\}}$$

Measurement Splitting Revisited

What happens if we have **noisy data?**

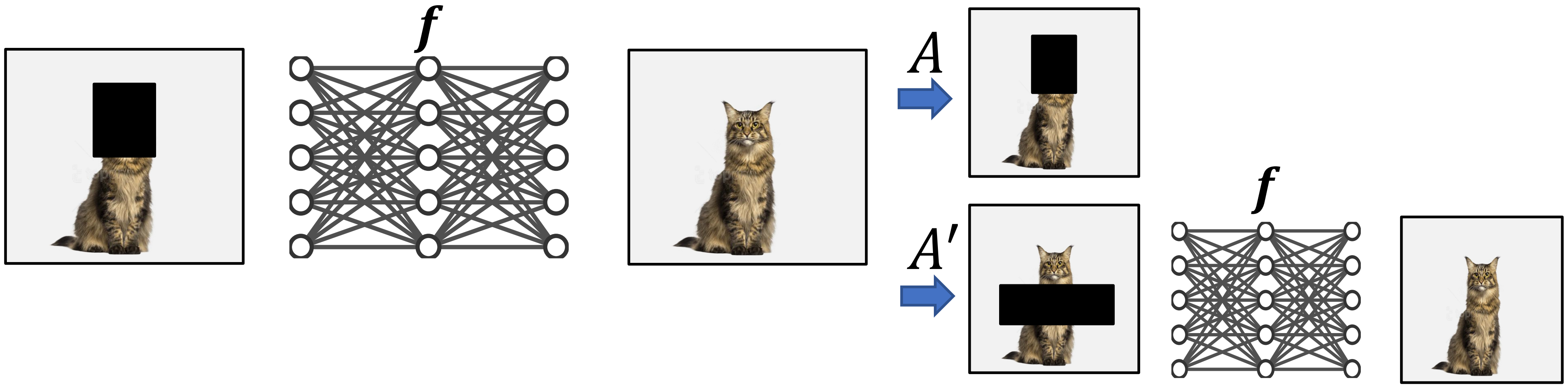
$$\mathcal{L}_{\text{SPLIT}}(\boldsymbol{y}, f) = \mathbb{E}_S || \, \boldsymbol{y} - A \, f(S\boldsymbol{y}, SA)||^2$$

$$= \mathbb{E}_S || \, \underbrace{S\boldsymbol{y} - SA \, f(S\boldsymbol{y}, SA)}_{\text{Can be replaced by SURE, R2R, etc.}}||^2 + ||\underbrace{(I-S)\boldsymbol{y} - (I-S)A \, f(S\boldsymbol{y}, SA)}_{\text{Not affected by separable noise}}||^2$$

Using All Measurements

Can we use all measurements?
Multi Operator Imaging (MOI) [Tachella et al., NeurIPS 2022]

$$\mathcal{L}_{\mathrm{MOI}}(\boldsymbol{y}, f) = \underbrace{\|\boldsymbol{y} - Af(\boldsymbol{y}, A)\|^2}_{\text{Replaced by SURE, R2R, etc.}} + \mathbb{E}_{A'} \underbrace{\|f(A'\hat{\boldsymbol{x}}, A') - \hat{\boldsymbol{x}}\|^2}_{\text{Enforces } f(A\boldsymbol{x}, A) \approx f(A'\boldsymbol{x}, A')} \quad \text{with } \hat{\boldsymbol{x}} = f(\boldsymbol{y}, A)$$



Magnetic Resonance Imaging

- Unrolled network
- FastMRI dataset (single coil)
- A_g are subsets of Fourier measurements (x4 downsampling)

Measurements y | Signal x | Supervised | Measurement Splitting | MOI

Inpainting

- U-Net network
- CelebA dataset
- A_g are inpainting masks

Measurements y | Signal $\boldsymbol{x}$ | Supervised | MOI

Part 4: Learning with equivariance

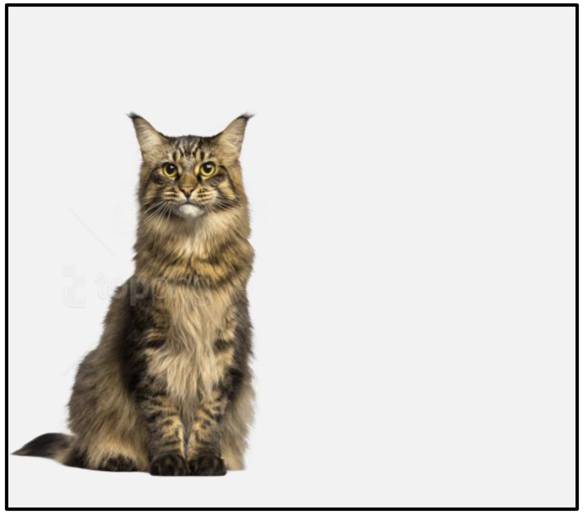
Symmetry Prior

Idea: Most natural signals sets $\mathcal{X}$ are invariant to groups of transformations.

Example: natural images are translation invariant

- Mathematically, a set $\mathcal{X}$ is invariant to $\{T_g \in \mathbb{R}^{n\times n}\}_{g\in G}$ if

$$\forall \boldsymbol{x} \in \mathcal{X},\ \ \forall g \in G,\ T_g\boldsymbol{x} \in \mathcal{X}$$



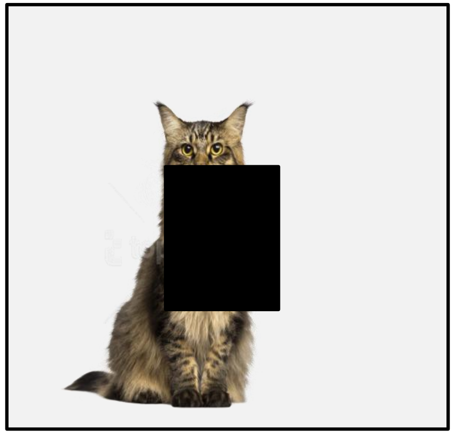
Other symmetries: rotations, permutation, amplitude

Symmetry Prior

Equivariant Imaging [Chen, Davies and Tachella, ICCV 2021]

For all $g \in G$ we have

$$\boldsymbol{y} = A\boldsymbol{x} = \underbrace{AT_g}_{A_g}\overbrace{T_g^{-1}\boldsymbol{x}}^{x'} = A_g\boldsymbol{x}'$$



- We get multiple virtual operators $\{A_g\}_{g\in G}$ ‘for free’!
- Each AT_g might have a different nullspace

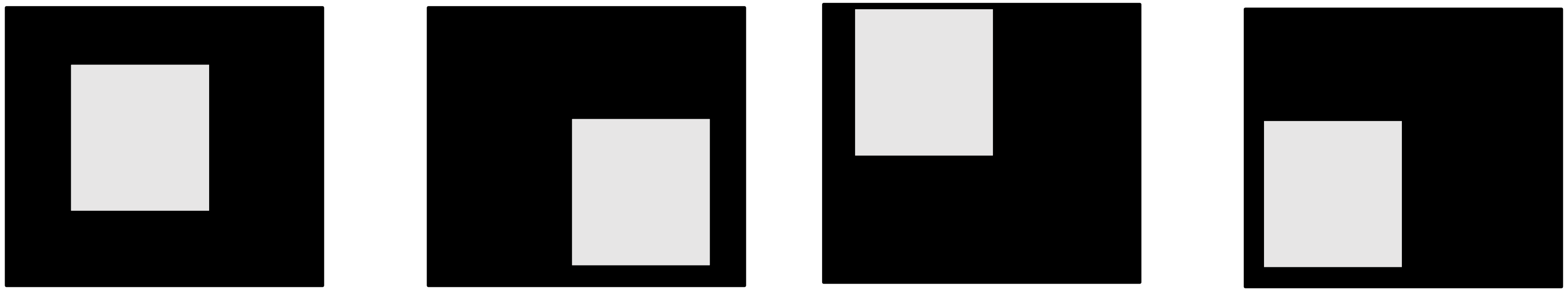
Necessary condition

Proposition [T. et al., 2023]: Learning reconstruction mapping f from observed measurements possible only if

$$\mathrm{rank}\left(\mathbb{E}_g\, T_g^\top A^\top A T_g\right) = n,$$

and thus if $m \geq \max \frac{c_j}{s_j} \geq \frac{n}{|G|}$ where s_j and c_j are dimension and multiplicity of irreps.

A_1 A_2 A_3 A_4



(Non)-Equivariant Operators

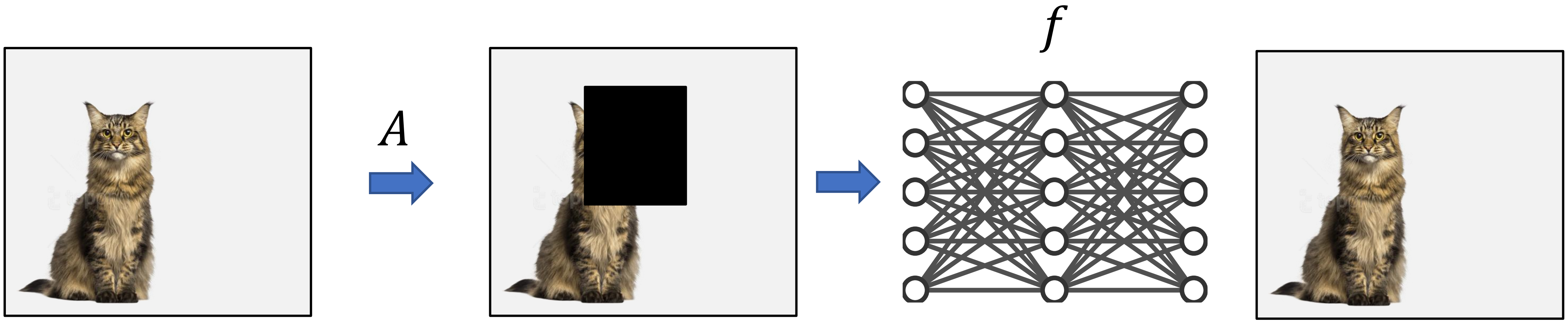
Theorem [T. et al., 2023]*:* The full rank condition requires that A **is not equivariant:** $AT_g \neq \tilde{T}_g A$

$$\operatorname{rank}\left(\mathbb{E}_g\, T_g^\top A^\top A T_g\right) = \operatorname{rank}\left(A^\top (\mathbb{E}_g \tilde{T}_g^\top \tilde{T}_g) A\right) = \operatorname{rank}\left(A^\top A\right) = m < n$$

Equivariant Imaging

How can we enforce equivariance in practice?

Idea: we should have $f\left(AT_g\boldsymbol{x}\right) = T_g f(A\boldsymbol{x})$, i.e. $f \circ A$ should be G-equivariant



Equivariant Imaging

How can we enforce equivariance in practice?

$$\boxed{\mathcal{L}_{\mathrm{EI}}(\boldsymbol{y}, f) = \mathbb{E}_g \,||\, T_g \hat{\boldsymbol{x}} - f\left(A T_g \hat{\boldsymbol{x}}\right)||^2}$$

where $\hat{\boldsymbol{x}} = f(\boldsymbol{y})$ is used as reference

Proposition [Tachella & Pereyra, *2024*]**:** *For linear and measurement consistent $Af(A\boldsymbol{x}) = A\boldsymbol{x}$ reconstruction, we have*

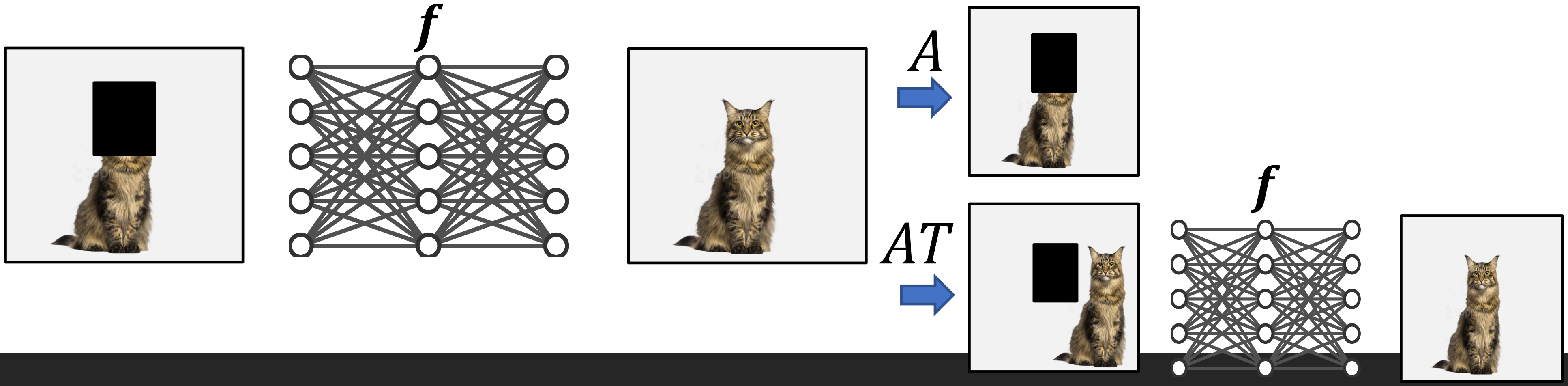
$$\mathcal{L}_{\mathrm{EI}}(\boldsymbol{y}, f) = ||\boldsymbol{x} - f(\boldsymbol{y})||^2 + bias$$

where the *bias* term is small if $f \circ A$ is **not** equivariant.

Equivariant Imaging

Robust Equivariant Imaging [Chen, Tachella and Davies, CVPR 2022]

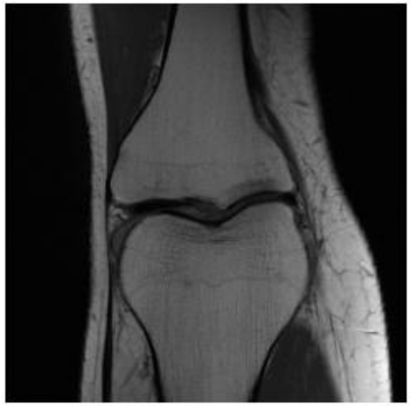
$$\mathcal{L}_{\text{REI}}(\boldsymbol{y}, f) = \underbrace{\left|\left|f(\boldsymbol{y}) - \boldsymbol{x}\right|\right|^2}_{\text{Replaced by SURE, R2R, etc}} + \underbrace{\mathcal{L}_{\text{EI}}(\boldsymbol{y}, f)}_{\text{enforces equivariance of } f \circ A}$$

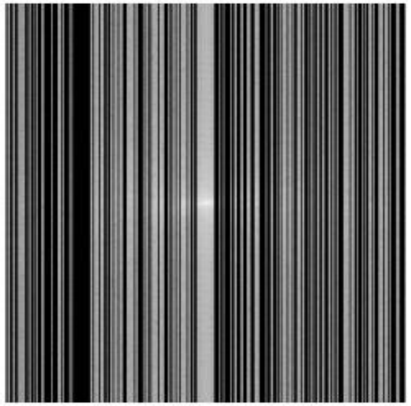


MRI

- Operator A is a subset of Fourier measurements (x2 downsampling)
- Dataset is approximately **rotation invariant**

Signal x Measurements y





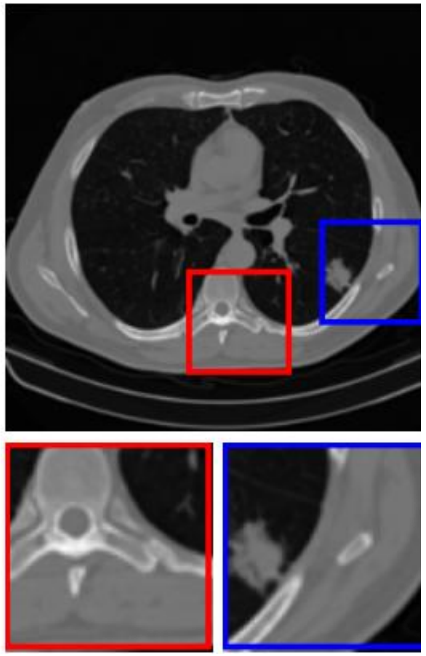
Chen, T., Davies, *CVPR* 2022

Computed Tomography

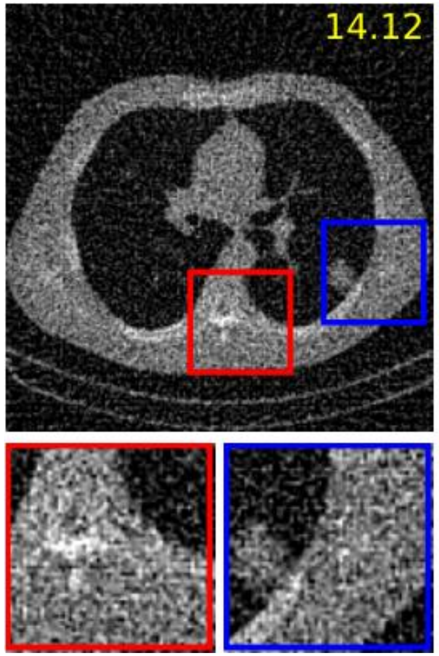
- Operator A is (non-linear variant) sparse radon transform
- Mixed Poisson-Gaussian noise
- Dataset is approximately **rotation invariant**

Noisy measurements y

Clean signal x



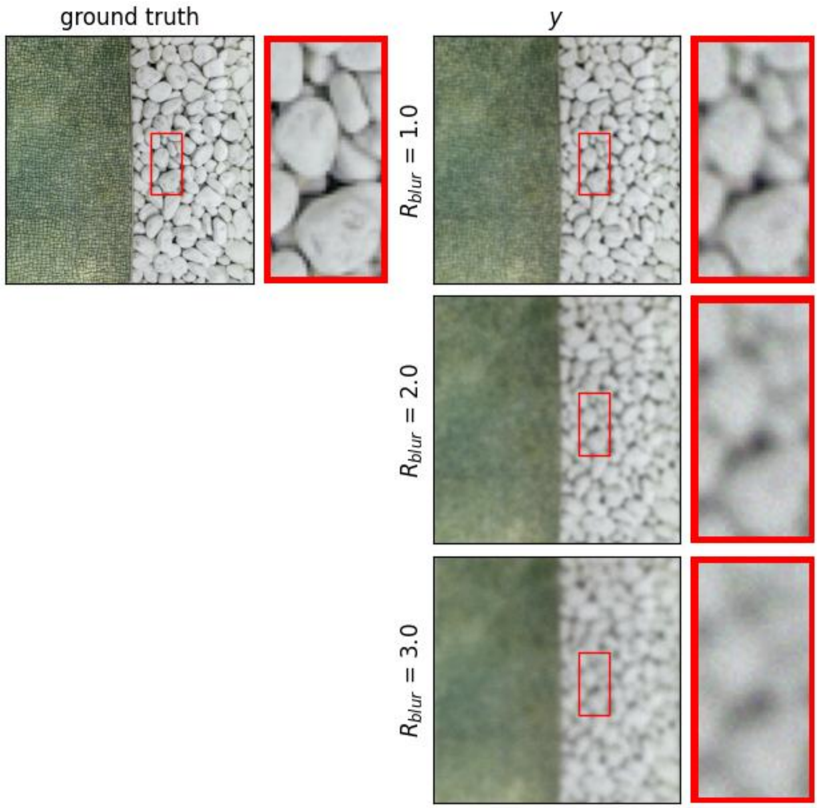
Meas. consistency



Chen, T., Davies, *CVPR* 2022

Image Deblurring

- Operator A is isotropic blur with Gaussian noise
- Dataset is approximately **scale invariant**

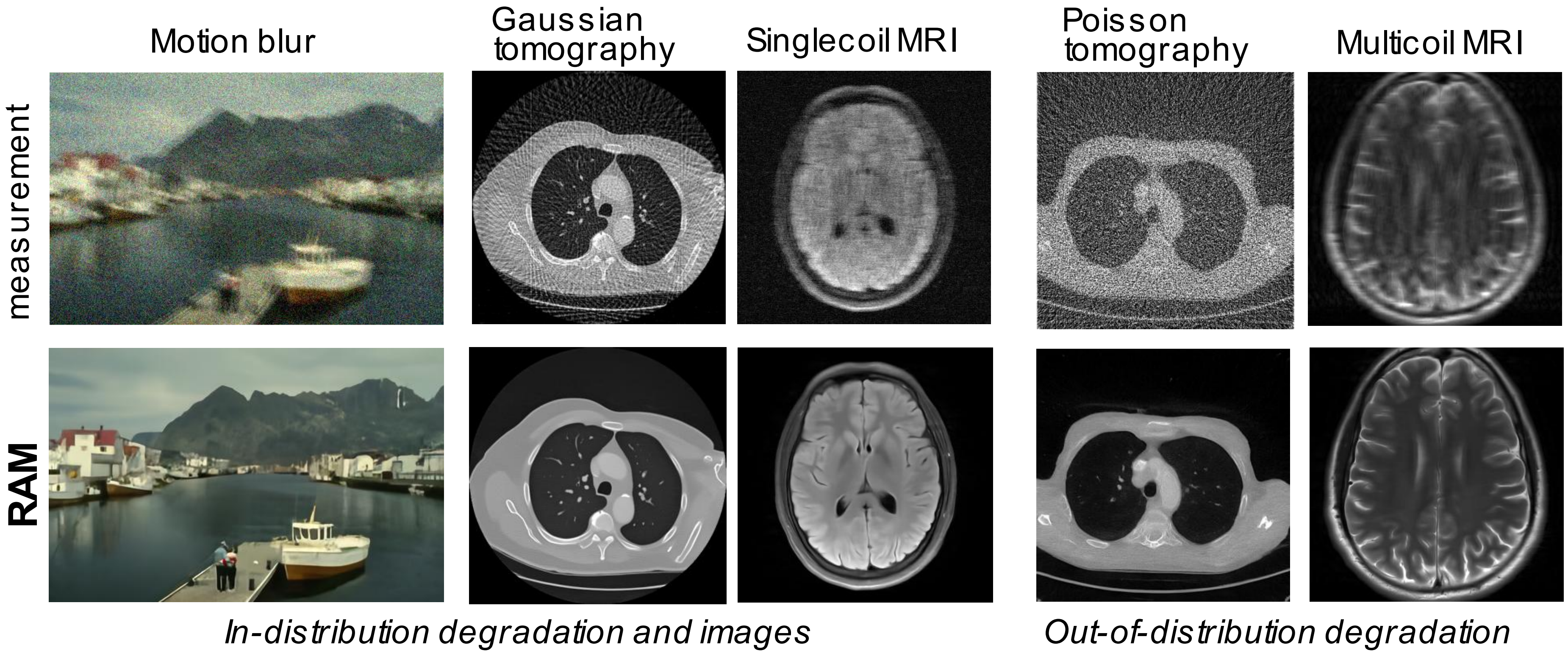


Scanvic, Davies, Abry, T., 2024

Bonus: Finetuning foundation models

Reconstruct Anything Model

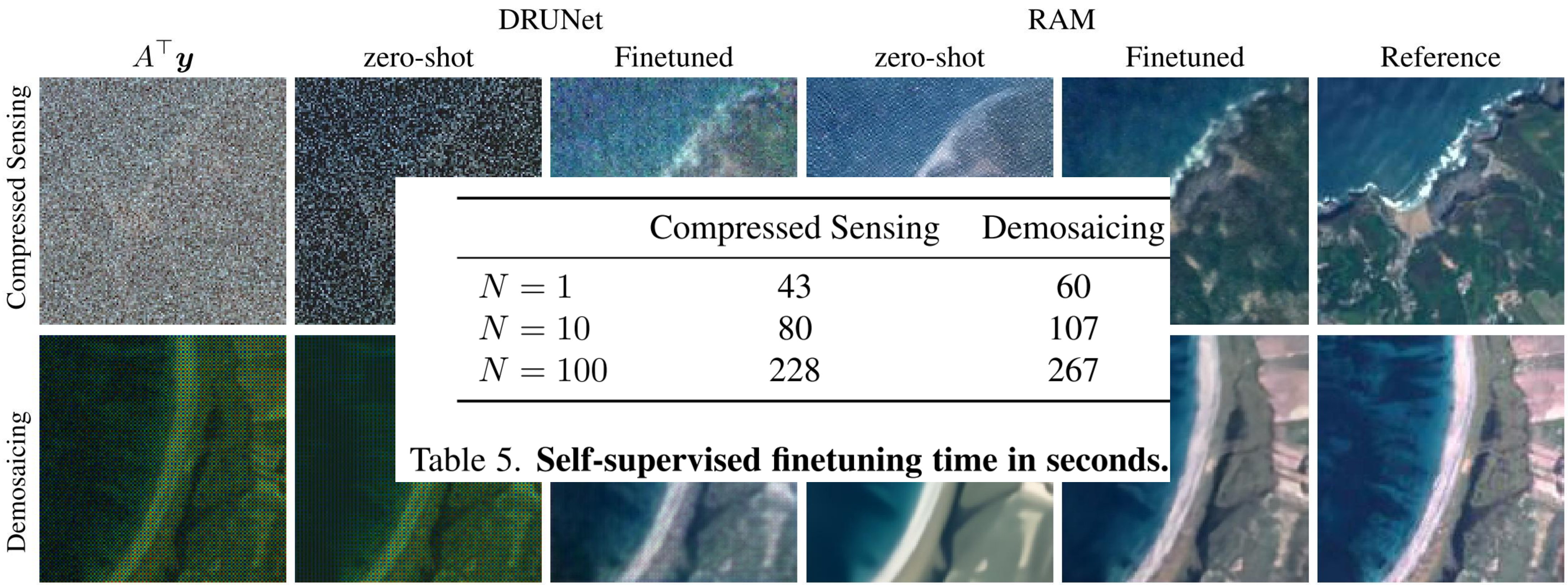
- We trained a model that can solve many inverse problems at once [Terris et al., 2025]



Zero-shot performance

Finetuning

- The model can be finetuned with self-supervised losses on up to a single $\boldsymbol{y}$ ($N = 1$)
- Finetuning can be done in a **few seconds**



	Compressed Sensing	Demosaicing
$N = 1$	43	60
$N = 10$	80	107
$N = 100$	228	267

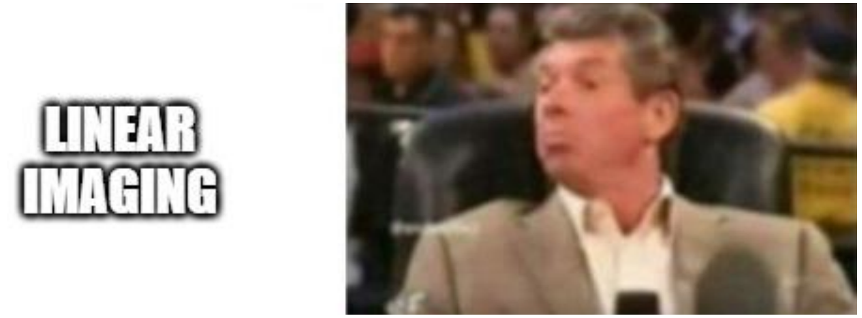
Table 5. **Self-supervised finetuning time in seconds.**

Conclusions

Self-supervised learning for imaging problems

- **Theory:** Necessary & sufficient conditions for learning
    - Unbiased risk estimators
    - Number of measurements
    - Interplay between forward operator & data invariance

- **Practice:** self-supervised losses
    - Can be applied to any model (including foundation ones!)
    - Losses can be combined together



- Poor reconstructions
- Cannot handle noise/missing data

Quickstart Examples User Guide API Finding Help More

Section Navigation

- Basics
- Optimization
- Plug-and-Play
- Sampling
- Unfolded
- Patch Priors
- Self-Supervised Learning
- Adversarial Learning
- Advanced

Examples

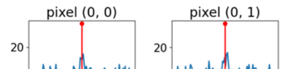
Examples

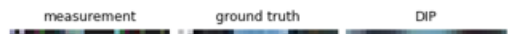
All the examples have a download link at the end. You can load the example's notebook on Google Colab and run them by adding the line

```
pip install git+https://github.com/deepinv/deepinv.git#egg=deepinv
```

to the top of the notebook (e.g., as in here).

Basics





Deep Inverse

References

Paper references:

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

Code examples:

https://andrewwango.github.io/deepinv-selfsup-fastmri/demo

https://deepinv.github.io/deepinv/auto_examples/self-supervised-learning/index.html

YouTube version (3 hours):

https://youtu.be/gf-WCHXAdfk?si=bRC6Pq0WpZHNrRLU