Medical Imaging Research Laboratory
www.creatis.insa-lyon.fr

Deep learning for Inverse Problems: a Focus on Compressive Optics

Nicolas DUCROS[1]

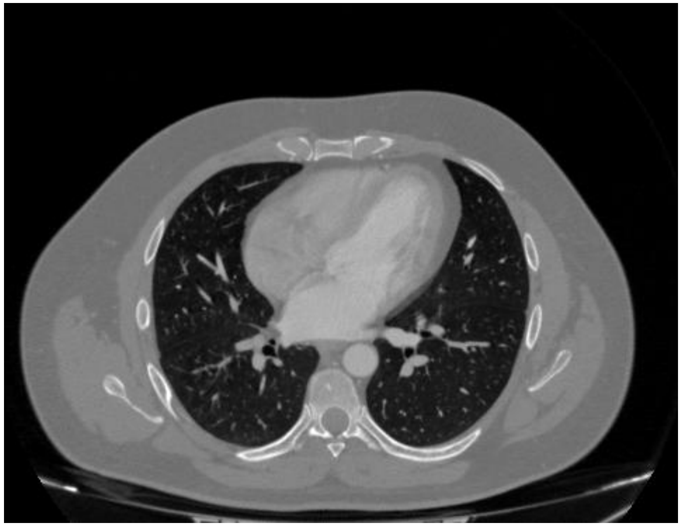
[1]CREATIS, Univ Lyon, INSA-Lyon, UCB Lyon 1,CNRS, Inserm, CREATIS UMR 5220, U1206, Lyon, France
https://www.creatis.insa-lyon.fr/~ducros/

This work was supported by the French National Research Agency (ANR), under Grant ANR-17-CE19-0003 (ARMONI Project). It was performed within the framework of the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the programme "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the ANR.

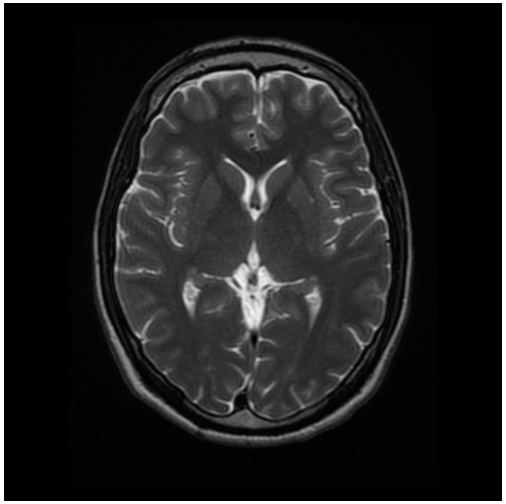
INVERSE PROBLEMS

Ultrasound Imaging

Computerized tomography (CT)



Magnetic Resonance



INVERSE PROBLEMS

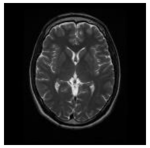
- **Internal unknowns from external measurements**

$$m = A(f)$$

External Measurement

Forward Model

Internal Quantity of Interest



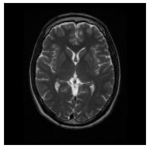
- **Most medical imaging problem are linear. In a discrete setting:**

$$\boldsymbol{m} = \boldsymbol{A}\boldsymbol{f}$$

$\boldsymbol{m} \in \mathbb{R}^M$

$\boldsymbol{A} \in \mathbb{R}^{M \times N}$

$\boldsymbol{f} \in \mathbb{R}^N$



Traditional approaches

$$\min_{\boldsymbol{f}} \; \underbrace{\|\boldsymbol{m} - \boldsymbol{A}\boldsymbol{f}\|_2^2}_{\text{Data fidelity}} + \underbrace{\mathcal{R}(\boldsymbol{f})}_{\text{Regularization}}$$

- The regularizer is hand-crafted

$$\mathcal{R}(\boldsymbol{f}) = \|\boldsymbol{f}\|_2^2 \qquad\qquad \mathcal{R}(\boldsymbol{f}) = \|\nabla\boldsymbol{f}\|_1$$

- Minimization usually required iterative algorithms
- Time consuming

Deep reconstruction methods

$$\boldsymbol{f}^* = \mathcal{G}_\omega(\boldsymbol{m})$$

Neural Network

Network Parameters

Training

- Database

$$\{\boldsymbol{f}^{(\ell)}; \boldsymbol{m}^{(\ell)})\},\ 1 \leq \ell \leq L$$

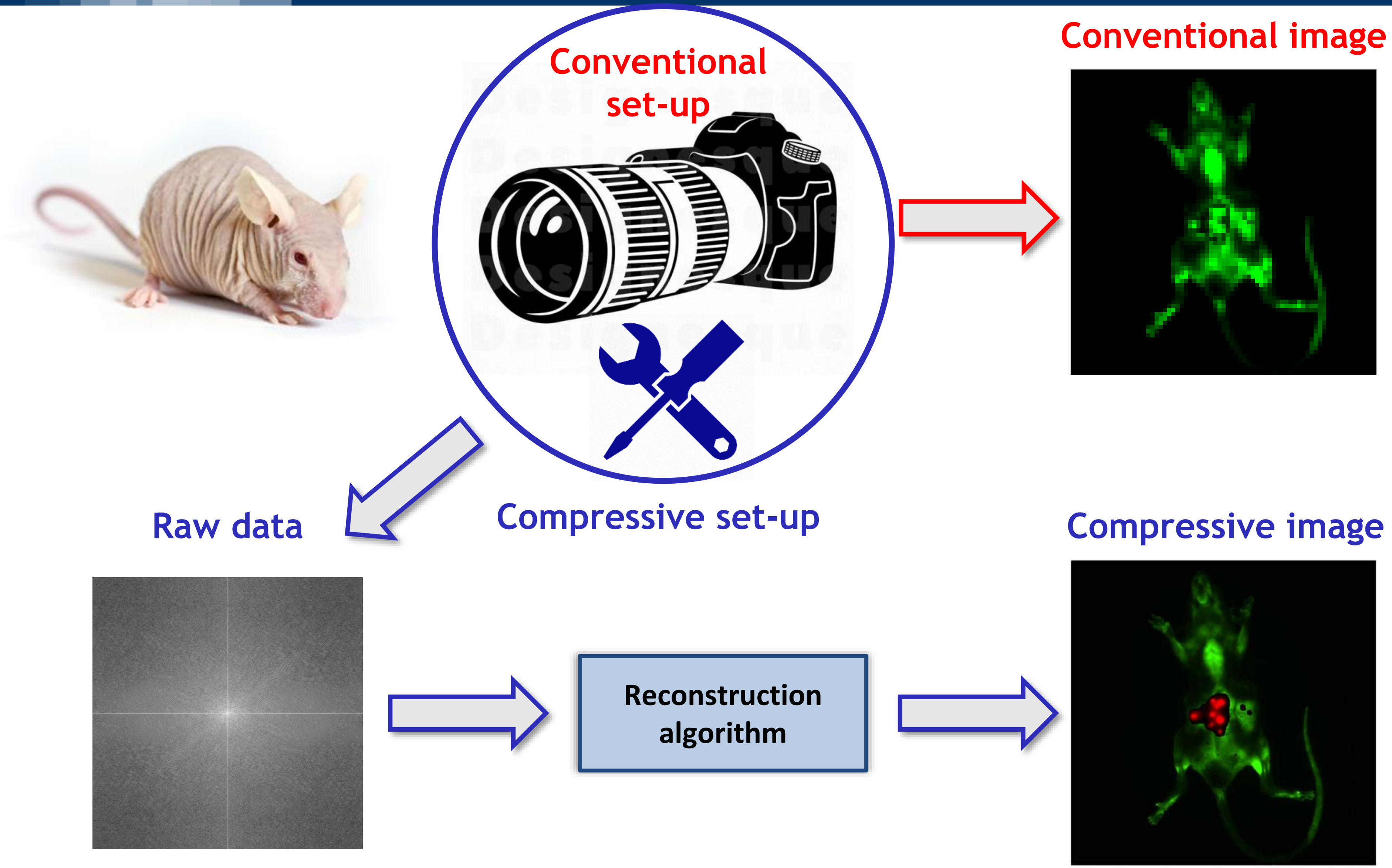
- (Stochastic) optimization of a 'loss'

$$\min_{\underline{\boldsymbol{\omega}}} \sum_{\ell=1}^{L} \|\boldsymbol{f}^{(\ell)} - \mathcal{G}(\underline{\boldsymbol{\omega}}; \boldsymbol{m}^{(\ell)})\|_2^2$$

Computation times

- Training phase is slow (e.g., several hours or days)
- Evaluation is fast (e.g., tens of millliseconds)

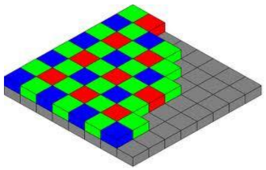
COMPRESSIVE OPTICS



HYPERSPECTRAL IMAGING

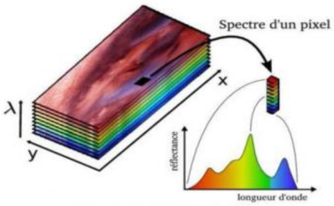
Array

Colour



Multi-spectral

Hyper-spectral



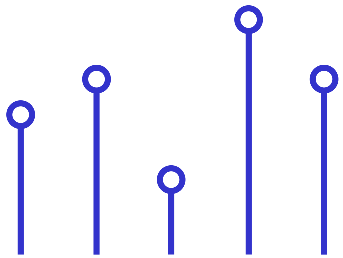
Point

Spectrometer

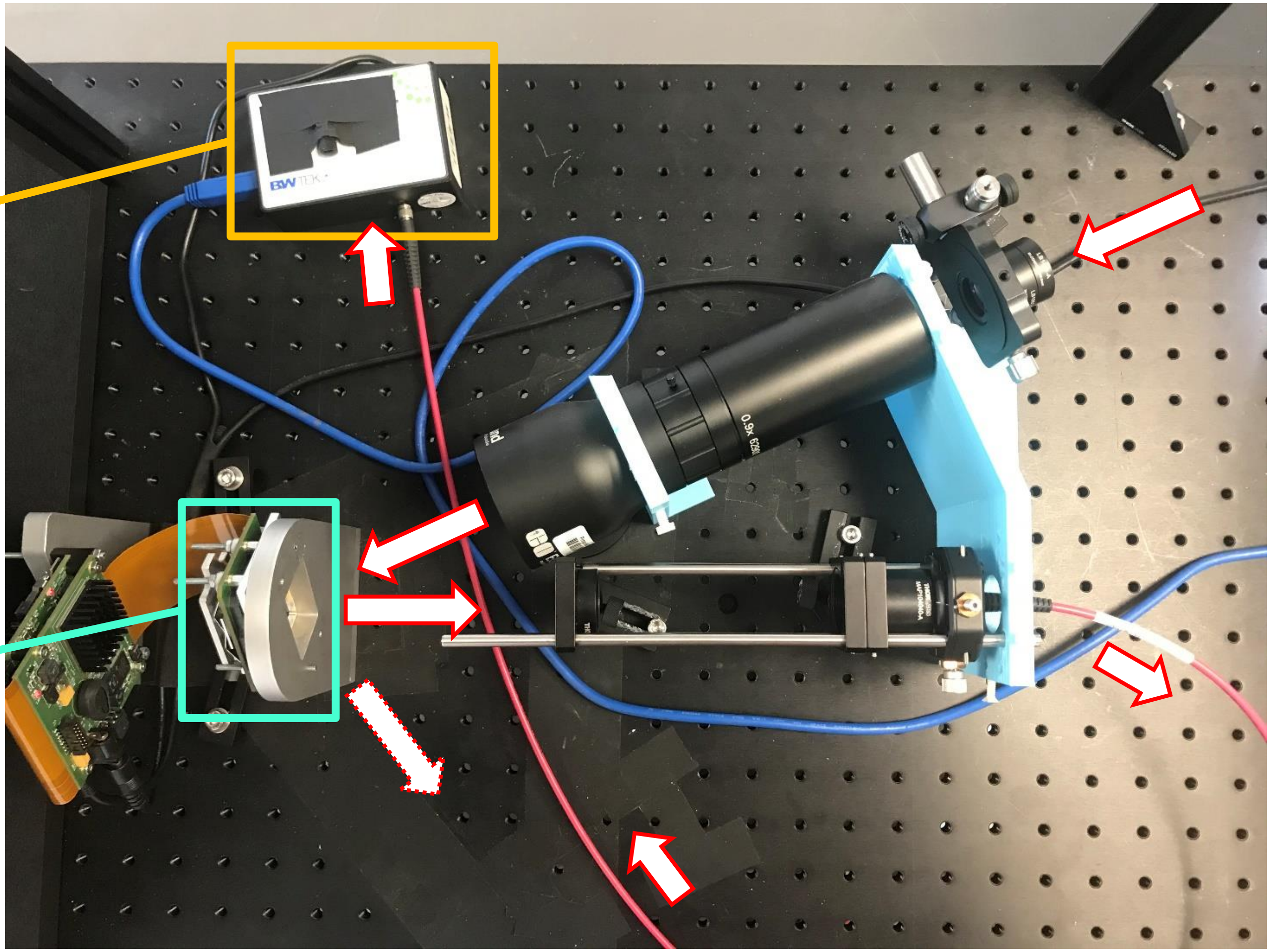
	Colour	Multi-spectral	Hyper-spectral	Spectrometer
Spatial resolution	yes	yes	yes	no
Number of spectral channels / *Spectral channels*	3	2—10	10—100	100—500
Cost	~€1k	~€10k	~€100k	~€1k

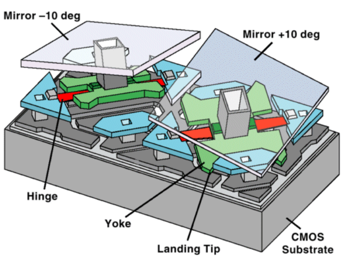
Need for **low cost array** with **high spectral** resolution

COMPRESSIVE (SINGLE-PIXEL) CAMERA

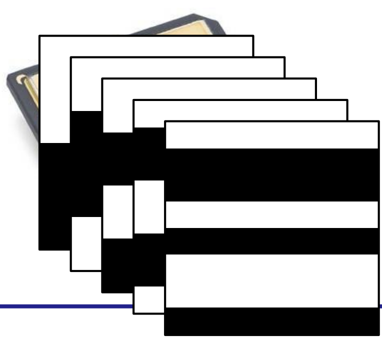


Point detector (spectrometer)

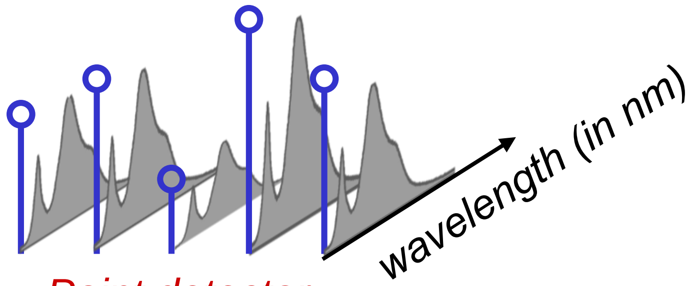




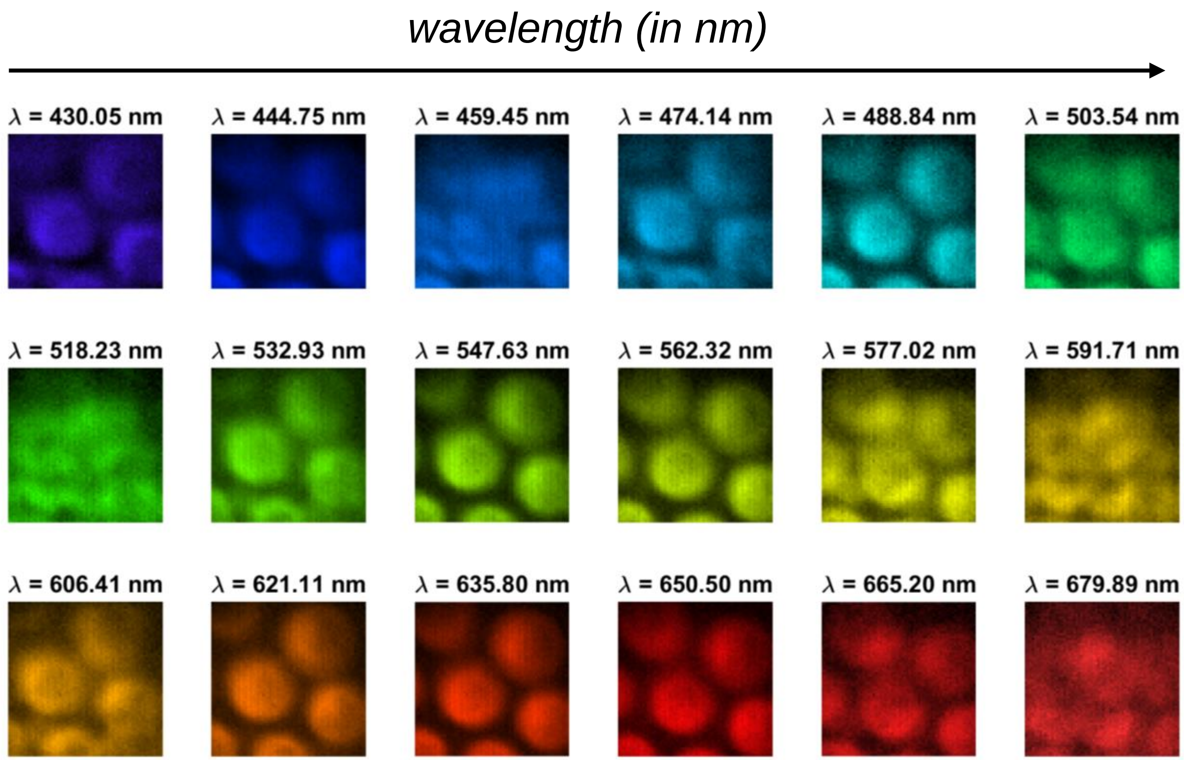
Spatial light modulator



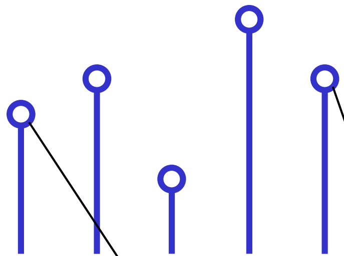
COMPRESSIVE HYPERSPECTRAL CAMERA



Point detector
(spectrometer)

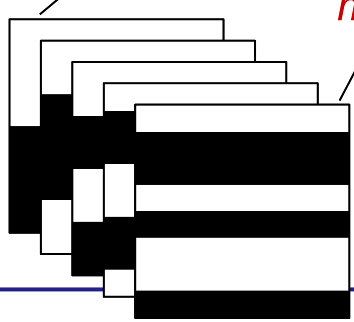


ACQUISITION MODEL



$$\boldsymbol{m} = [m_1, \ldots, m_M]^\top \in \mathbb{R}^M$$

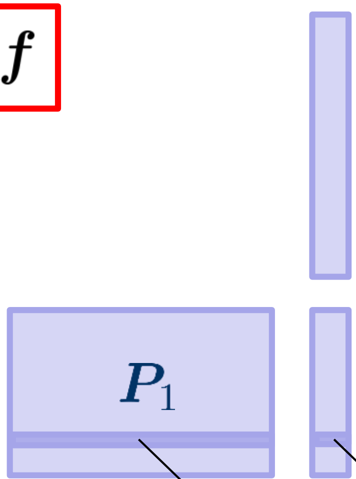
$$\boldsymbol{P}_1 = [\boldsymbol{p}_1^\top, \ldots, \boldsymbol{p}_M^\top]^\top \in \mathbb{R}^{M \times N}$$



➢ **Linear model**

$$\boxed{\boldsymbol{m} = \boldsymbol{P}_1 \boldsymbol{f}}$$

$M \ll N$



➢ **Challenge**

- A small M limits the acquisition time
- A small M limits the image resolution too!

ACQUISITION-RECONSTRUCTION

$$\boxed{m = P_1 f}$$

1. **Weight design: How to choose the P?**
2. **Reconstruction: How to recover the image f?**

Acquisition

$$\mathbf{P} = \mathbf{H}_4$$

+	+	+	+
+	+	-	-
+	-	-	+
+	-	+	-

Reconstruction

$$\mathbf{H}_N^{-1} = \frac{1}{N}\mathbf{H}_N$$

$$\mathbf{f}^* = \frac{1}{N}\mathbf{H}_N\mathbf{m}$$

Noise reduction

$$m_k \sim \mathcal{G}(\mu = 0, \sigma^2) \qquad 1 \leq k \leq K$$

- Raster scan

$$\text{var}\left(f_n^*\right) = \sigma^2$$

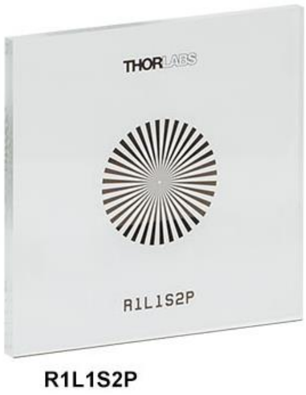
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

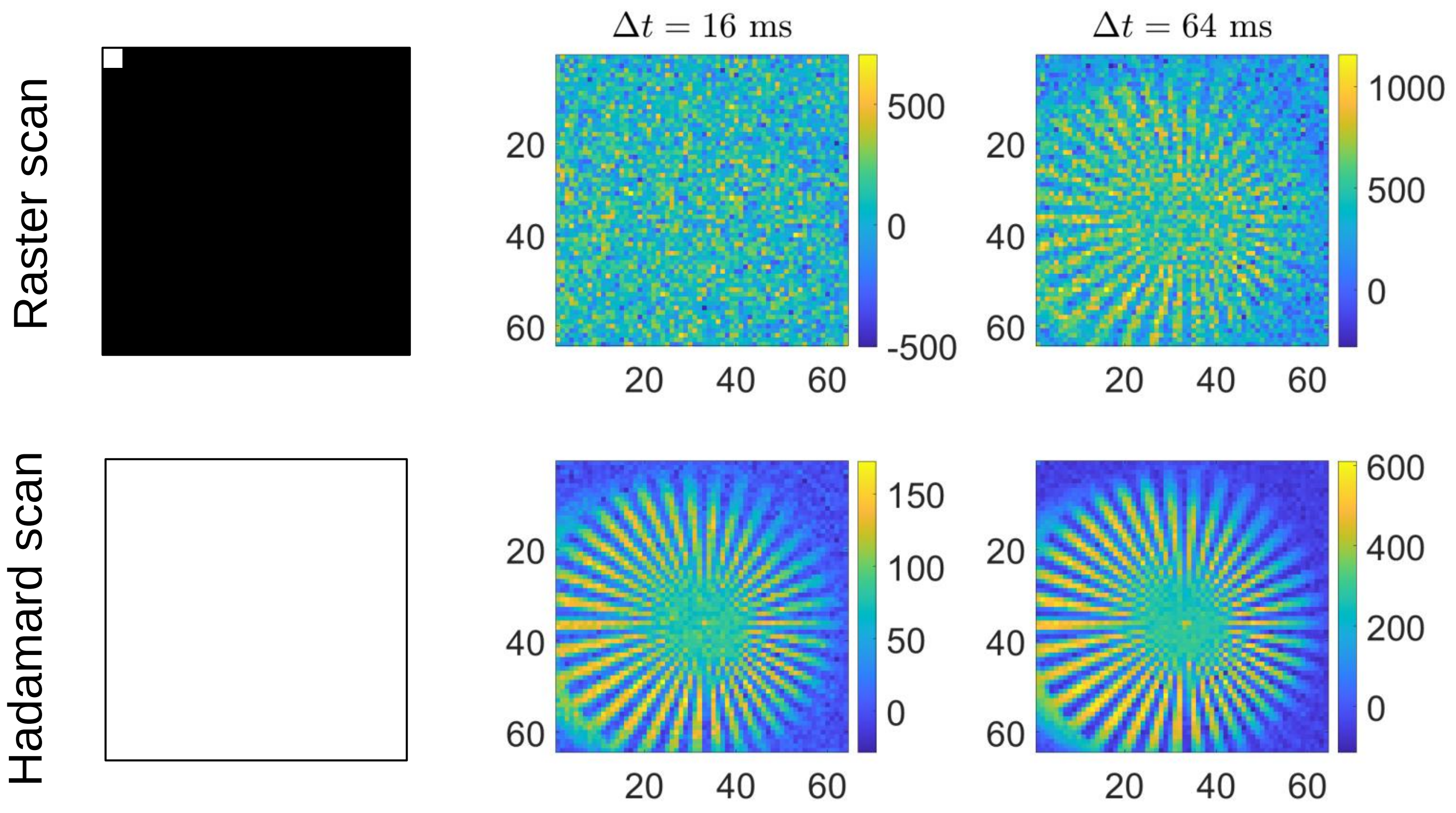
- Hadamard

$$\text{var}\left(f_n^*\right) = \frac{1}{N}\sigma^2$$

+	+	+	+
+	+	-	-
+	-	-	+
+	-	+	-

➢ Boosting effect

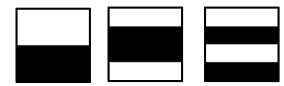




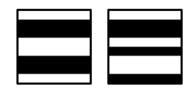
[N. Ducros *et al*, working paper, 2020]

1. **Experiment design:** How to choose the patterns (codes) P?
Not addressed here! We choose

$$\boxed{\boldsymbol{m} = \boldsymbol{P}_1 \boldsymbol{f}}$$



orthogonal basis → $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \end{bmatrix}$ ← **acquired** patterns ← **missing** patterns



2. **Reconstruction:** How to recover the image ***f*** from ***m***?

❖ Constrained optimization

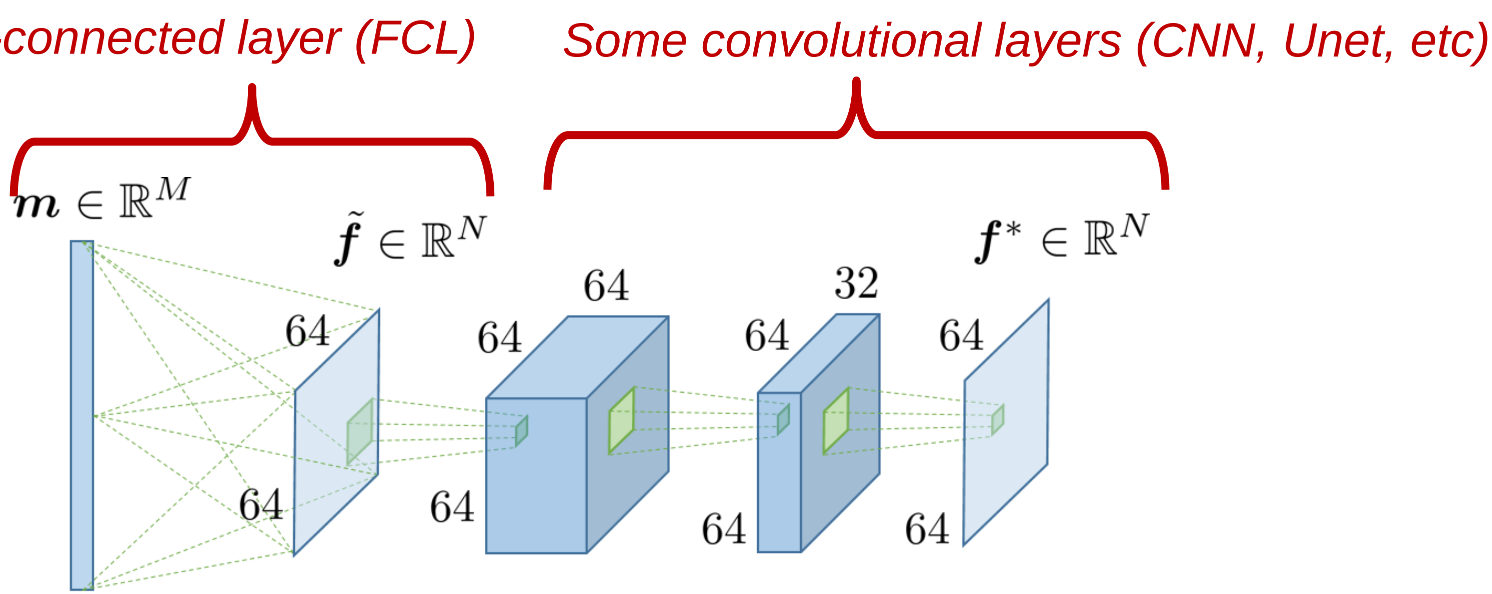
$$\min_{\boldsymbol{f}} \mathcal{R}(\boldsymbol{f}) \quad \text{such that} \quad \boldsymbol{m} = \boldsymbol{P}_1 \boldsymbol{f}.$$

- Least squares: fast but low resolution [Rousset *et. al*, IEEE TCI, 2017]
- Total variation: higher resolution but time consuming [Duarte *et. al*, IEEE SPM, 2009]

2. Reconstruction: How to recover the image $\boldsymbol{f}$ from $\boldsymbol{m}$?

- Deep learning: Learn a nonlinear reconstructor [Higham *et. al*, Sci. Rep., 2018]

$$\boldsymbol{f}^* = \mathcal{H}_{\boldsymbol{\theta}}(\boldsymbol{m}),$$



→ How to choose the non linear 'model'?

→ How does this relate to traditional approaches?

RECONSTRUCTION AS COMPLETION

- **The least-squares problem**

$$\min_{\boldsymbol{f}} \|\boldsymbol{f}\|_2^2 \quad \text{such that} \quad \boldsymbol{m} = \boldsymbol{P}_1 \boldsymbol{f}.$$

…has the closed-form solution

$$\boldsymbol{f}^* = \boldsymbol{P}_1^\top \boldsymbol{m}$$

… equivalent to

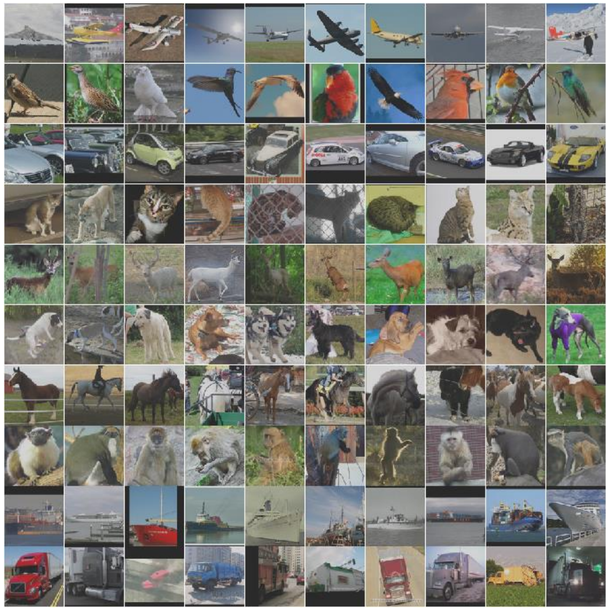
$$\boldsymbol{f}^* = \boldsymbol{P}^\top \boldsymbol{y}^*, \quad \text{with } \boldsymbol{y}^* = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^N$$

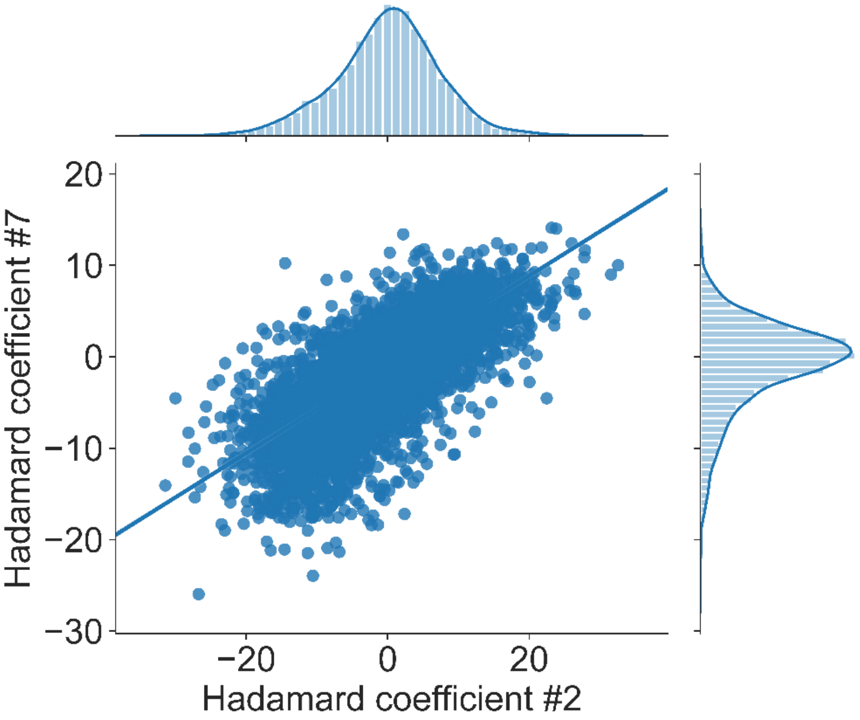
What about completing the missing measurements by relevant values?

RECONSTRUCTION AS COMPLETION

➢ **How to complete?**

STL-10 dataset





→ Exploiting the correlation between the measured coefficients

RECONSTRUCTION AS COMPLETION

Completion approach

$$\boxed{\boldsymbol{f}^* = \boldsymbol{P}^\top \boldsymbol{y}^*, \quad \text{with } \boldsymbol{y}^* = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{y}_2^* \end{bmatrix},}$$

with

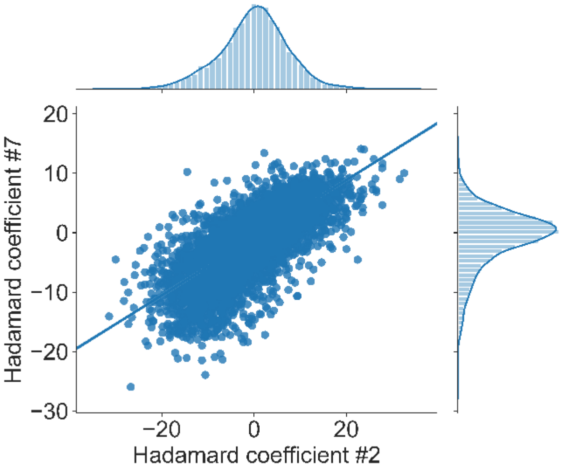
$$\boldsymbol{y}_2^*(\boldsymbol{m}) = \mathbb{E}\left(\mathbf{y}_2 \,|\, \mathbf{y}_1 = \boldsymbol{m}\right)$$

- **Under Gaussian assumptions**

$$\boldsymbol{y}_2^*(\boldsymbol{m}) = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{m} - \boldsymbol{\mu}_1)$$

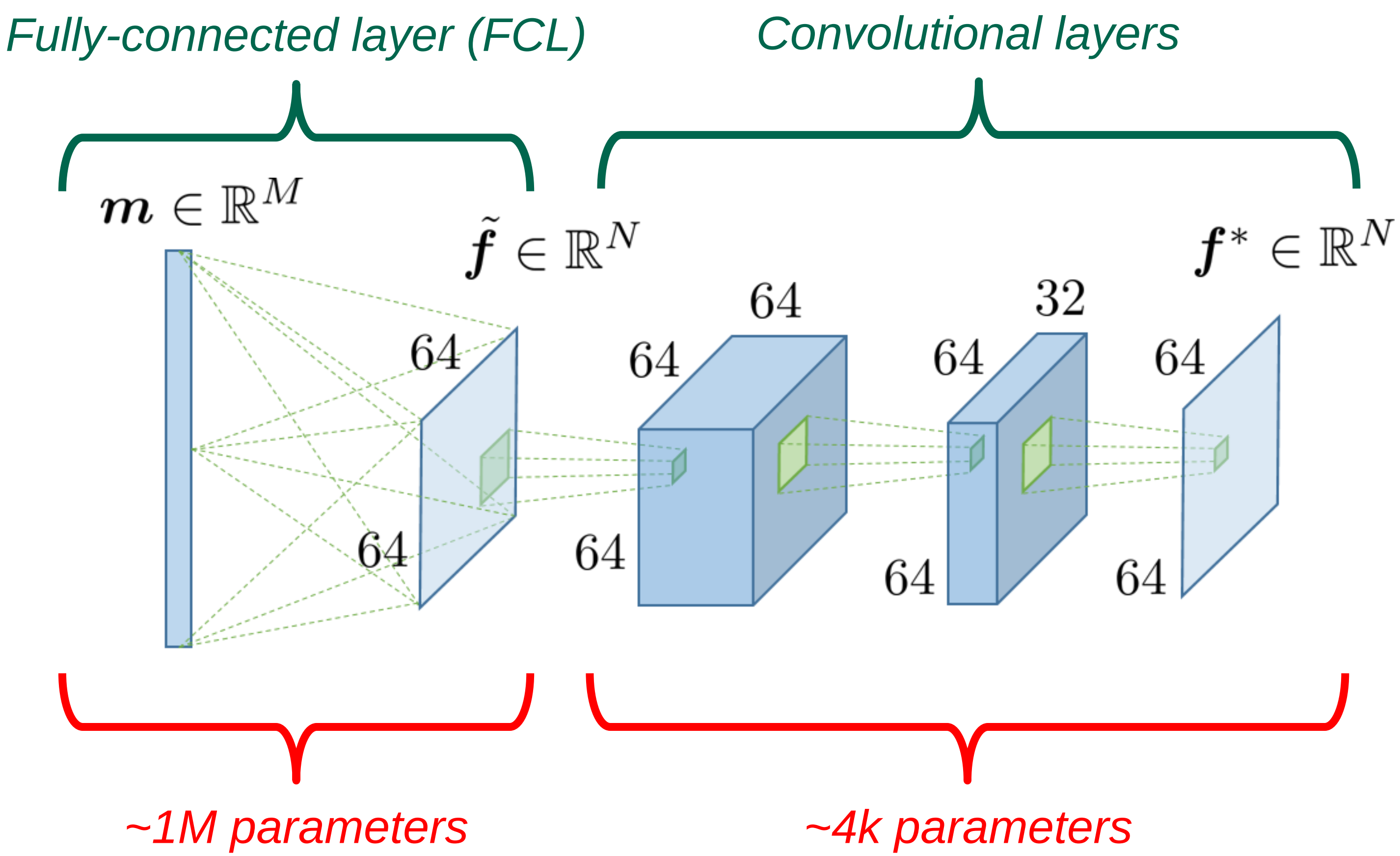
Covariance between measured and missing

Covariance of measured



- **No assumption: This is the best linear solution!**

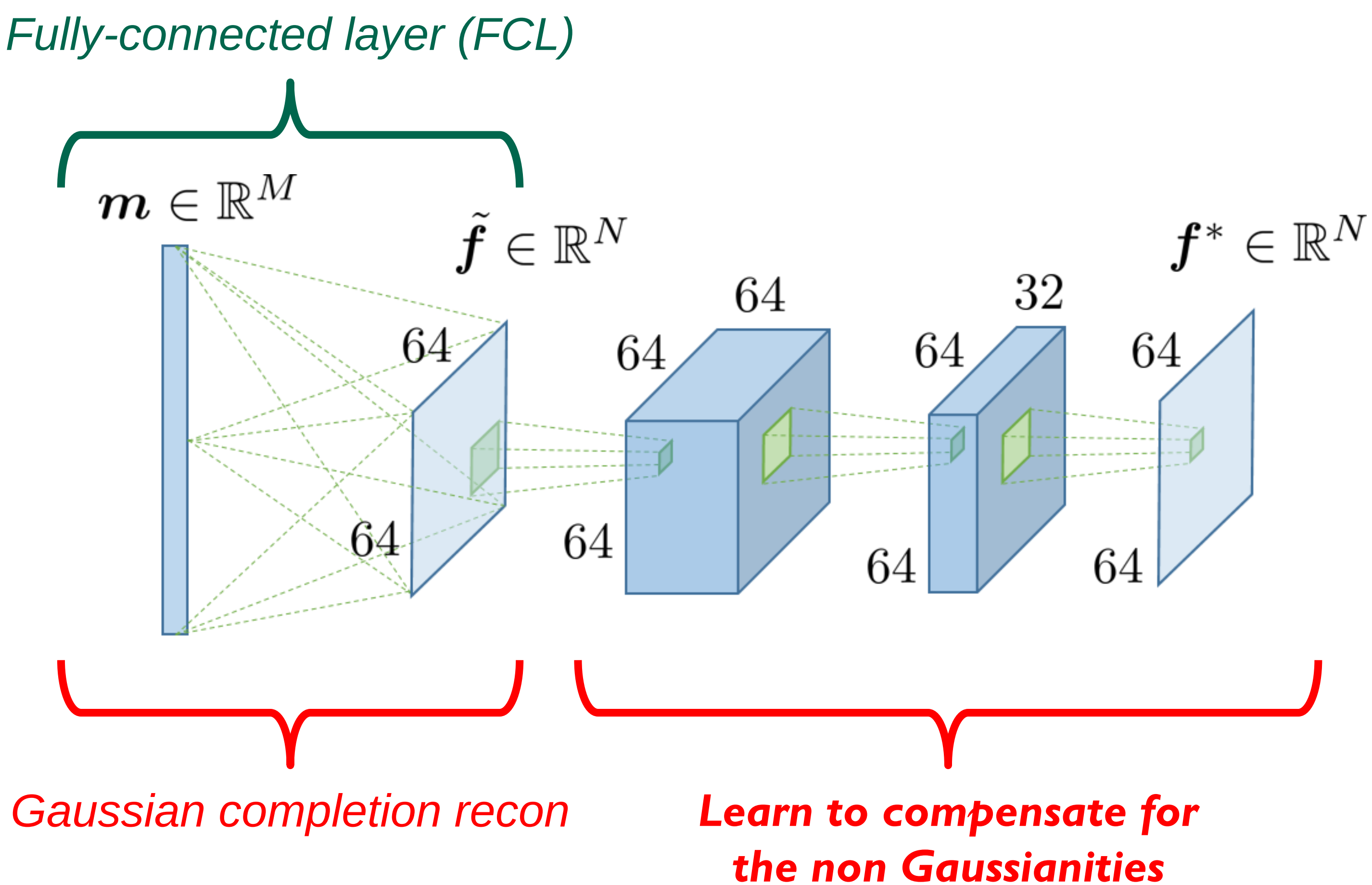
Traditional CNN architecture



[Higham *et. al*, Sci. Rep., 2018]

DEEP COMPLETION RECONSTRUCTION

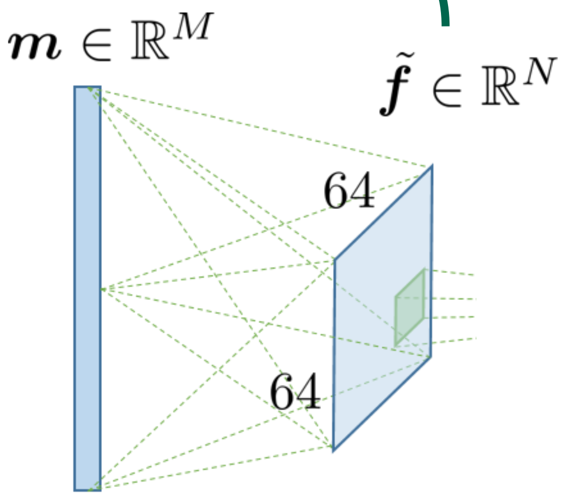
- **Traditional CNN architecture**



DEEP COMPLETION RECONSTRUCTION

➢ CNN architecture

Fully-connected layer (FCL)



measurement-to-image mapping

➢ Choices for the FCL

- **Free** [Higham *et. al*, Sci. Rep., 2018]

$$\tilde{\boldsymbol{f}} = \mathcal{H}_{\theta_1}(\boldsymbol{m})$$

- **Pseudo inverse** [Jin *et. al*, IEEE TIP, 2017, Ravishankar *et. al*, Proc. IEEE, 2020]

$$\tilde{\boldsymbol{f}} = \boldsymbol{P}_1^\top \boldsymbol{m}$$

- **Bayesian completion** [N. Ducros *et. al*, IEEE ISBI, 2020]

$$\tilde{\boldsymbol{f}} = \boldsymbol{P}^\top \boldsymbol{y}^*, \quad \text{with } \boldsymbol{y} = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{y}_2^* \end{bmatrix}$$

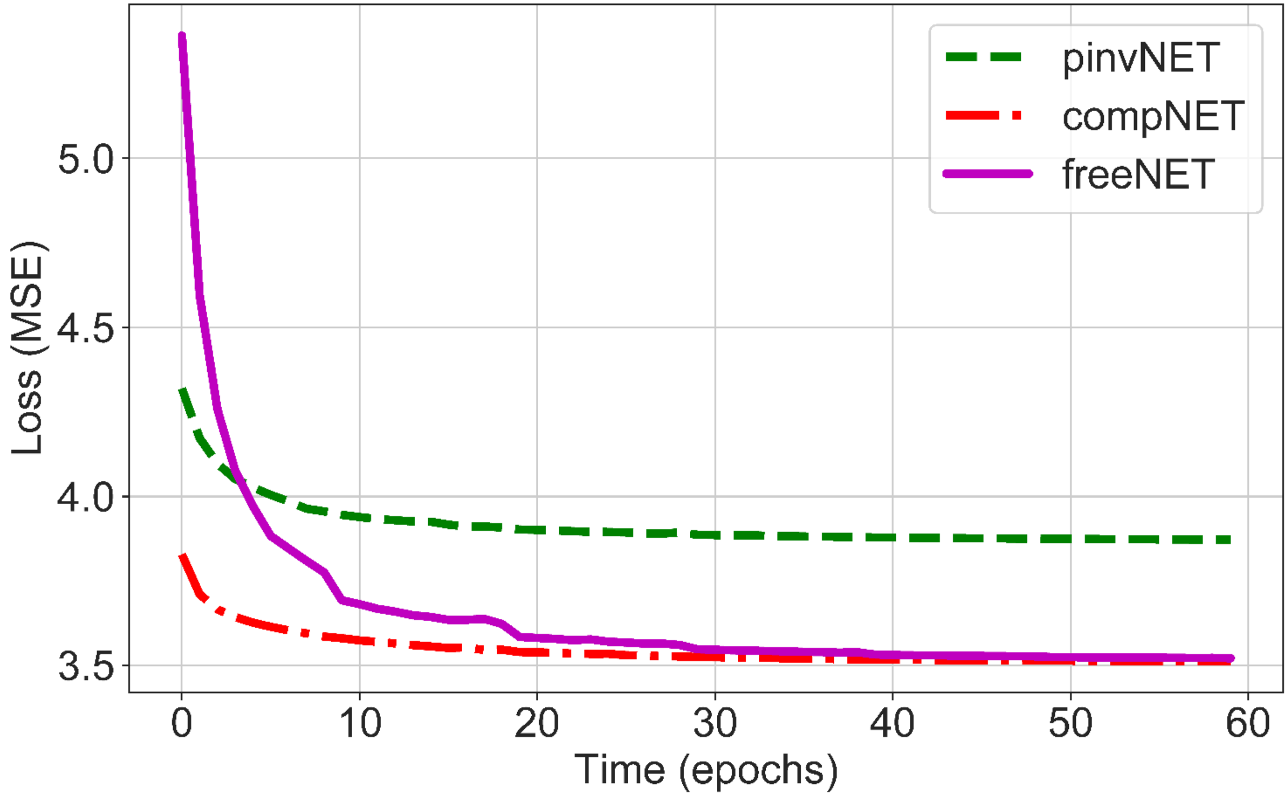
➢ 3 network variants

- freeNet: (~1M parameters)
- pinvNet: (~4k parameters)
- compNet: (~4k parameters)

SIMULATION RESULTS

- **STL-10 (training using ~100k images; testing using 8k images)**

$$\min_{\boldsymbol{\theta}} \sum_i \|\boldsymbol{f}^{(i)} - \mathcal{H}_{\boldsymbol{\theta}}(\boldsymbol{m}^{(i)})\|^2$$

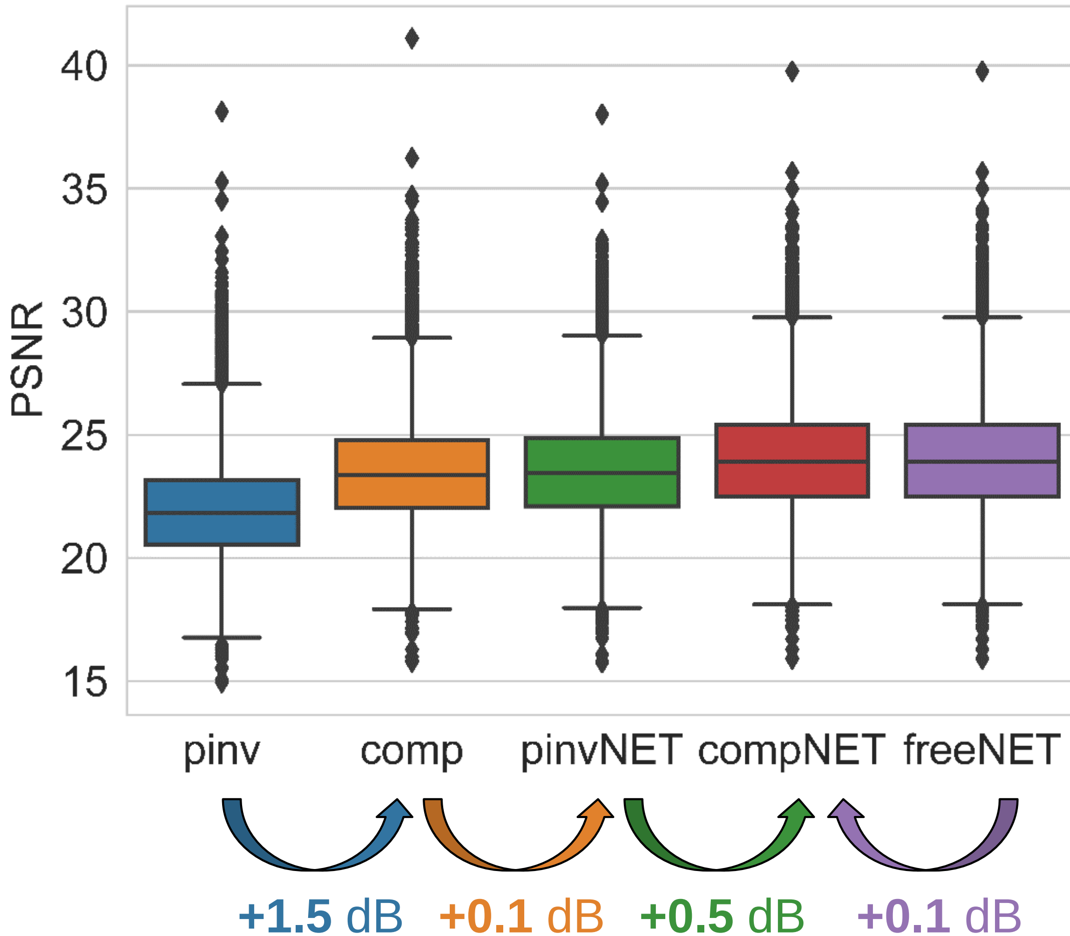


TEST image set

SIMULATION RESULTS

➢ **STL-10 (Training using ~100k images, test using 8k images)**

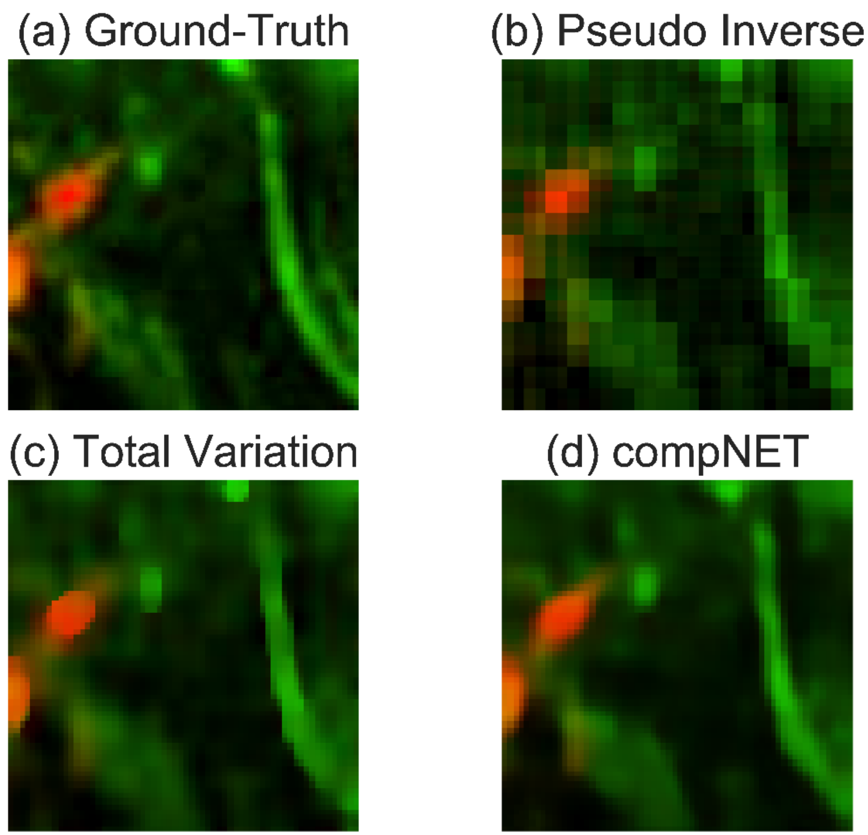
pinv: 22.0 ± 2.2 dB
comp: 23.5 ± 2.2 dB
pinvNET: 23.6 ± 2.2 dB
compNET: 24.1 ± 2.3 dB
freeNET: 24.0 ± 2.2 dB



SIMULATION RESULTS

➢ **Fluorescence microscopy images (not from STL-10!)**

$$\boldsymbol{f}^{\mathrm{true}}$$



$$\boldsymbol{f}^* = \underset{\substack{\boldsymbol{f} \\ \text{s.t.}\ \boldsymbol{m}=\boldsymbol{P}\boldsymbol{f}}}{\arg\min} \ \|\boldsymbol{f}\|_2^2$$

red: 27.15 dB
green: 24.27 dB

$$\boldsymbol{f}^* = \underset{\substack{\boldsymbol{f} \\ \text{s.t.}\ \boldsymbol{m}=\boldsymbol{P}\boldsymbol{f}}}{\arg\min} \ \|\boldsymbol{\nabla}\boldsymbol{f}\|_1$$

$$\boldsymbol{f}^* = \mathcal{H}_{\boldsymbol{\theta}^*}(\boldsymbol{m})$$

red: PINV + **3.52** dB
green: PINV + **2.41**dB

red: TV + **0.8** dB
green: TV + **1.16** dB

SO WHAT?

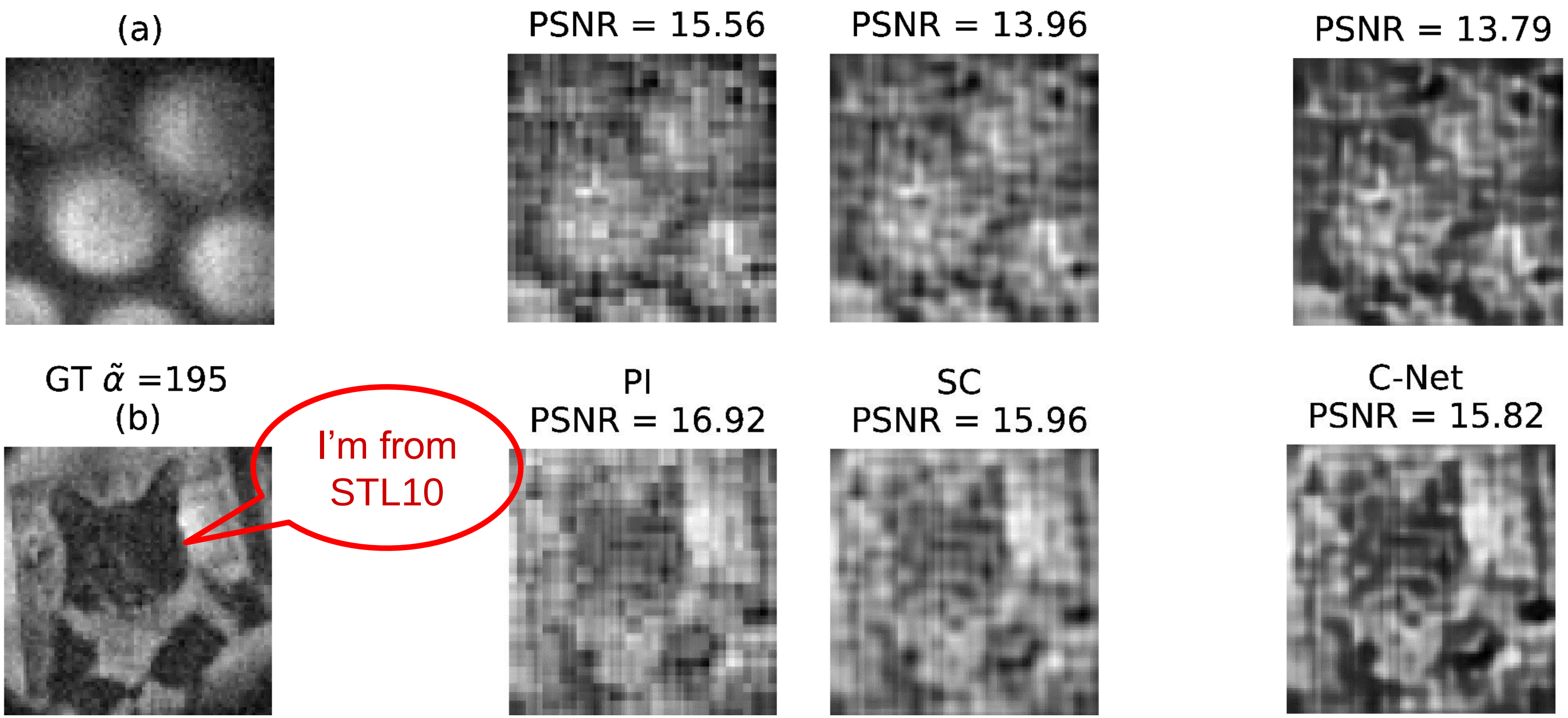
… Does it work with experimental data?

Ground Truth

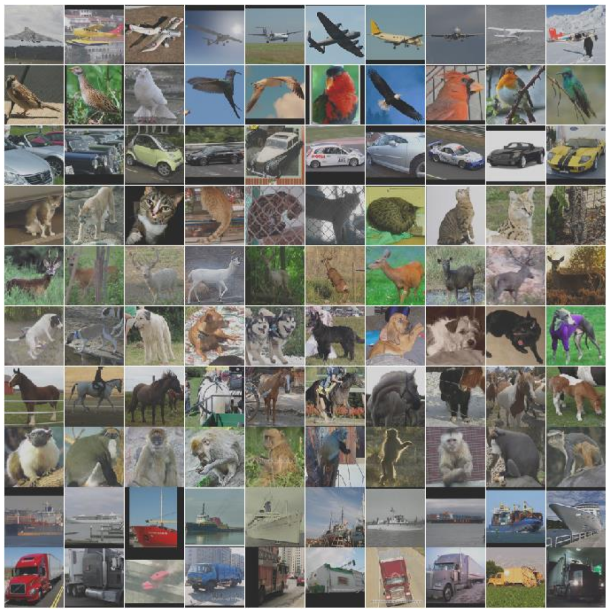
No Completion

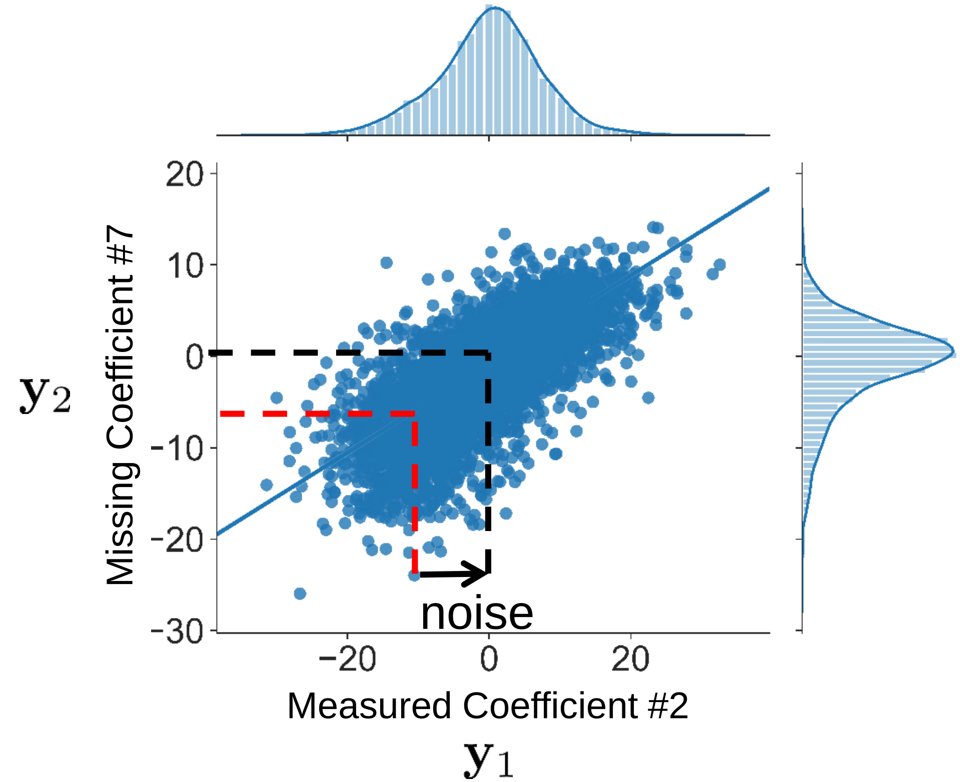
Statistical Completion

Completion Network



NOISE IS A BIG ISSUE!





$$\boldsymbol{y}_2^*(\boldsymbol{m}) = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{m} - \boldsymbol{\mu}_1)$$

Covariance between measured and missing

Covariance of measured

CAN WE 'UPGRADE' OUR APPROACH?

- **Gaussian Models (hypotheses)**

Data Model

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Noise Model

$$\mathbf{m}^{\alpha} \,|\, \mathbf{y} \sim \mathcal{N}(\boldsymbol{y}_1, \boldsymbol{\Sigma}_{\alpha})$$

Noisy Measurement

(Noiseless) Coefficients

- **We reconstruct our images by computing**

$$\boldsymbol{y}^{*}(\boldsymbol{m}^{\alpha}) = \mathbb{E}\left(\mathbf{y}_1, \mathbf{y}_2 \,|\, \mathbf{m}^{\alpha} = \boldsymbol{m}^{\alpha}\right)$$

CAN WE 'UPGRADE' OUR APPROACH?

- **Our problem…**

$$\boldsymbol{y}^*(\boldsymbol{m}^\alpha) = \mathbb{E}\left(\mathbf{y}_1, \mathbf{y}_2 \,|\, \mathbf{m}^\alpha = \boldsymbol{m}^\alpha\right)$$

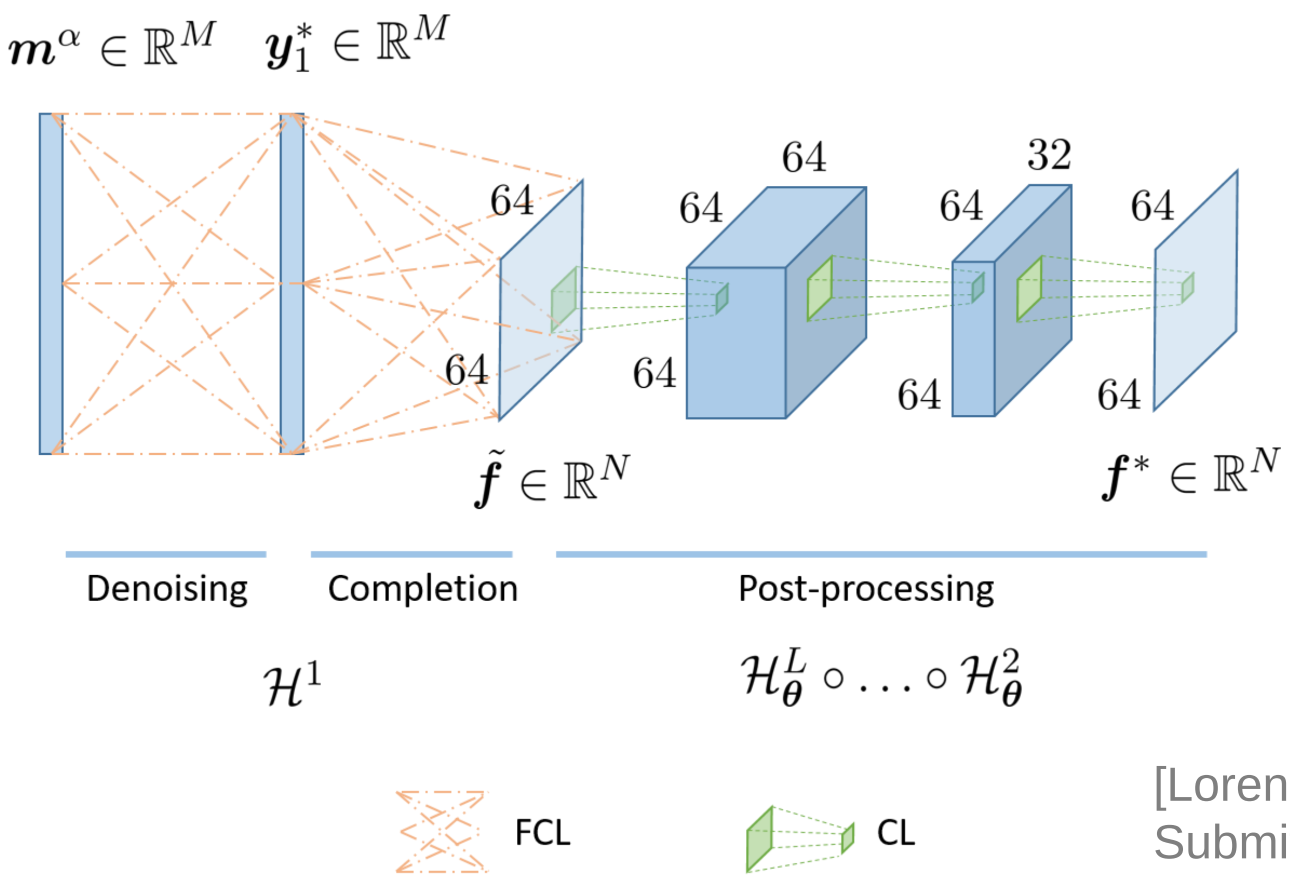
…has the following solution

Denoising Step: $\boldsymbol{y}_1^*(\boldsymbol{m}^\alpha) = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_1[\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_\alpha]^{-1}(\boldsymbol{m}^\alpha - \boldsymbol{\mu}_1)$

Completion Step: $\boldsymbol{y}_2^*(\boldsymbol{m}^\alpha) = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}[\boldsymbol{y}_1^*(\boldsymbol{m}^\alpha) - \boldsymbol{\mu}_1]$

Denoised measurements

DENOISED COMPLETION NETWORK



[Lorente-Mur *et. al*, Submitted, 2021]

POISSON NOISE IS A MESS!

- **Poisson noise model for the raw data**

Measurement (in photons)

Image intensity (in photons)

$$\hat{\mathbf{m}}^{\alpha} \,|\, \mathbf{y} \sim \mathcal{P}(\alpha\, \boldsymbol{y}_1)$$

- Problem 1: Such scaling is incompatible with non linear reconstructors

- **Noise model for the normalised data**

$$\mathbf{m}^{\alpha} \,|\, \mathbf{y} \sim \frac{1}{\alpha}\mathcal{P}(\alpha\, \boldsymbol{y}_1)$$

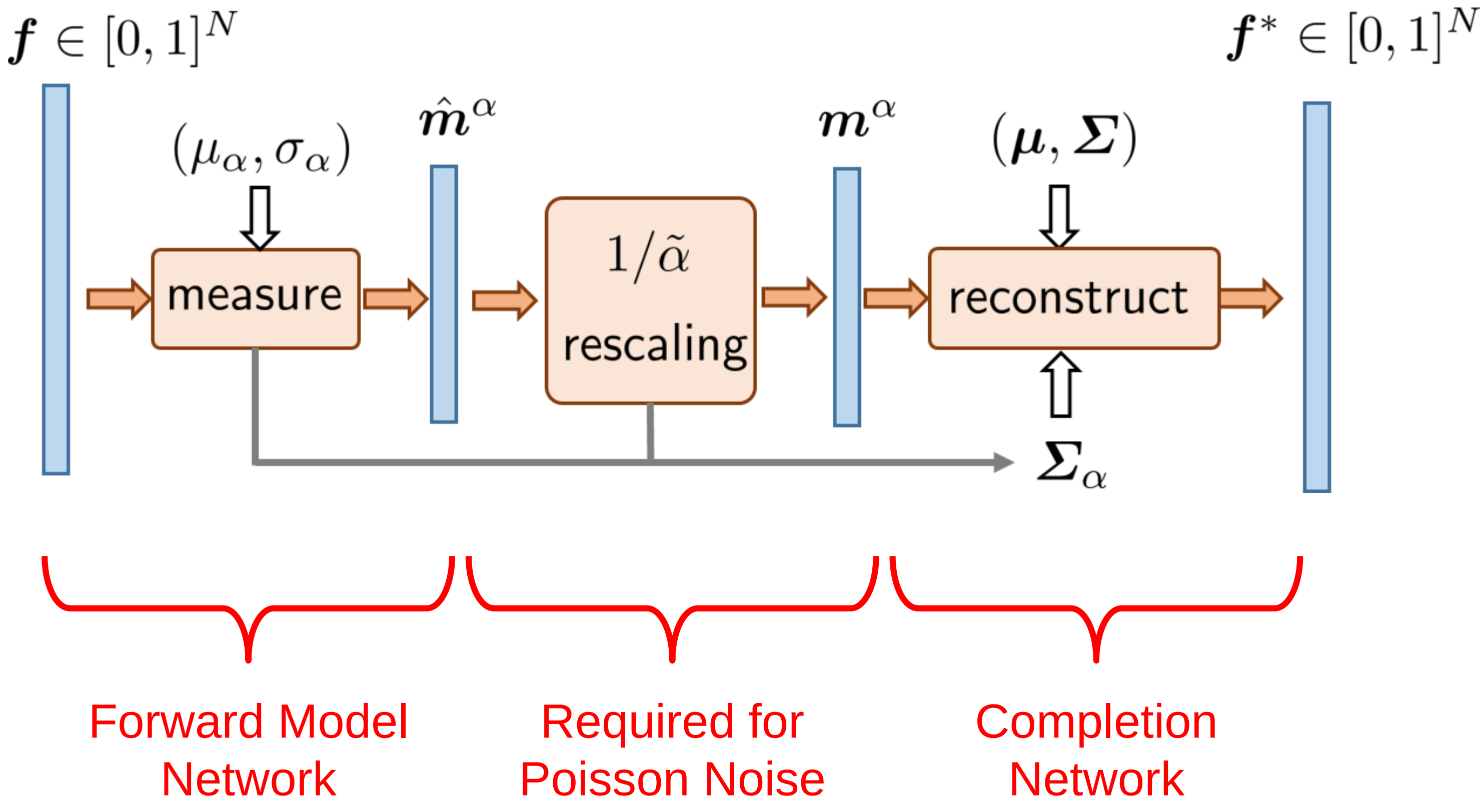
- Problem #2: the image intensity is unknown
- Problem #3: does not satisfy the Gaussian assumption

- **Approximate noise model for the normalised data**

$$\mathbf{m}^{\alpha} \,|\, \mathbf{y} \approx \mathcal{N}(\boldsymbol{y}_1, \boldsymbol{\Sigma}_{\alpha}) \text{ where } \boldsymbol{\Sigma}_{\alpha} = \mathrm{Diag}\,(\boldsymbol{y}_1/\alpha)$$

- Problem #4: The covariance depends on the intensity of the image under acquisition

➢ Training pipeline, i.e., the *full* network

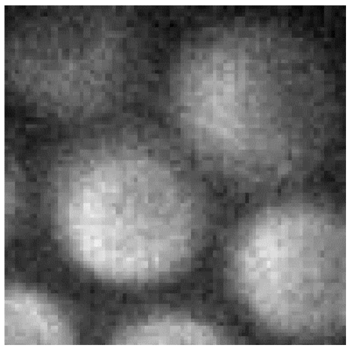


❖ Trained under varying noise levels driven, i.e., $\alpha \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$

EXPERIMENTAL RESULTS

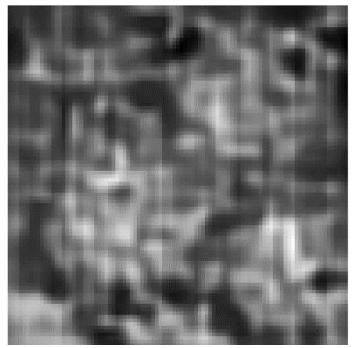
Ground Truth

GT $\tilde{\alpha}$ =148
(a)



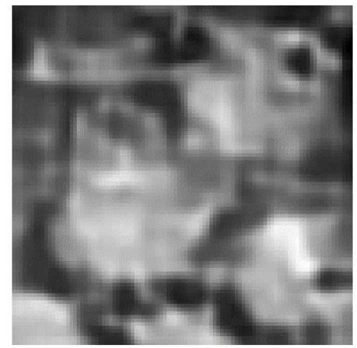
Completion Network

PSNR = 13.79



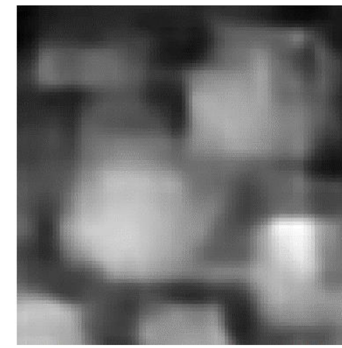
Completion Network trained with Noise

PSNR = 15.65

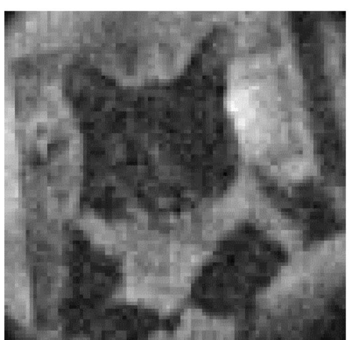


Denoised Completion Network

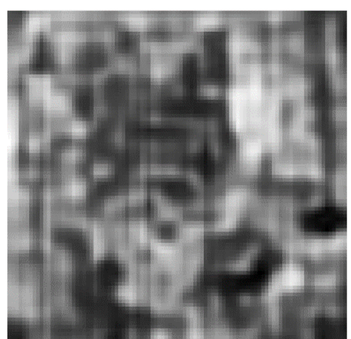
PSNR = 16.14



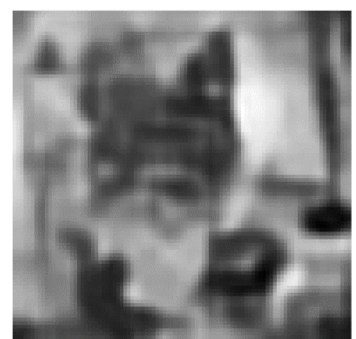
GT $\tilde{\alpha}$ =195
(b)



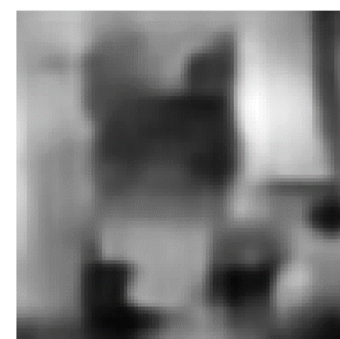
C-Net
PSNR = 15.82



NC-Net
PSNR = 18.18



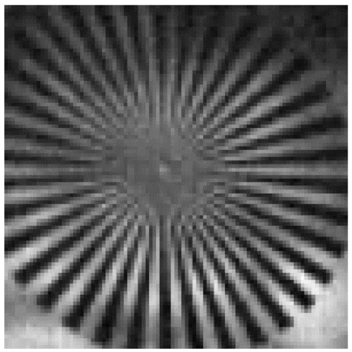
DC-Net
PSNR = 18.91



EXPERIMENTAL RESULTS

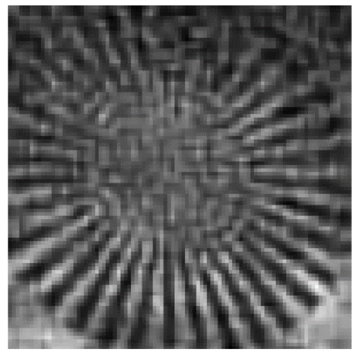
Ground Truth

GT $\tilde{\alpha}$ =295
(c)



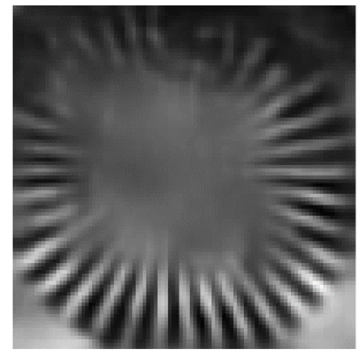
Completion Network

PSNR = 15.87



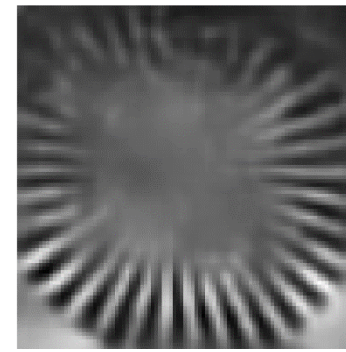
Completion Network trained with Noise

PSNR = 16.02



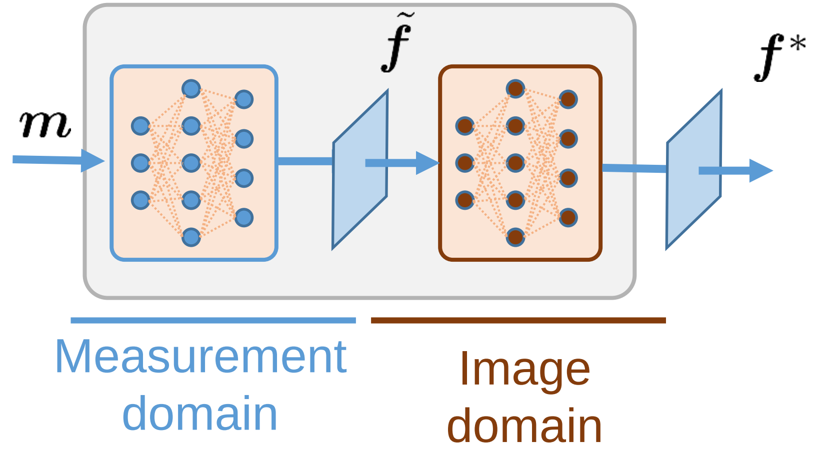
Denoised Completion Network

PSNR = 16.48



LIMITATIONS

- **Processing are sequential (i.e., data domain, followed by measurement domain)**



- **Open questions**
  - Interpretation of the solution

$$\boldsymbol{f}^* = \mathcal{G}(\boldsymbol{m})$$

  - Consistency of the solution

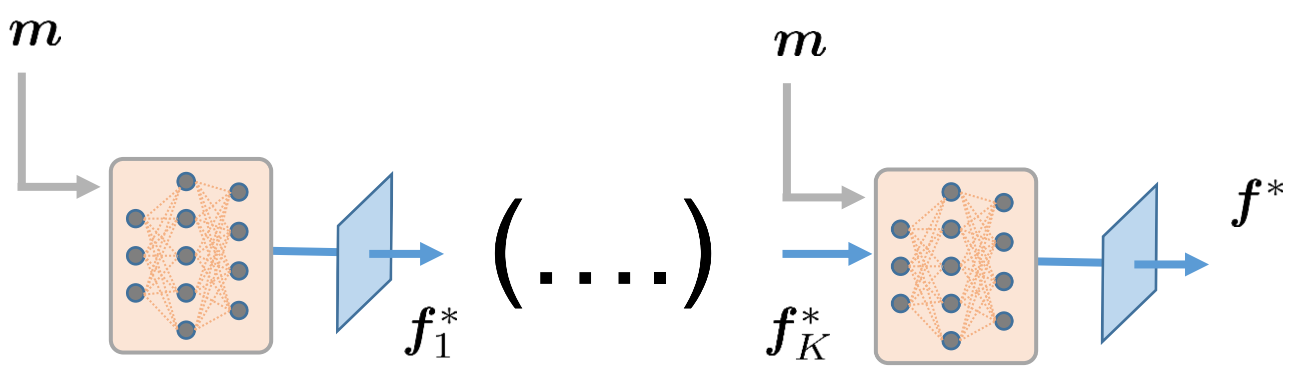
$$\|\boldsymbol{m} - \boldsymbol{P}_1 \boldsymbol{f}^*\|^2 \leq \epsilon$$

  - Link with traditional reconstruction algorithms that solve

$$\min_{\boldsymbol{f}} \; \|\boldsymbol{m} - \boldsymbol{P}_1 \boldsymbol{f}\|^2 + \mathcal{R}(\boldsymbol{f})$$

ITERATIVE SCHEMES

➢ **Data domain and image domain processings are nested**

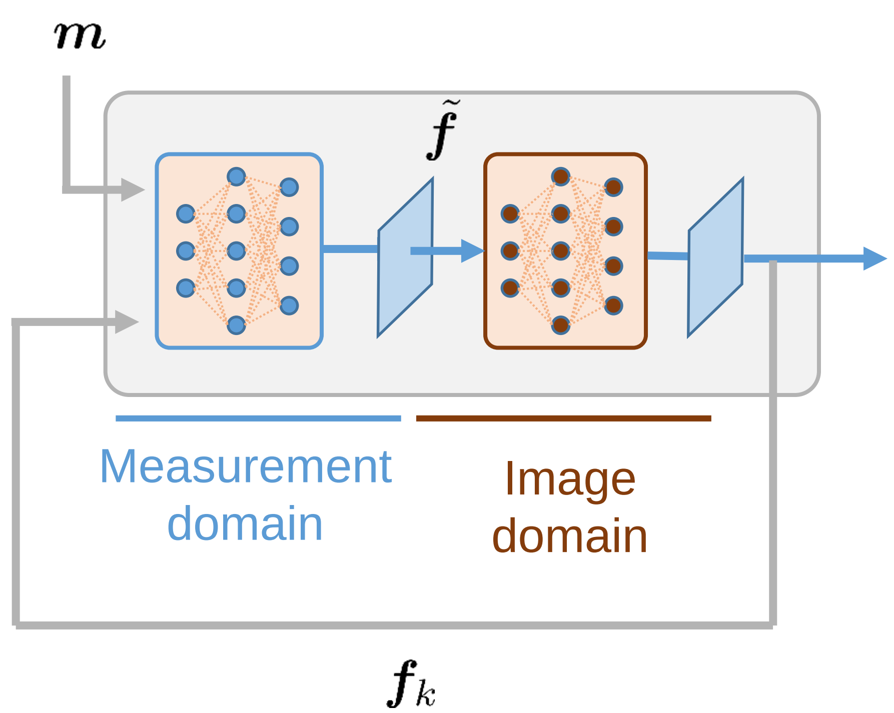


➢ **Many variants**

- Unrolled
- Neumann
- …

DEEP EXPECTATION MAXIMIZATION

- **Simple network architecture, just loop over the previous one**



[Lorente-Mur *et. al*, IEEE ISBI, 2021]

DEEP EXPECTATION MAXIMIZATION

- **We wish to solve**

$$\underset{\boldsymbol{f}}{\operatorname{argmax}} \quad \log p(\boldsymbol{m}^\alpha | \boldsymbol{f}) + \log p(\boldsymbol{f}),$$

Noise Model

$$\mathbf{m}^\alpha \,|\, \mathbf{f} \sim \mathcal{N}(\boldsymbol{P}_1 \boldsymbol{f}, \boldsymbol{\Sigma}_\alpha)$$

$$\mathbf{f} \sim \text{unknown!}$$

Ground Truth

Gaussian

$$p(\boldsymbol{f}) \propto \exp \|\boldsymbol{f}\|_2^2$$

Laplace

$$p(\boldsymbol{f}) \propto \exp \|\boldsymbol{f}\|_1$$

- **We create an image sequence**

$$\bar{\boldsymbol{x}}^{(k)} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{P}_1\boldsymbol{x} - \boldsymbol{m}^\alpha\|^2_{\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\alpha}}{}^{-1}} + \|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{f}^{(k)}\|^2_{\boldsymbol{\Sigma}^{-1}}$$

$$\boldsymbol{f}^{(k+1)} = \underset{\boldsymbol{f}}{\operatorname{argmin}} \|\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{P}\boldsymbol{f}\|^2_{\boldsymbol{\Sigma}^{-1}} - \log p(\boldsymbol{f})$$

Latent measurement

Still unknown!

- **We can learn the unknown term**

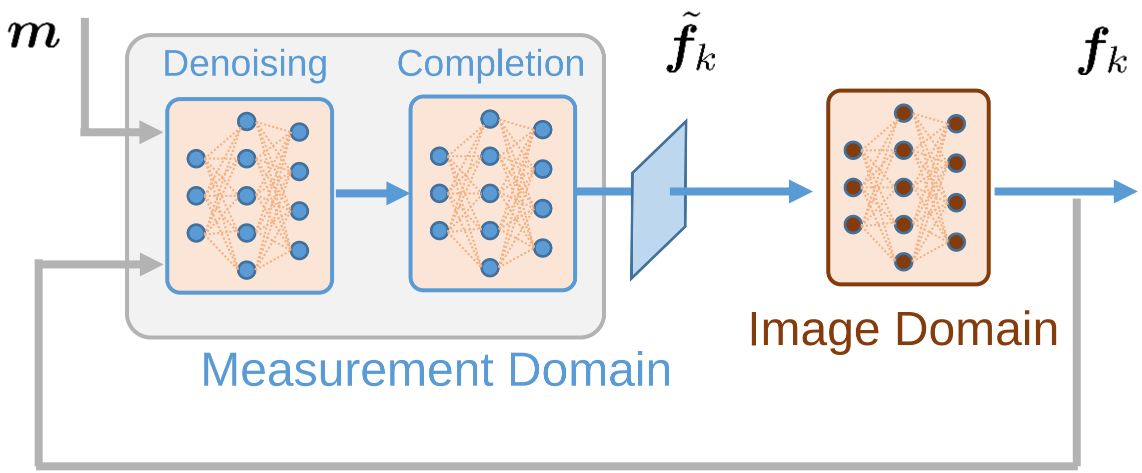
$$\bar{\boldsymbol{x}}^{(k)} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{P}_1\boldsymbol{x} - \boldsymbol{m}^\alpha\|^2_{\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\alpha}}{}^{-1}} + \|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{f}^{(k)}\|^2_{\boldsymbol{\Sigma}^{-1}}$$

$$\boldsymbol{f}^{(k+1)} = \mathcal{D}(\boldsymbol{P}^\top \bar{\boldsymbol{x}}^{(k)})$$

Image domain network

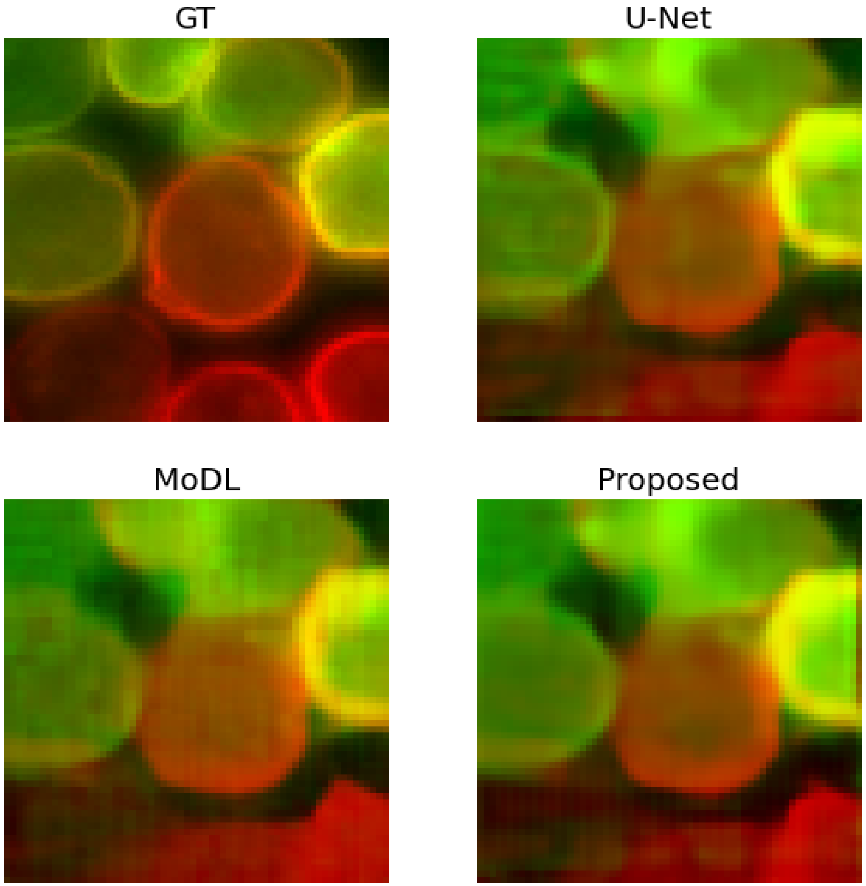
- **We recognize the deep completion architecture**

Denoising (measurements)	$\boldsymbol{y}_1^{(k)} = \boldsymbol{\sigma}_1^2 / (\boldsymbol{\sigma}_1^2 + \tilde{\boldsymbol{\sigma}}_\alpha^2)(\boldsymbol{m}^\alpha - \boldsymbol{P}_1 \boldsymbol{f}^{(k-1)})$
Completion (measurements)	$\boldsymbol{y}_2^{(k)} = \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_1^{-1} \boldsymbol{y}_1^{(k)}$
Update and mapping (image)	$\tilde{\boldsymbol{f}}^{(k)} = \boldsymbol{f}^{(k-1)} + \boldsymbol{P}^\top \boldsymbol{y}^{(k)}$
Denoising (image)	$\boldsymbol{f}^{(k)} = \mathcal{D}(\tilde{\boldsymbol{f}}^{(k)})$

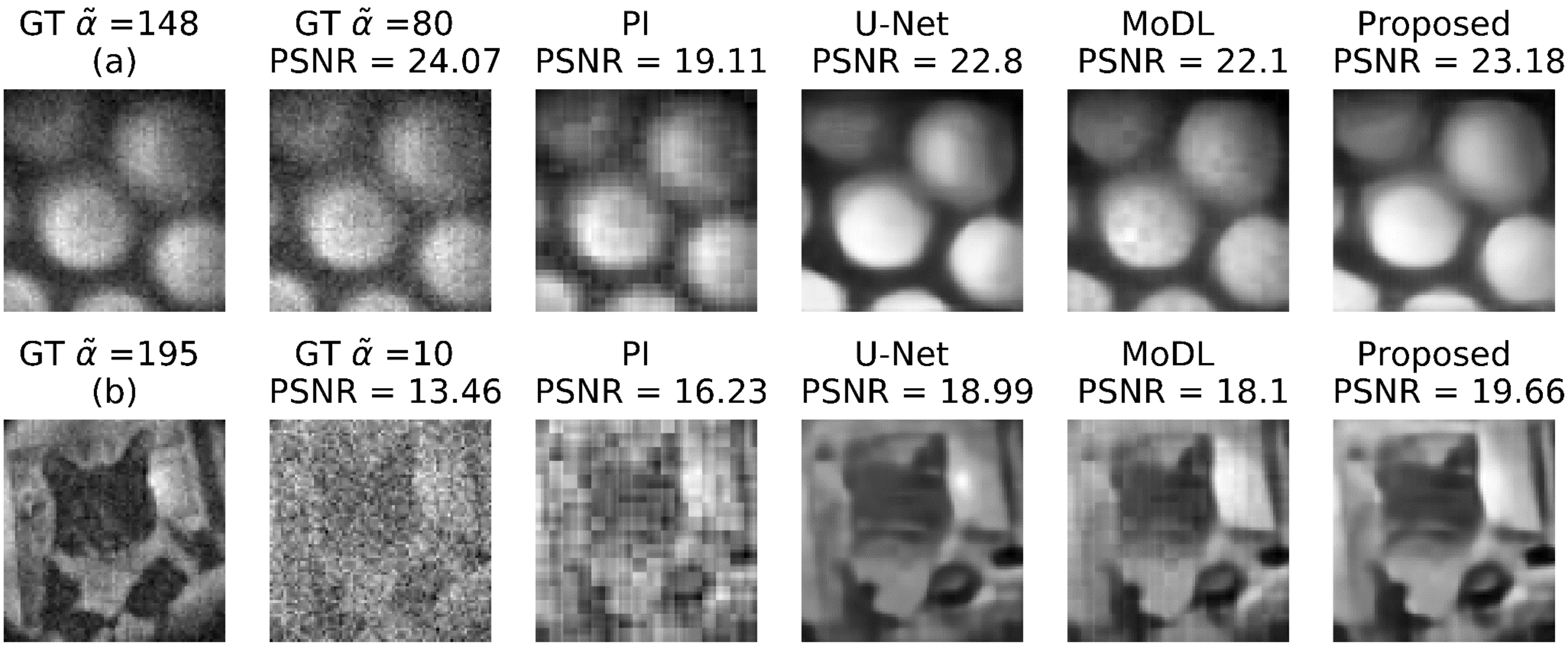


[Lorente-Mur *et. al*, IEEE ISBI, 2021]

SIMULATION RESULTS

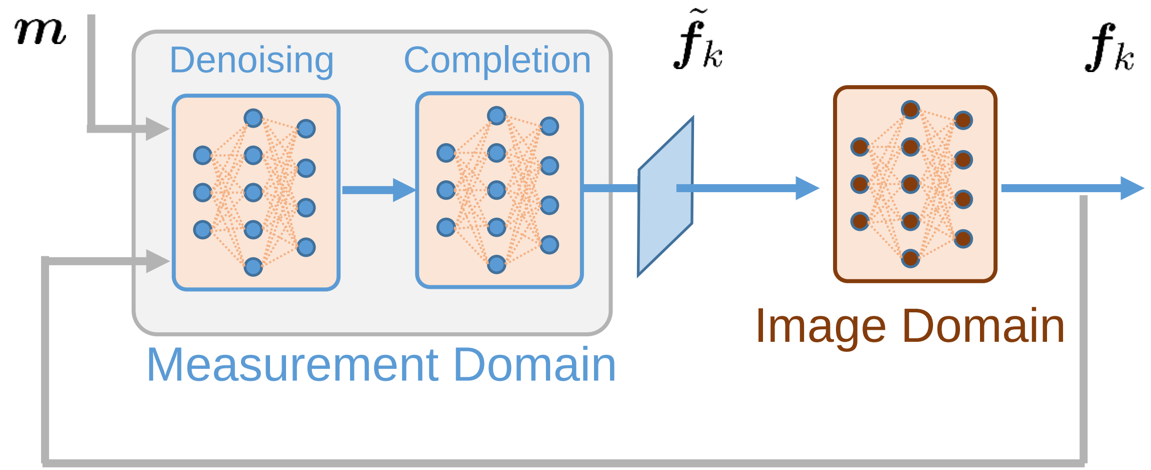


EXPERIMENTAL RESULTS



CONCLUSIONS

- **Deep reconstruction networks can be interpreted as traditional algorithm in a Bayesian framework**



  - Deep completion network, as conditional mean
  - Deep expectation-maximization network

- **Training**
  - Very dependent on the noise level
  - Interpretable architecture are more robust to unseen noise levels
  - Noise Adaptive (behaves well for different noise levels)

OPEN SOURCE

- **SPIRiT: a single-pixel image reconstruction toolbox**
  - Matlab code for statistical completion
    https://github.com/nducros/SPIRIT

- **SPyRiT a single-pixel image reconstruction toolbox (in Python)**
  - Python **package** for traditional and deep reconstruction
    https://github.com/openspyrit/spyrit
  - Python **scripts** for traditional and deep reconstruction
    https://github.com/openspyrit/spyritexamples
  - **Hands-on session** in part of it
    https://github.com/openspyrit/spyritexamples/tree/master/2021_DLMIS_Hands-on

ACKNOWLEGDMENTS

- **Results**
  - Pierre LECLERC
  - Antonio LORENTE MUR
  - Laurent MAHIEU-WILLIAME
  - Bruno MONTCEL
  - Françoise PEYRIN
- **Hands-on**
  - Antonio LORENTE MUR
  - Theo LEULIET
  - Louise Friot-Giroux
  - Thomas GRENIER

See you at the hands-on session!