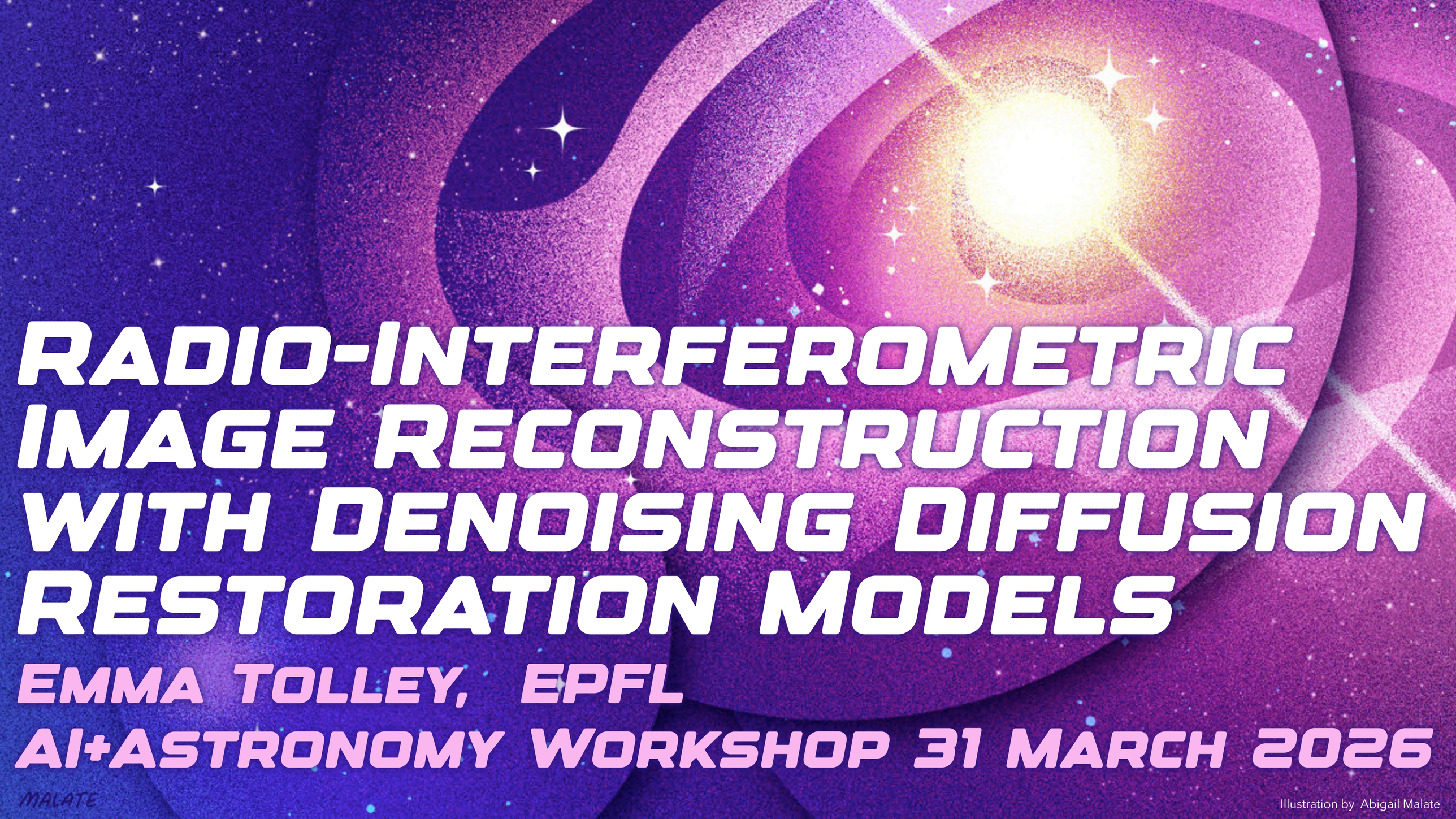


Radio-Interferometric Image Reconstruction with Denoising Diffusion Restoration Models

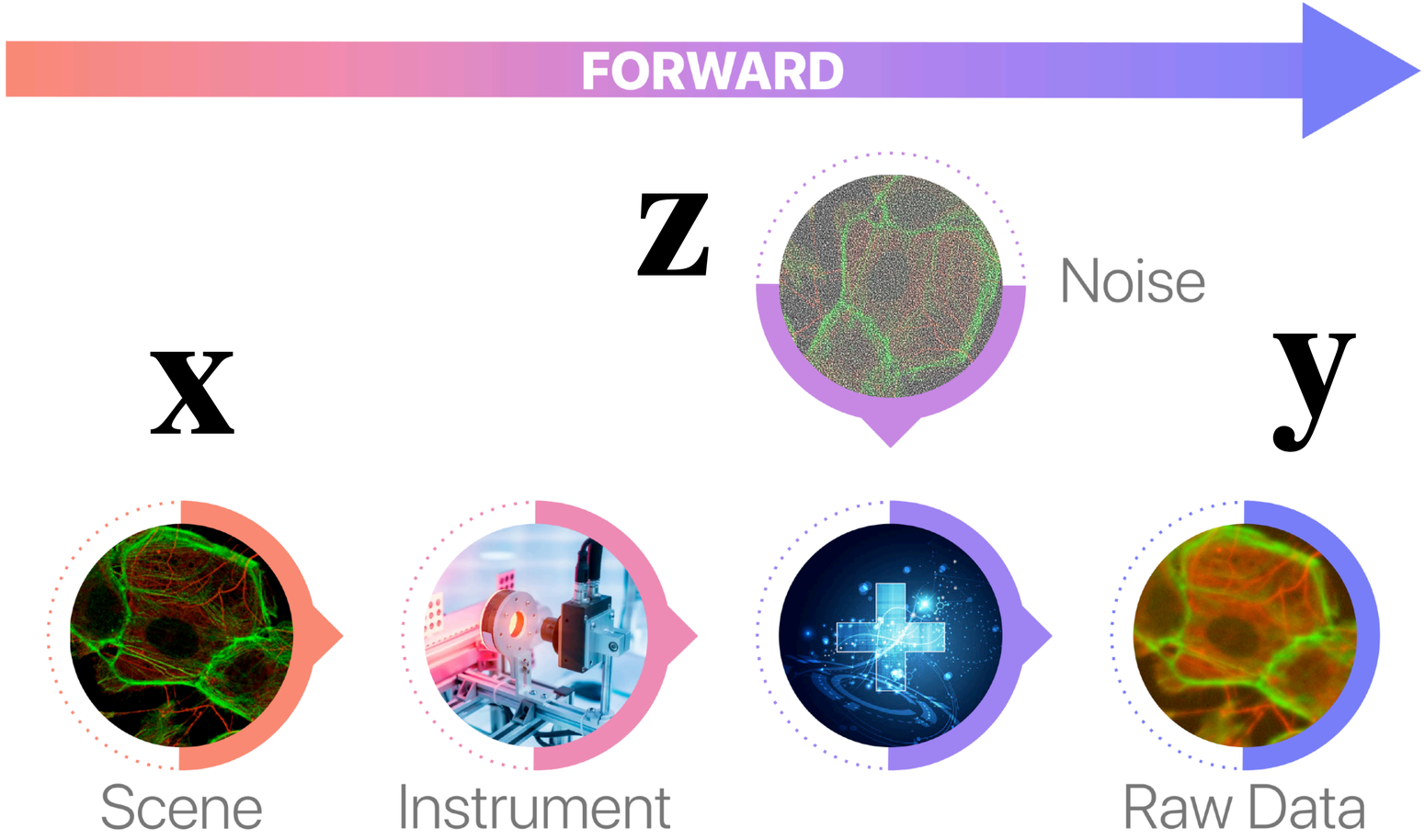
Emma Tolley, EPFL

AI+Astronomy Workshop 31 March 2026

Illustration by Abigail Malate

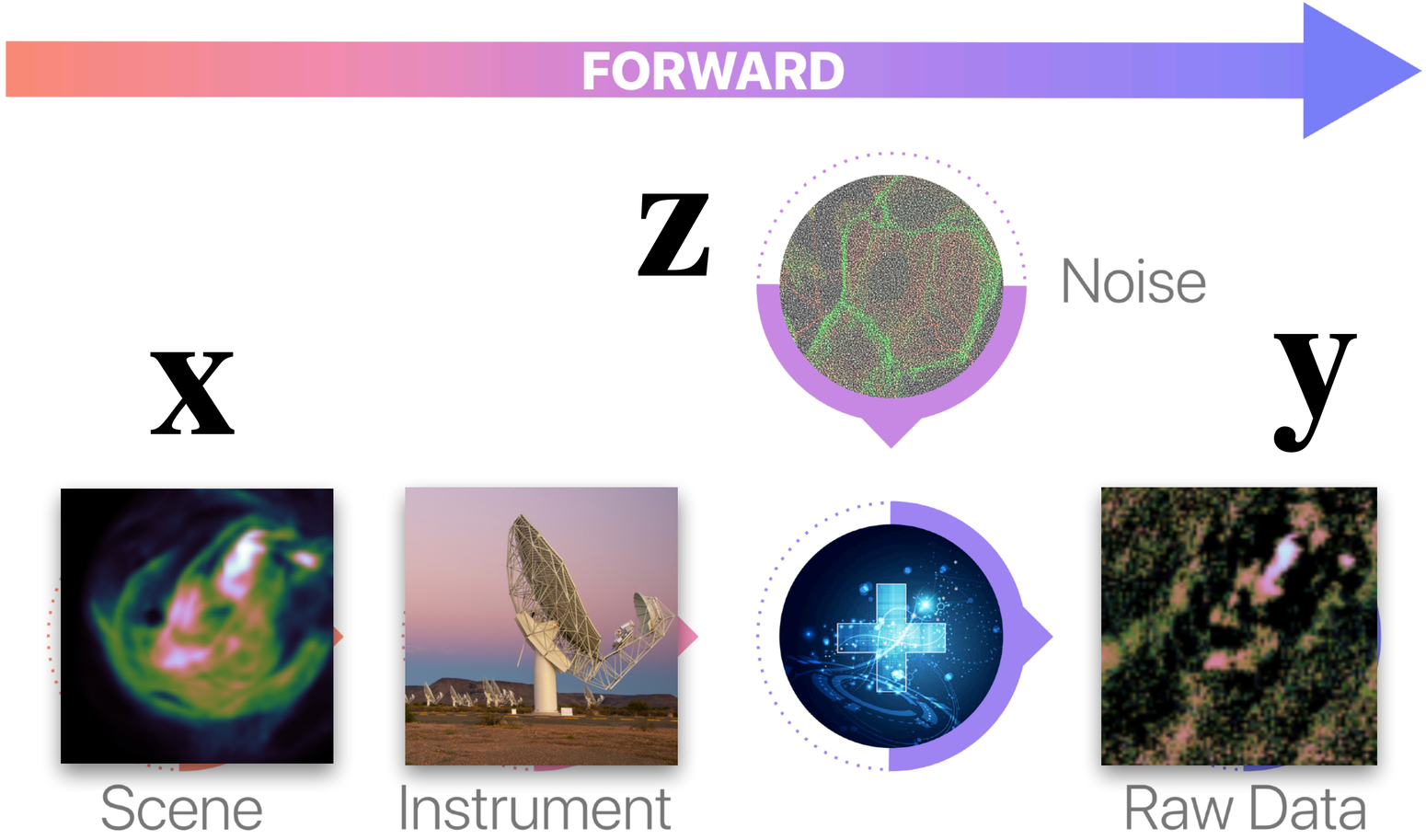
INVERSE PROBLEMS

pyxu-org.github.io



$$y = Hx + z$$

Inverse problems in radio astronomy

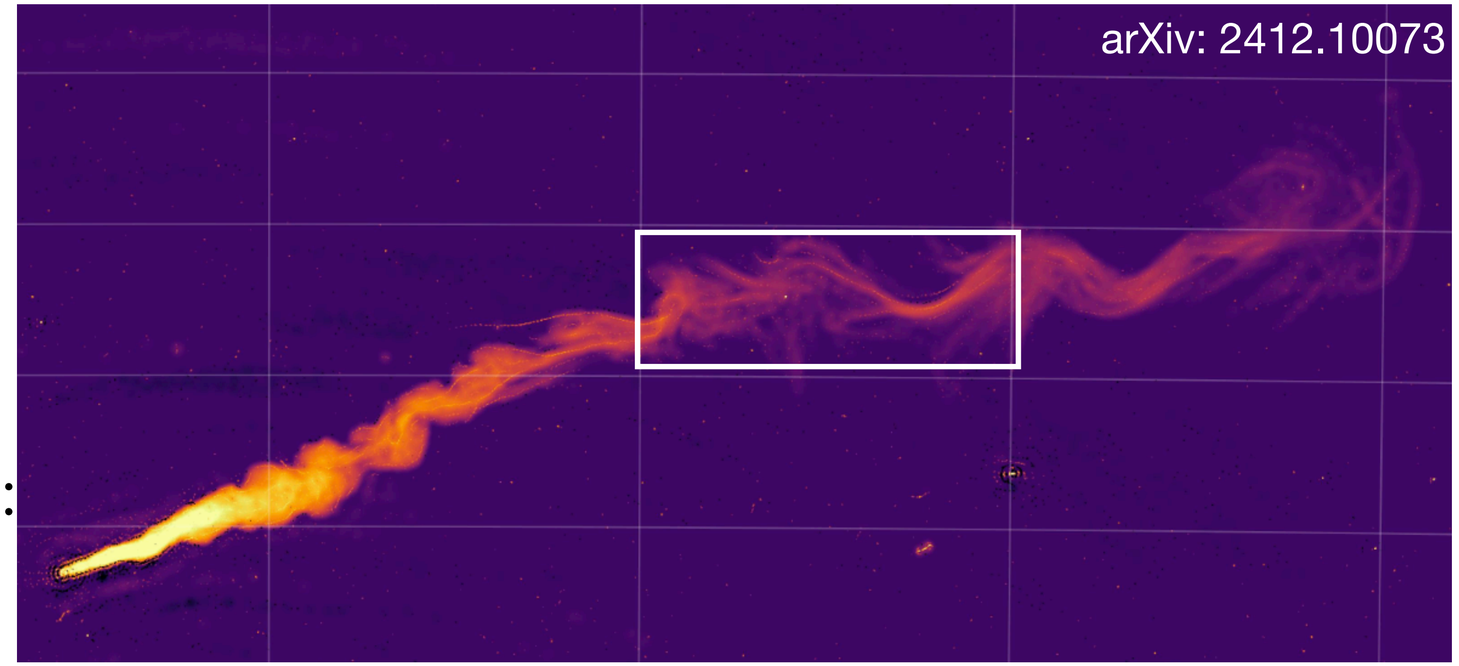


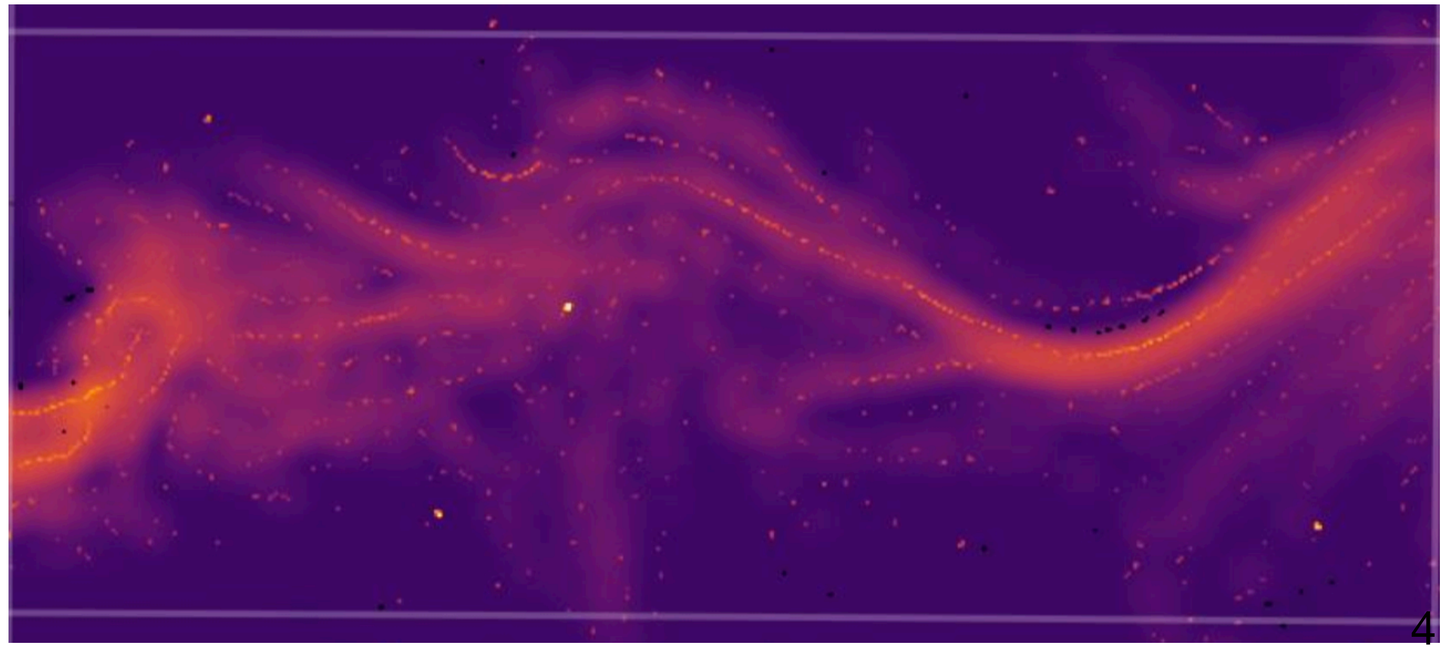
$$y = Hx + z$$

Inverse problems in radio astronomy

CLEAN algorithms:

- Create an empty estimate $\hat{\mathbf{x}}$
- Iteratively add discrete components to $\hat{\mathbf{x}}$ consistent with $\mathbf{y}$ using the forward model: $\hat{\mathbf{y}} = \mathbf{H}\hat{\mathbf{x}}$

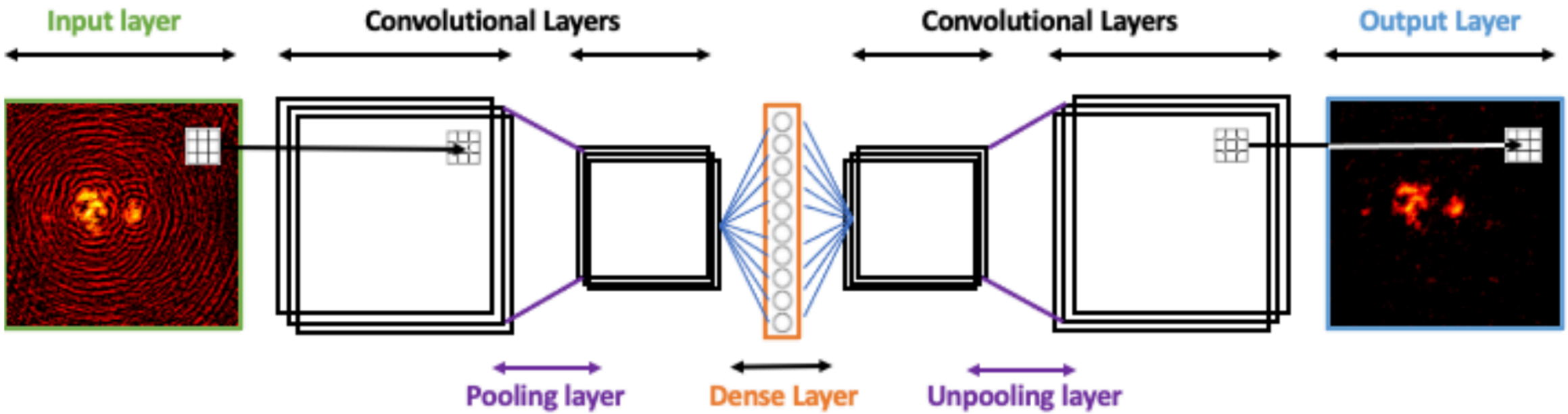




Problem: sky is not always made of sparse discrete components!

Inverse problems in radio astronomy

Machine learning approaches: learn to predict $\mathbf{x}$ from $\mathbf{y}$
with a **conditional generative model**

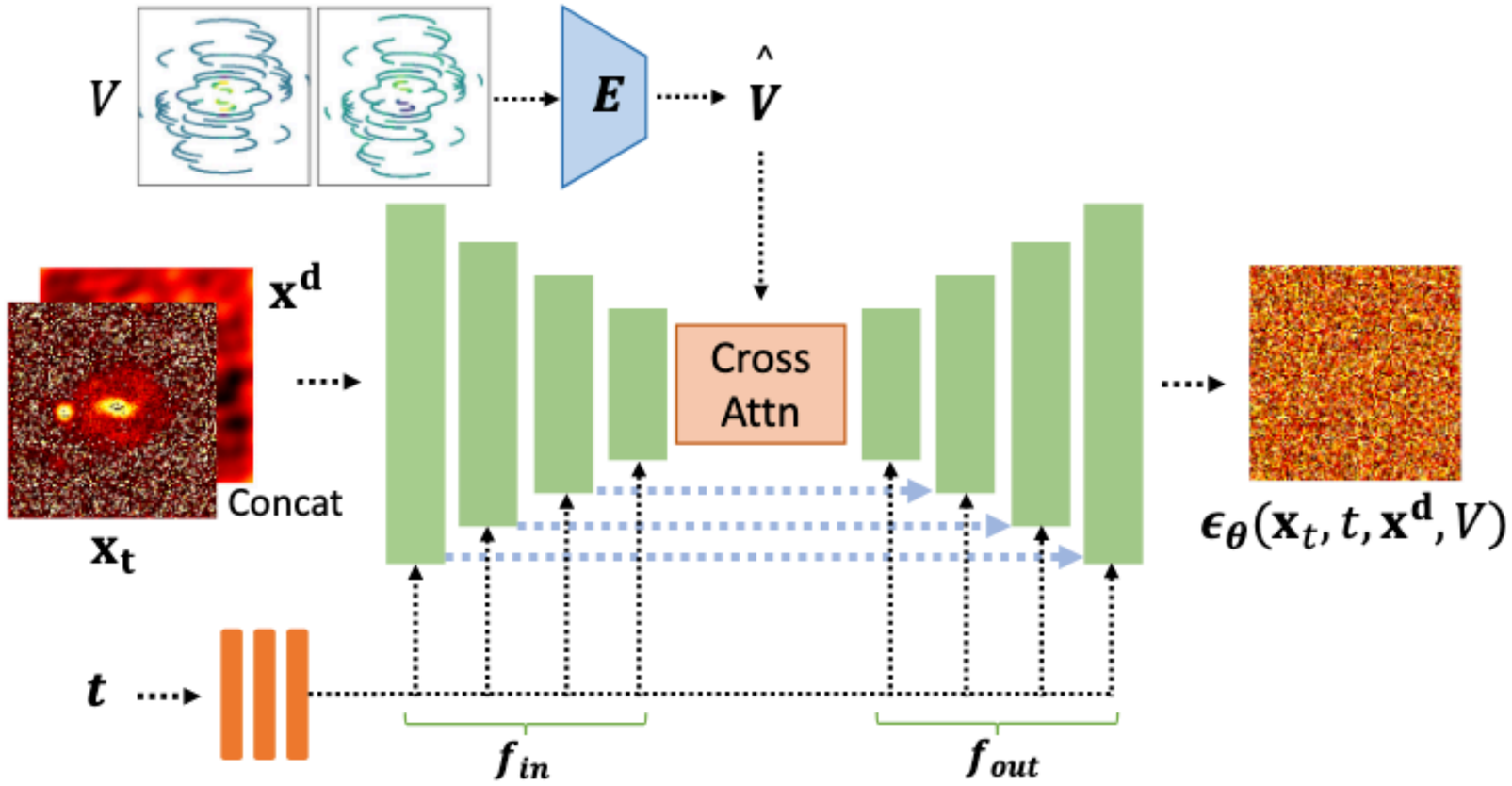


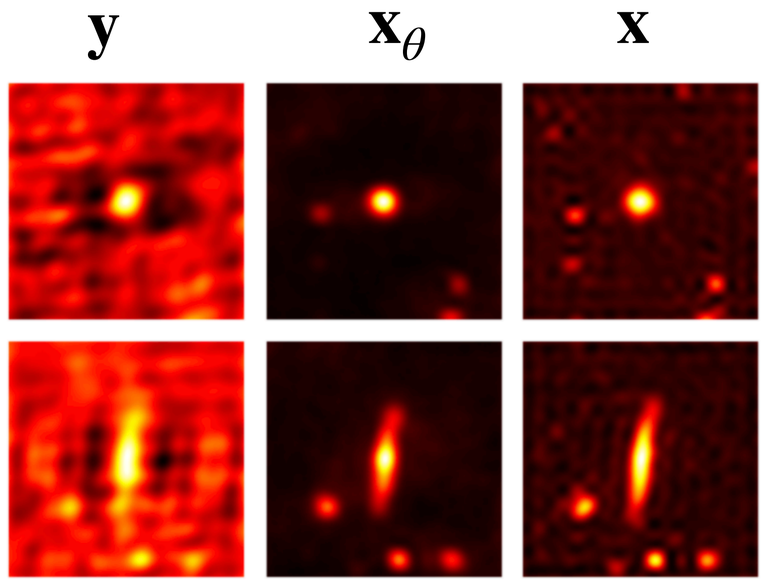
Convolutional Deep Denoising Autoencoders for Radio Astronomical Images

C. Gheller[1], F. Vazza[3,2,1⋆]

Inverse problems in radio astronomy

Machine learning approaches: learn to predict $\mathbf{x}$ from $\mathbf{y}$ with a **conditional generative model**



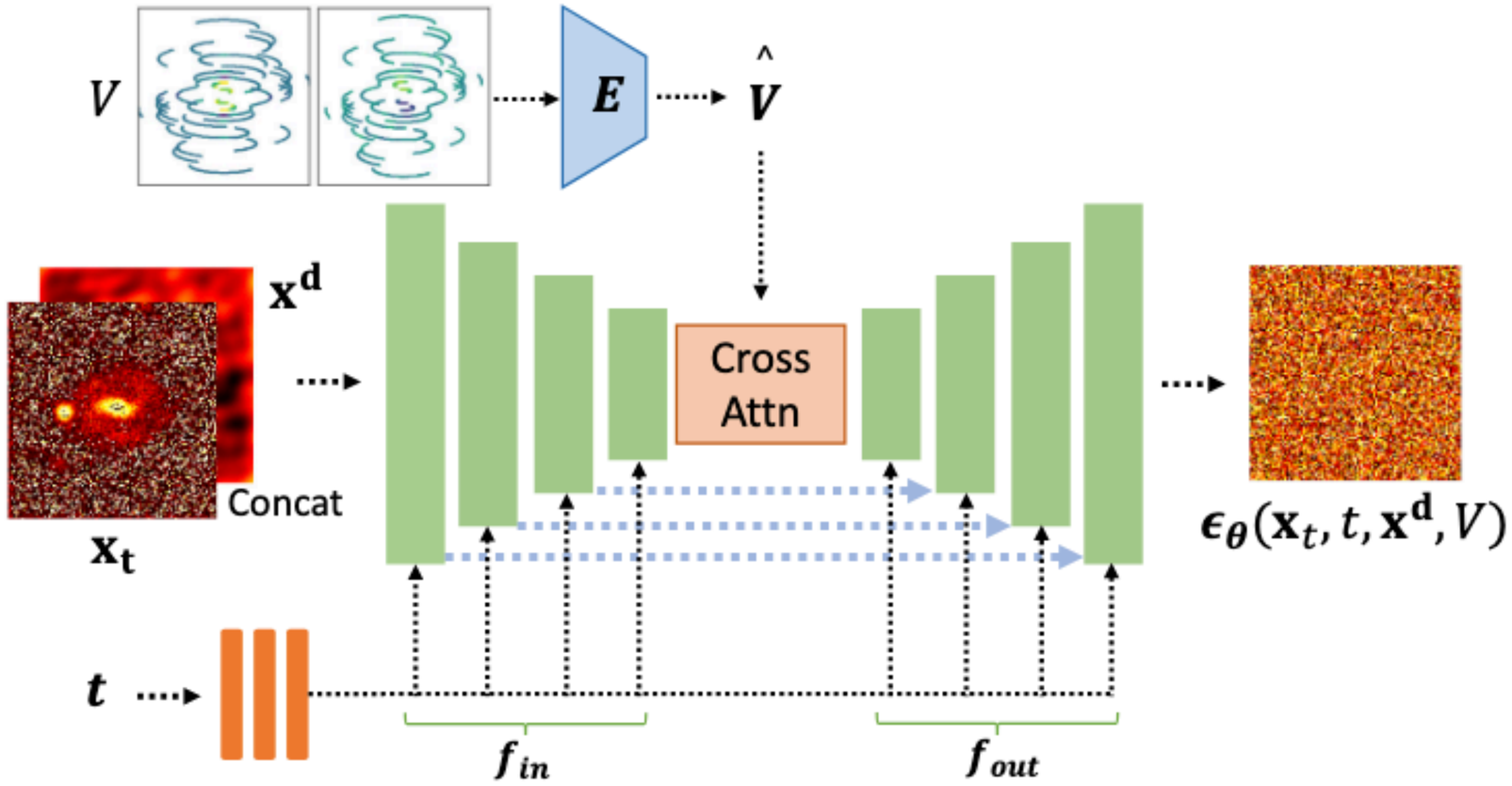


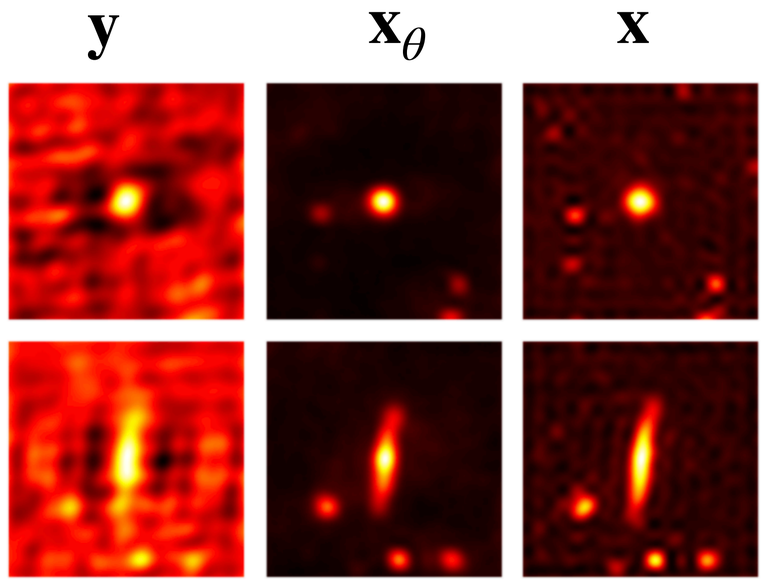
A Conditional Denoising Diffusion Probabilistic Model for Radio Interferometric Image Reconstruction

Ruoqi Wang[a], Zhuoyang Chen[a], Qiong Luo[a,*] and Feng Wang[b]

Inverse problems in radio astronomy

Machine learning approaches: learn to predict $\mathbf{x}$ from $\mathbf{y}$ with a **conditional generative model**





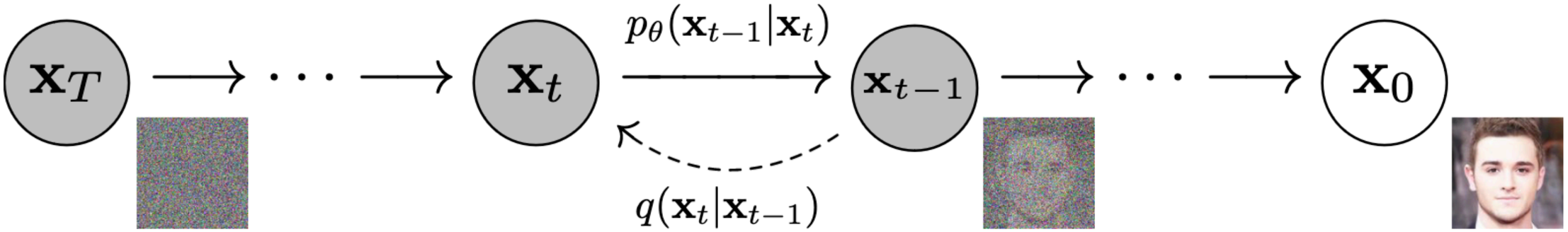
Problem: Network learns $\mathbf{H}$, but $\mathbf{H}$ depends on instrument, observation time, pointing direction, etc etc => retrain for every new instrument & observation configuration

Inverse problems in radio astronomy

Can we have the best of both worlds?

Goal: Use AI to learn a data-driven prior of the sky without needing to learn the form of the operator $\mathbf{H}$

Denoising Diffusion Probablistic Models (DDPM)

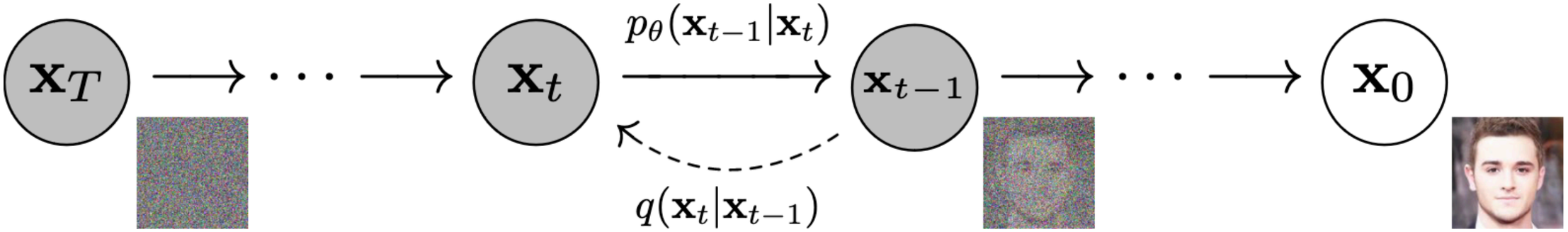


Forward process:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

arXiv:2006.11239

Denoising Diffusion Probablistic Models (DDPM)



Forward process:

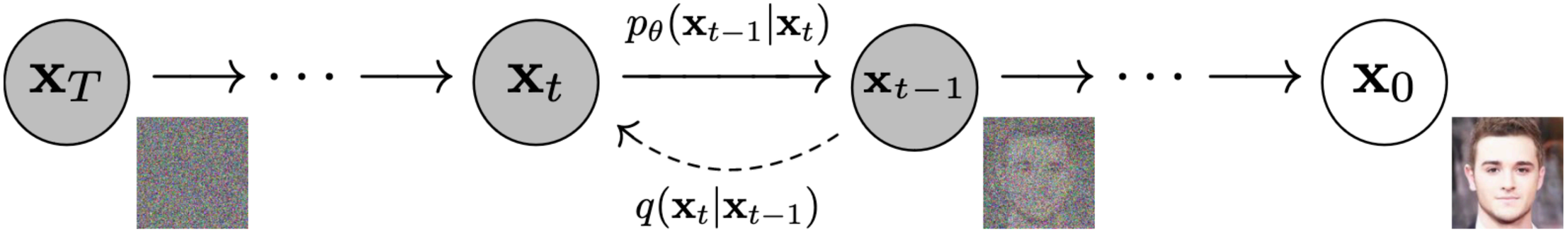
$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Reverse process:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

arXiv:2006.11239

Denoising Diffusion Probablistic Models (DDPM)



Forward process:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Reverse process:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Training is performed by optimizing the usual variational bound on negative log likelihood:

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

arXiv:2006.11239

Denoising Diffusion Probablistic Models (DDPM)

Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training

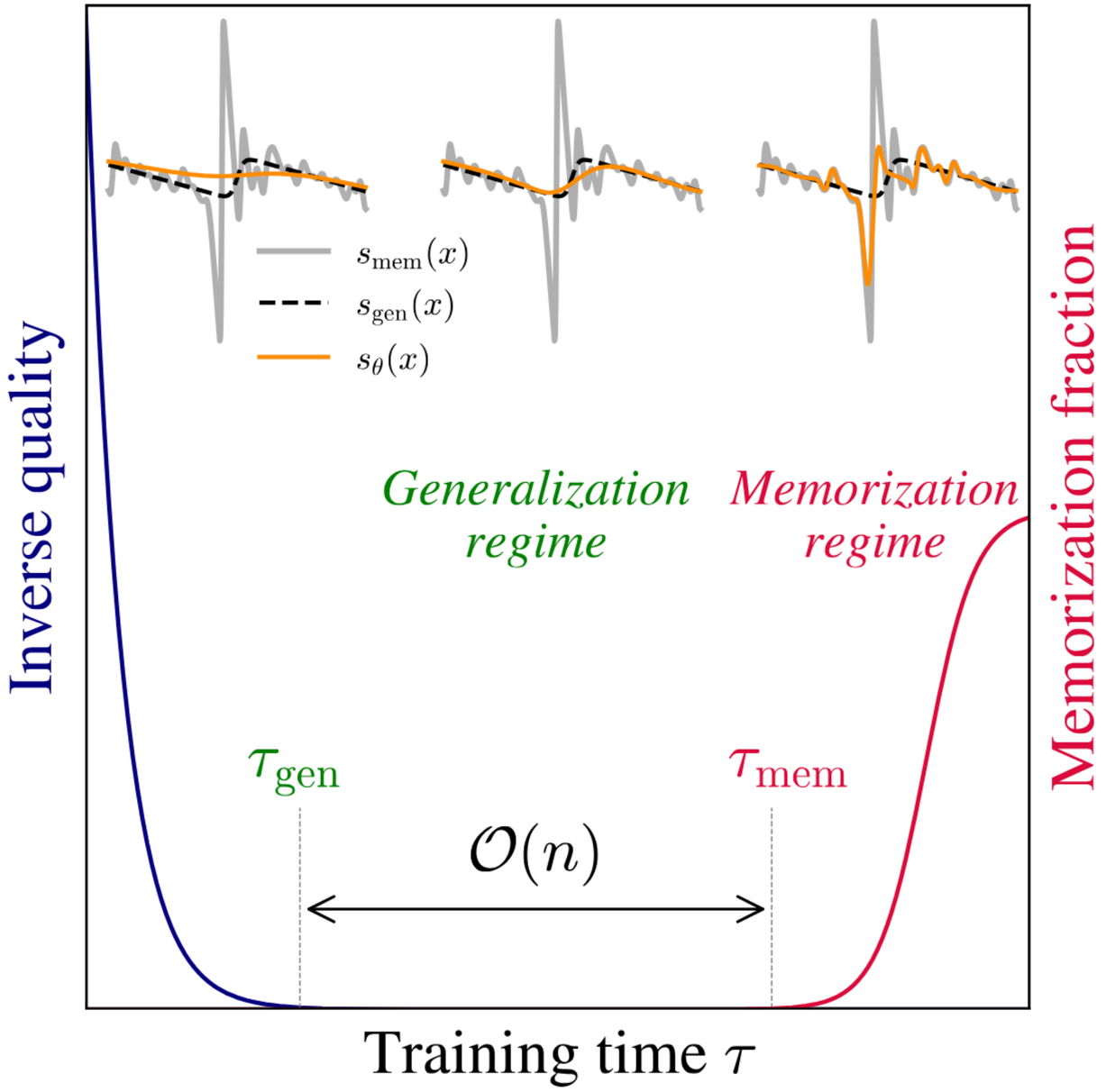
Tony Bonnaire[†]
LPENS
Université PSL, Paris
tony.bonnaire@phys.ens.fr

Raphaël Urfin[†]
LPENS
Université PSL, Paris
raphael.urfin@phys.ens.fr

Giulio Biroli
LPENS
Université PSL, Paris
giulio.biroli@phys.ens.fr

Marc Mézard
Department of Computing Sciences
Bocconi University, Milano
marc.mezard@unibocconi.it

"[Diffusion models have a] form of implicit dynamical regularization in the training dynamics, which allow to avoid memorization even in highly overparameterized settings"



Denoising Diffusion Probablistic Models (DDPM)

Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training

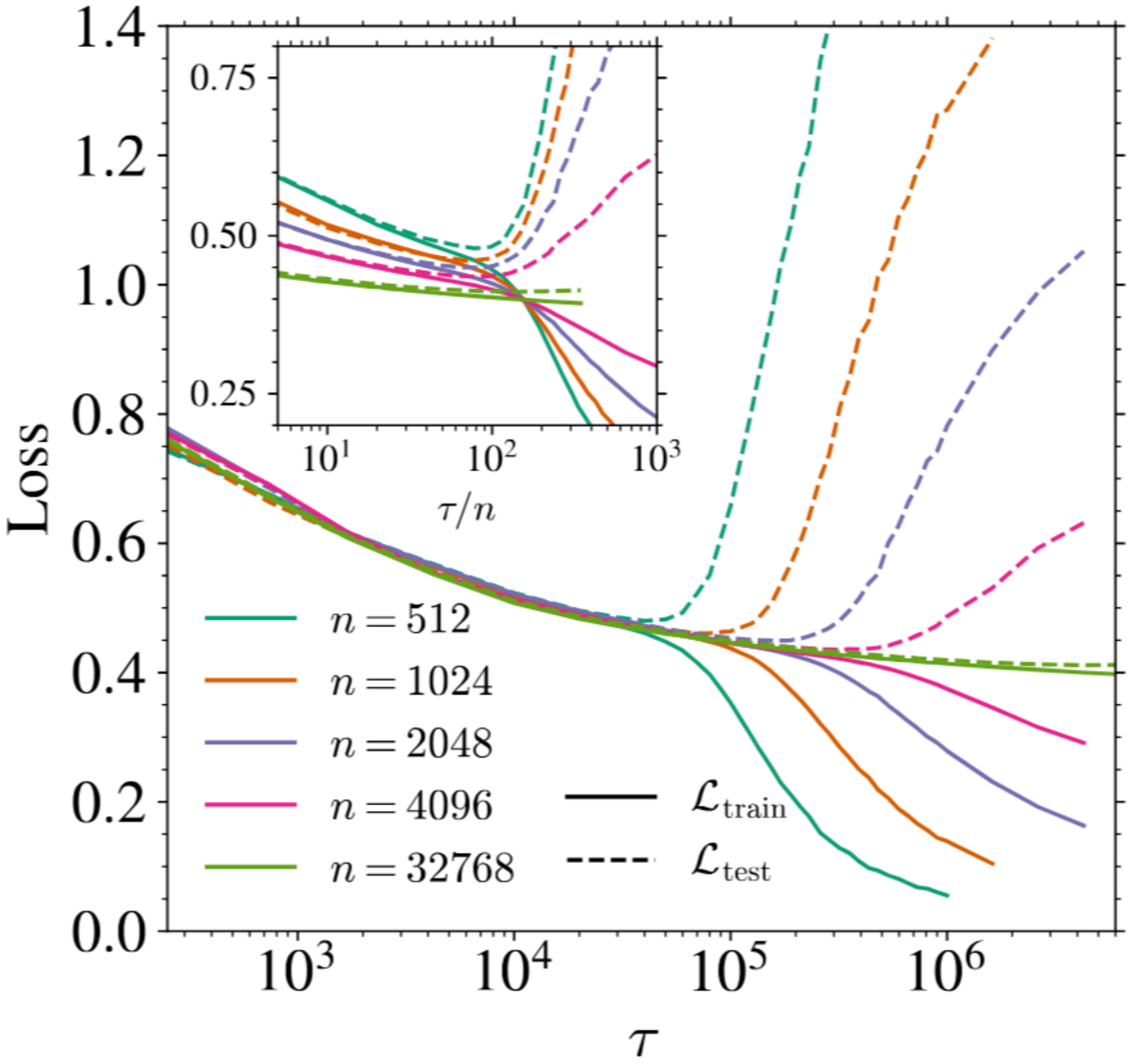
Tony Bonnaire[†]
LPENS
Université PSL, Paris
tony.bonnaire@phys.ens.fr

Raphaël Urfin[†]
LPENS
Université PSL, Paris
raphael.urfin@phys.ens.fr

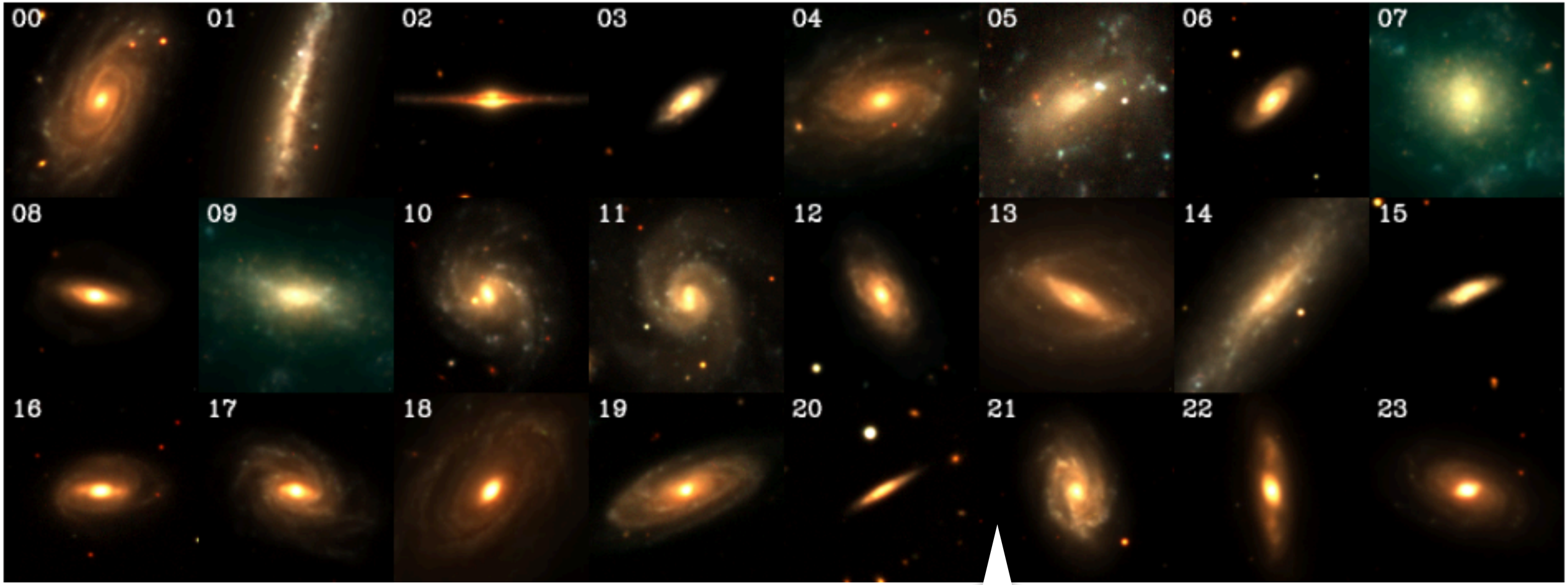
Giulio Biroli
LPENS
Université PSL, Paris
giulio.biroli@phys.ens.fr

Marc Mézard
Department of Computing Sciences
Bocconi University, Milano
marc.mezard@unibocconi.it

"[Diffusion models have a] form of implicit dynamical regularization in the training dynamics, which allow to avoid memorization even in highly overparameterized settings"



Galaxy generation with DDPM

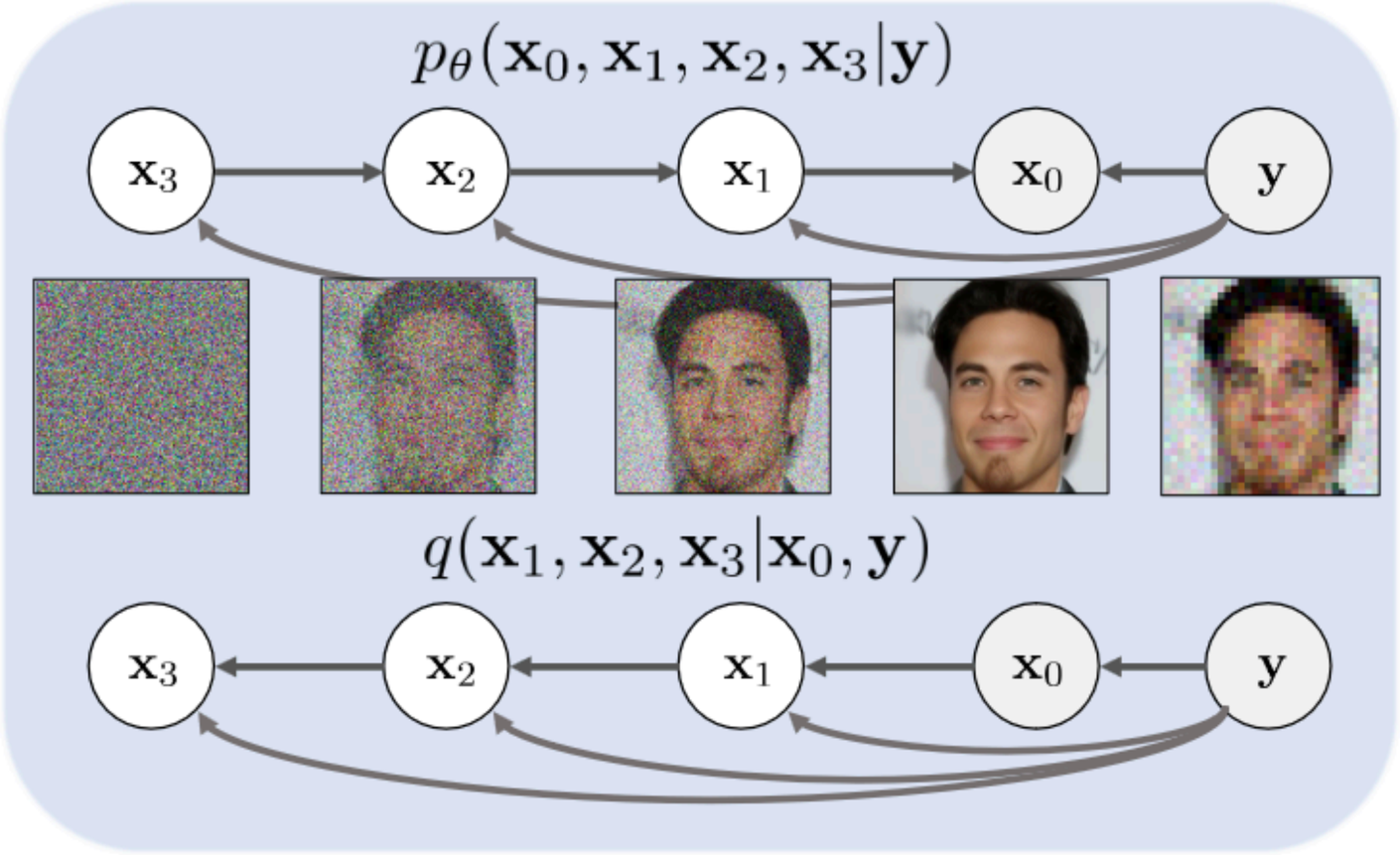


Realistic galaxy image simulation via score-based generative models

Michael J. Smith[1,2]★, James E. Geach[1,2], Ryan A. Jackson[3], Nikhil Arora[4], Connor Stone[4] and Stéphane Courteau[4]

Does a good job learning the prior $p(\mathbf{x})$

Denoising Diffusion Restoration Models (DDRM)



Denoising Diffusion Restoration Models
(Dependent on inverse problem)

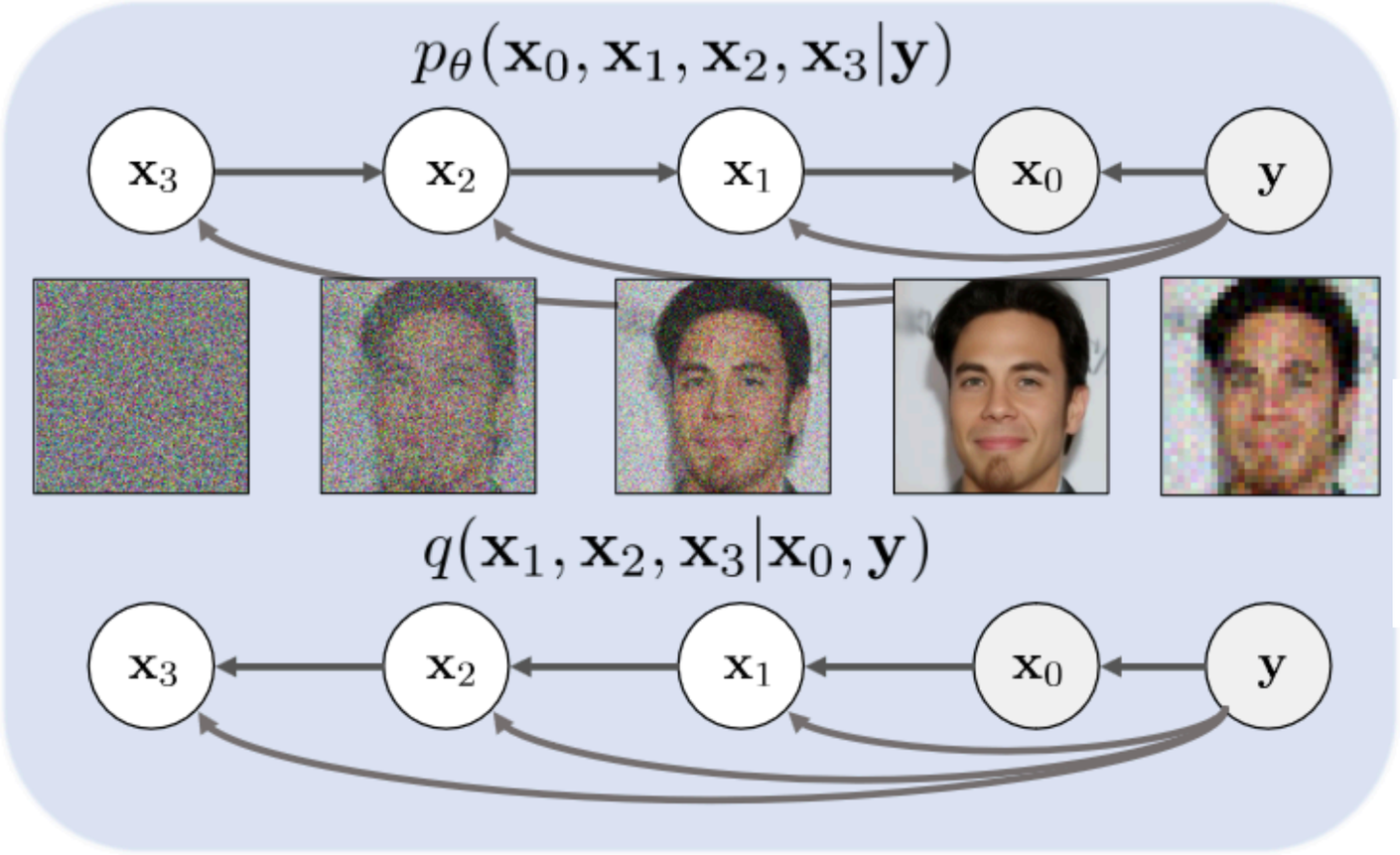
DDRM is a Markov chain conditioned on $\mathbf{y}$

Uses a pre-trained DDPM as a prior on $\mathbf{x}$

$\mathbf{x}$ and $\mathbf{y}$ are related by a linear forward model:

$$\mathbf{y} = \boldsymbol{H}\mathbf{x} + \mathbf{z}$$

Denoising Diffusion Restoration Models (DDRM)



Denoising Diffusion Restoration Models
(Dependent on inverse problem)

DDRM corresponds to the evidence lower bound:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \mathbf{y} \sim q(\mathbf{y}|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{y})] \\
\geq\ &-\mathbb{E}\left[\sum_{t=1}^{T-1} D_{\mathrm{KL}}(q^{(t)}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) \| p_\theta^{(t)}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}))\right] \\
&+\mathbb{E}\left[\log p_\theta^{(0)}(\mathbf{x}_0|\mathbf{x}_1, \mathbf{y})\right] \\
&-\mathbb{E}[D_{\mathrm{KL}}(q^{(T)}(\mathbf{x}_T|\mathbf{x}_0, \mathbf{y}) \| p_\theta^{(T)}(\mathbf{x}_T|\mathbf{y}))]
\end{aligned}$$

Denoising Diffusion Restoration Models (DDRM)

Linear inverse problem:

$$\mathbf{y} = \boldsymbol{H}\mathbf{x} + \mathbf{z}$$

Find the **Singular Value Decomposition** (SVD) of H:

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

U and V are orthogonal matrixes, S is a diagonal matrix containing singular values of H

Transform the measurement and and outputs:

$$\bar{\mathbf{x}} \equiv \mathbf{V}^\top \mathbf{x} \qquad \bar{\mathbf{y}} \equiv \mathbf{S}^+\mathbf{U}^\top \mathbf{y}$$

$$= \mathbf{S}^+\mathbf{U}^\top(\mathbf{H}\mathbf{x} + \mathbf{z})$$

$$= \mathbf{S}^+\mathbf{U}^\top(\mathbf{U}\mathbf{S}\mathbf{V}^\top\mathbf{x} + \mathbf{z})$$

$$= \mathbf{S}^+\mathbf{S}\mathbf{V}^\top\mathbf{x} + \mathbf{S}^+\mathbf{U}^\top\mathbf{z}$$

$$= \mathbf{S}^+\mathbf{S}\bar{\mathbf{x}} + \mathbf{S}^+\mathbf{U}^\top\mathbf{z}$$

This transformation map x and y to equivalent space

Denoising Diffusion Restoration Models (DDRM)

We have our pre-trained DDPM: $f_\theta(\mathbf{x}_{t+1}) = \mathbf{x}_{\theta,t}$

Define $\bar{\mathbf{x}}_{\theta,t}^{(i)}$ as the ith index of $\bar{\mathbf{x}}_{\theta,t} = \mathbf{V}^\top \mathbf{x}_{\theta,t}$

Denoising Diffusion Restoration Models (DDRM)

We have our pre-trained DDPM: $f_\theta(\mathbf{x}_{t+1}) = \mathbf{x}_{\theta,t}$

Define $\bar{\mathbf{x}}_{\theta,t}^{(i)}$ as the ith index of $\bar{\mathbf{x}}_{\theta,t} = \mathbf{V}^\top \mathbf{x}_{\theta,t}$

Define the Markov Chain updates as:

s_i are the singular values of H

$$p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)} | \mathbf{x}_{t+1}, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \end{cases}$$

No information from the observation $\bar{\mathbf{y}}$, update step comes fully from the DDPM

Denoising Diffusion Restoration Models (DDRM)

We have our pre-trained DDPM: $f_\theta(\mathbf{x}_{t+1}) = \mathbf{x}_{\theta,t}$

Define $\bar{\mathbf{x}}_{\theta,t}^{(i)}$ as the ith index of $\bar{\mathbf{x}}_{\theta,t} = \mathbf{V}^\top \mathbf{x}_{\theta,t}$

Define the Markov Chain updates as:

s_i are the singular values of H

$$p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)} | \mathbf{x}_{t+1}, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{\mathbf{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\mathbf{y}}}{s_i} \end{cases}$$

We have measurements $\mathbf{y}$ but they are noisy, update using DDPM and $\mathbf{y}$

Denoising Diffusion Restoration Models (DDRM)

We have our pre-trained DDPM: $f_\theta(\mathbf{x}_{t+1}) = \mathbf{x}_{\theta,t}$

Define $\bar{\mathbf{x}}_{\theta,t}^{(i)}$ as the ith index of $\bar{\mathbf{x}}_{\theta,t} = \mathbf{V}^\top \mathbf{x}_{\theta,t}$

Define the Markov Chain updates as:

s_i are the singular values of H

$$p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)} | \mathbf{x}_{t+1}, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{\mathbf{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\mathbf{y}}}{s_i} \\ \mathcal{N}((1-\eta_b)\bar{\mathbf{x}}_{\theta,t}^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_{\mathbf{y}}}{s_i} \end{cases}$$

We have measurements $\mathbf{y}$ which are not noisy, update strongly from $\mathbf{y}$ (controlled by η_b). We use $\eta_b = 1$

DDRM FOR RADIO ASTRONOMY

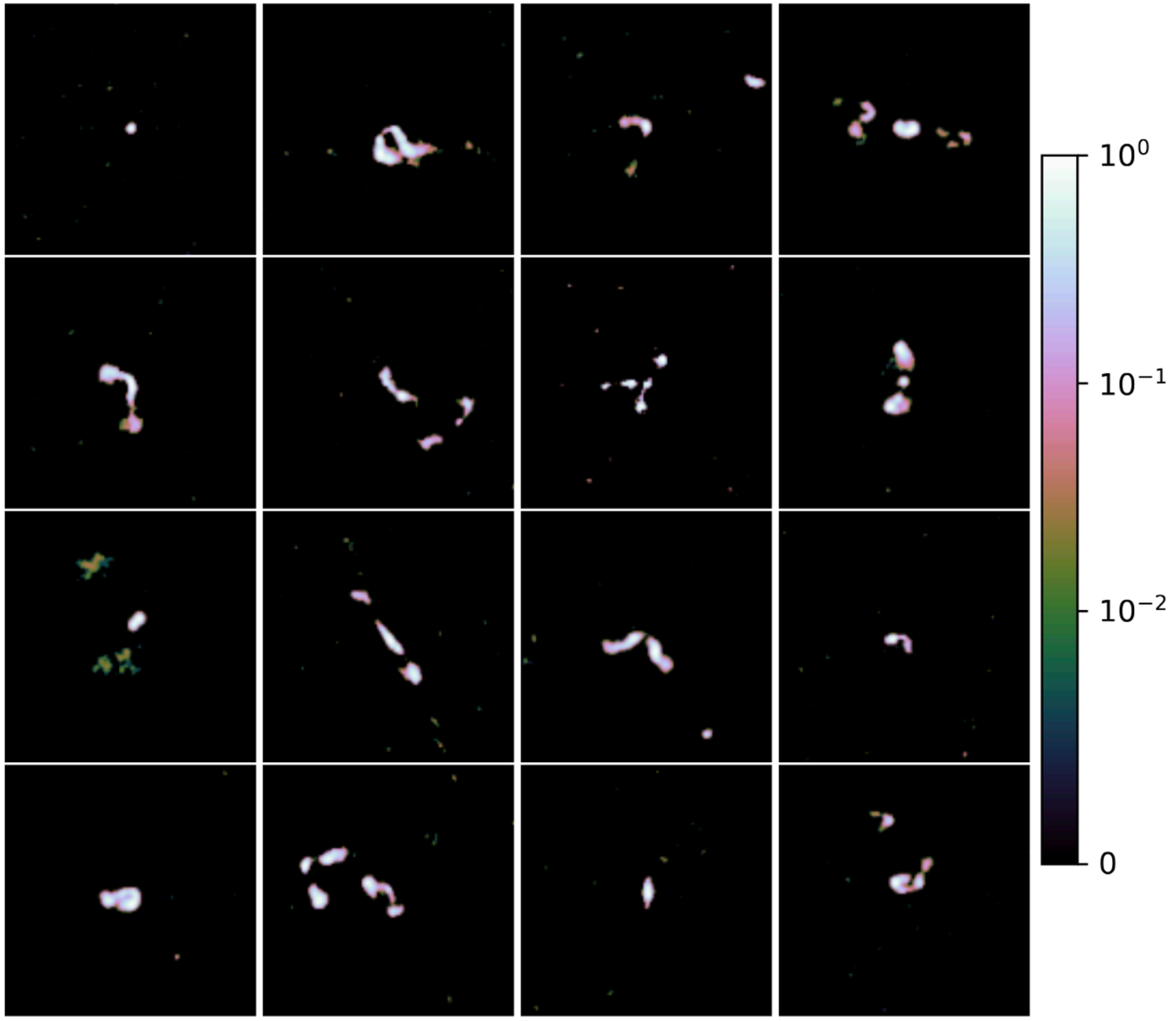
Radio-Interferometric Image Reconstruction with Denoising Diffusion Restoration Models

M. Morales,[1] E. Tolley,[1]★ R. Poitevineau[1]
[1] *Institute of Physics, Laboratory of Astrophysics, École Polytechnique Fédérale de Lausanne (EPFL), 1290 Sauverny, Switzerland*

Submitted to RASTI

DDRM FOR RADIO ASTRONOMY

First, train a DDPM on radio galaxies from the VLA FIRST survey (~20k 150x150 galaxies), reserve 200 for validation & 200 for test



https://arxiv.org/pdf/2601.15844

DDRM FOR RADIO ASTRONOMY

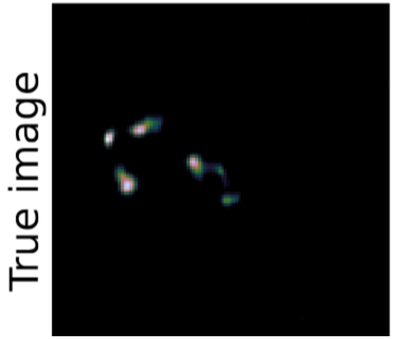
Radio astronomy forward model: $\mathbf{y} = \underbrace{\mathbf{S}\mathcal{F}}_{\mathbf{H}}\mathbf{x} + \mathbf{z}$

Start by finding SVD of $\mathbf{S}$: $\mathbf{S} = \mathbf{I}\Sigma\mathbf{P}$

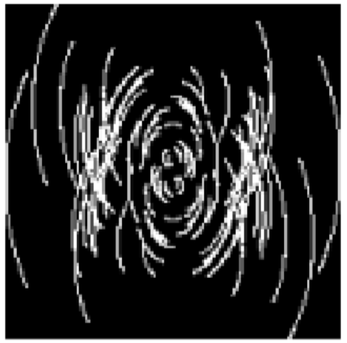
H is complex so we define DDRM using complex SVD:

$$\mathbf{H} = \mathbf{I}\Sigma\mathbf{V}^*, \quad \mathbf{V}^* = \mathbf{P}\mathcal{F}$$

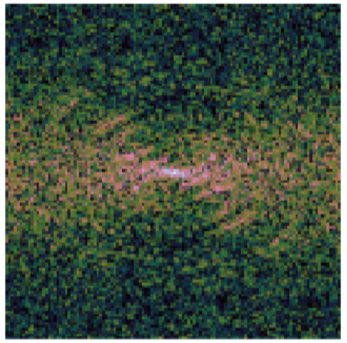
Run DDRM sampling using our pre-trained DDPM



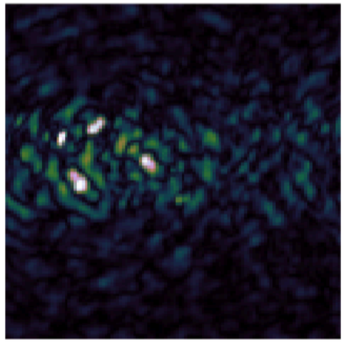
$\mathbf{x}$



$\mathbf{S}$



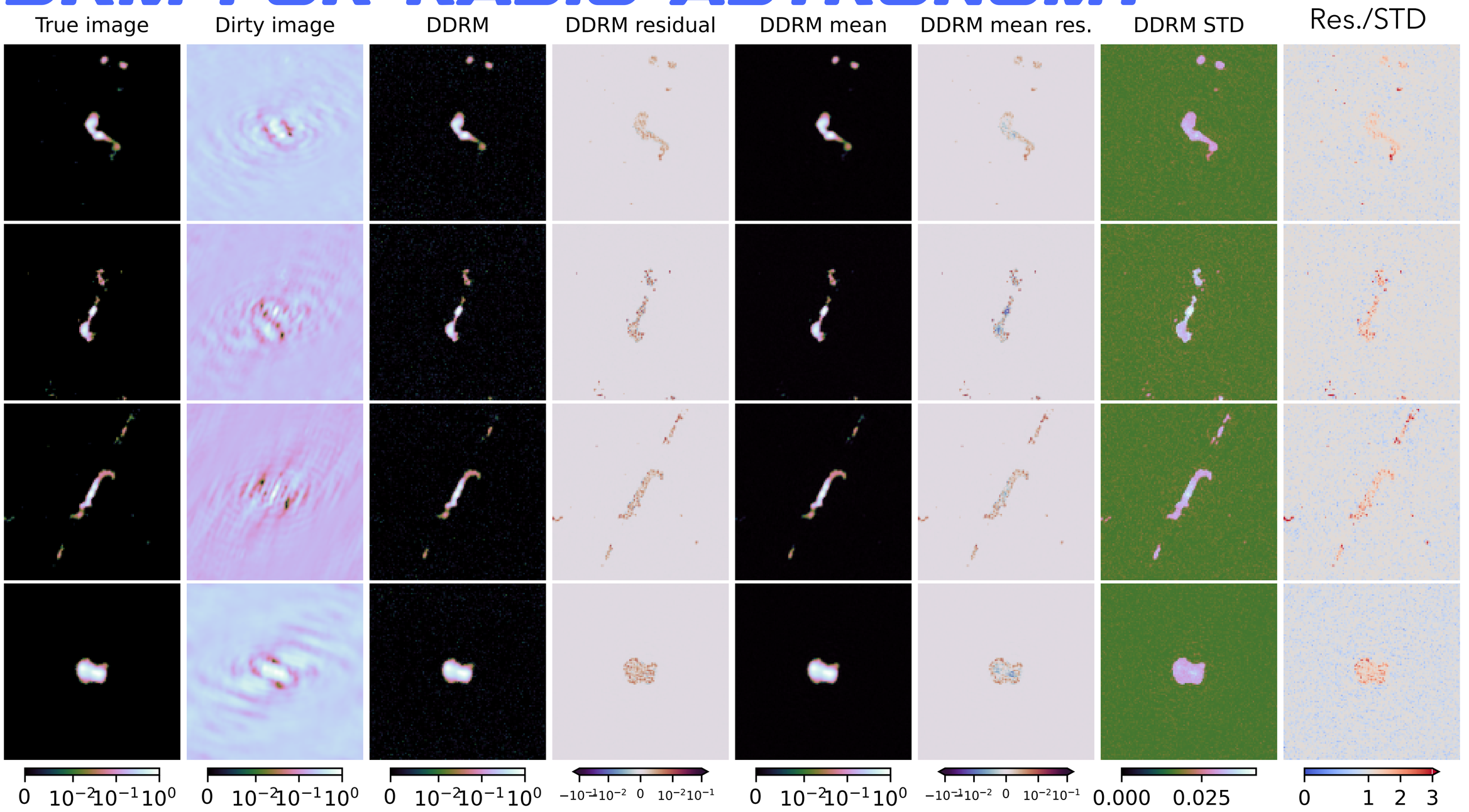
$\mathcal{F}^{-1}\mathbf{S}$



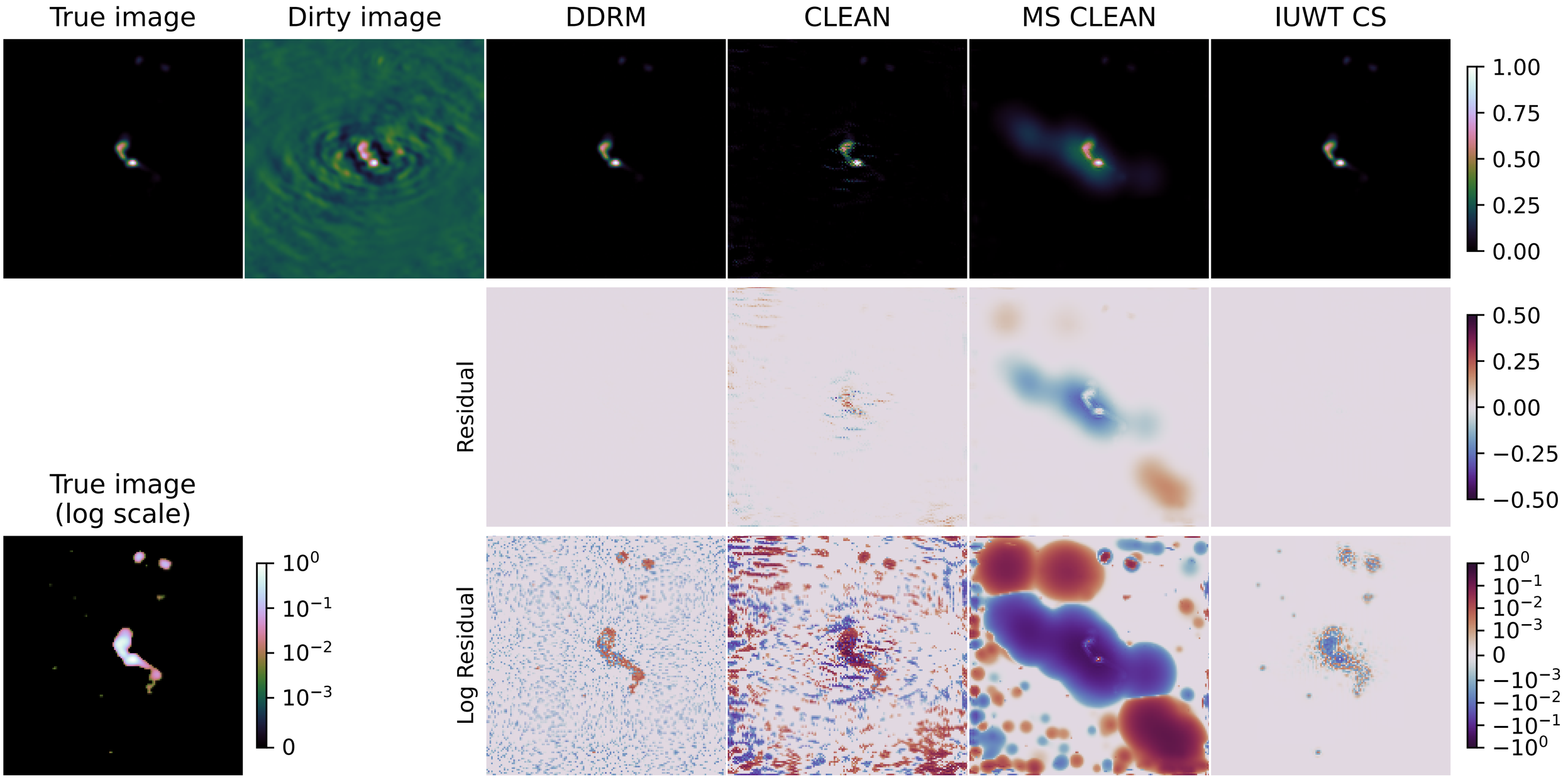
$\mathcal{F}^{-1}\mathbf{S} * \mathbf{x}$

https://arxiv.org/pdf/2601.15844

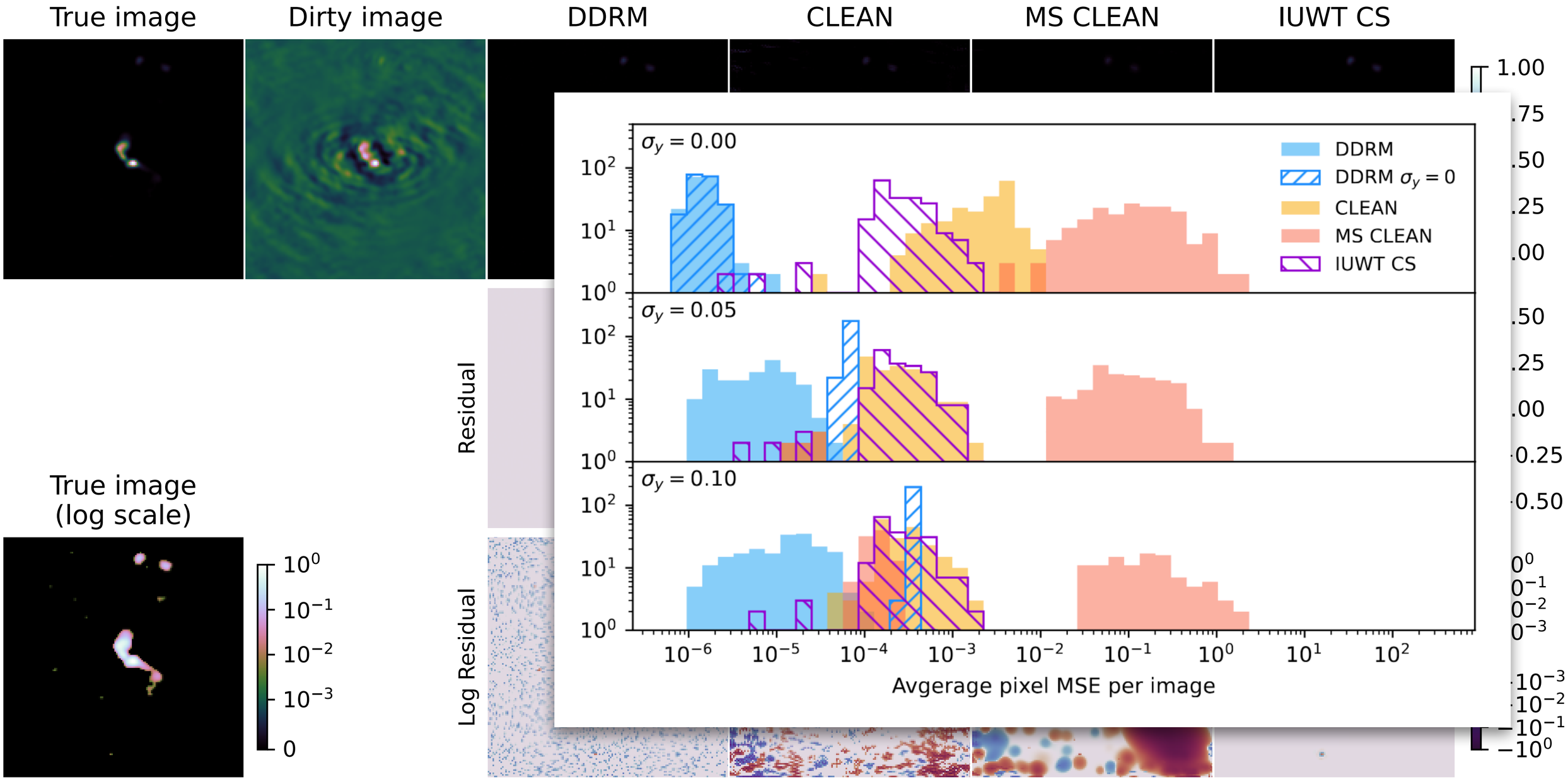
DDRM FOR RADIO ASTRONOMY



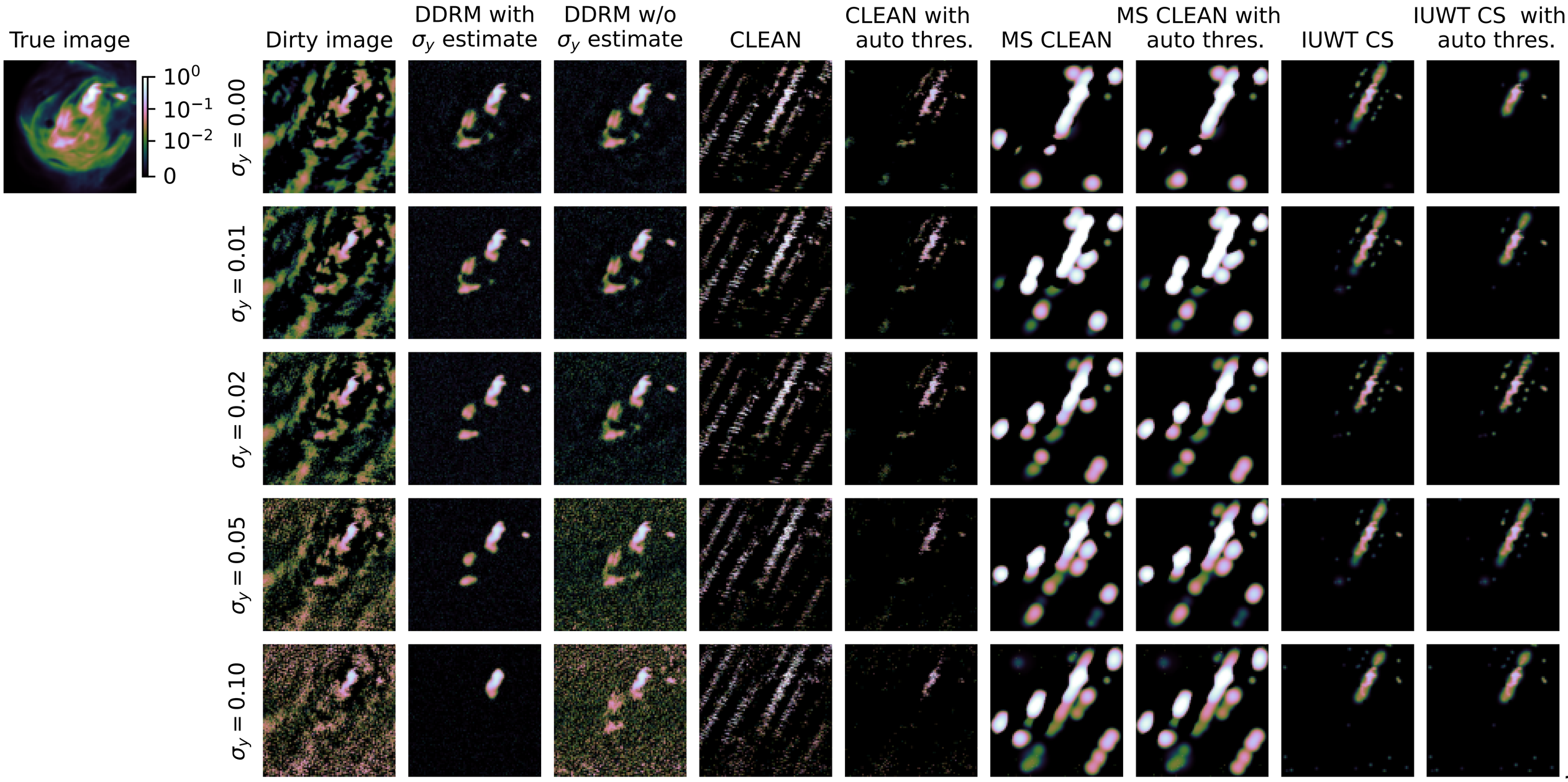
DDRM FOR RADIO ASTRONOMY



DDRM FOR RADIO ASTRONOMY



DDRM FOR RADIO ASTRONOMY



DDRM FOR RADIO ASTRONOMY

Excellent reconstruction performance, outperforming CLEAN by orders of magnitude.

Limitations:

- **fixed image size** (150x150) set by available training data.
- Does **not** provide a useful **uncertainty estimate** (SRE metric > 1), but could possibly fine-tune DDPM to calibrate
- Limited to **linear** forward operators (no calibration)

K	MSE	PSNR	SNR	SRE	t_{sampling} (s)
VLA array configuration					
10	3.2×10^{-5}	45.0	36.8	1.39	0.44
50	1.0×10^{-5}	49.9	41.7	1.548	2.21
100	6.6×10^{-6}	51.8	43.7	1.40	4.41
500	1.0×10^{-6}	59..8	51.7	1.44	22.03
1000	5.2×10^{-7}	62.9	54.7	1.04	45.47
EHT array configuration					
1000	6.8×10^{-7}	61.7	53.5	1.11	-
ALMA array configuration					
1000	5.25×10^{-7}	62.8	54.6	1.14	-

Backup Slides

Inverse problems in radio astronomy

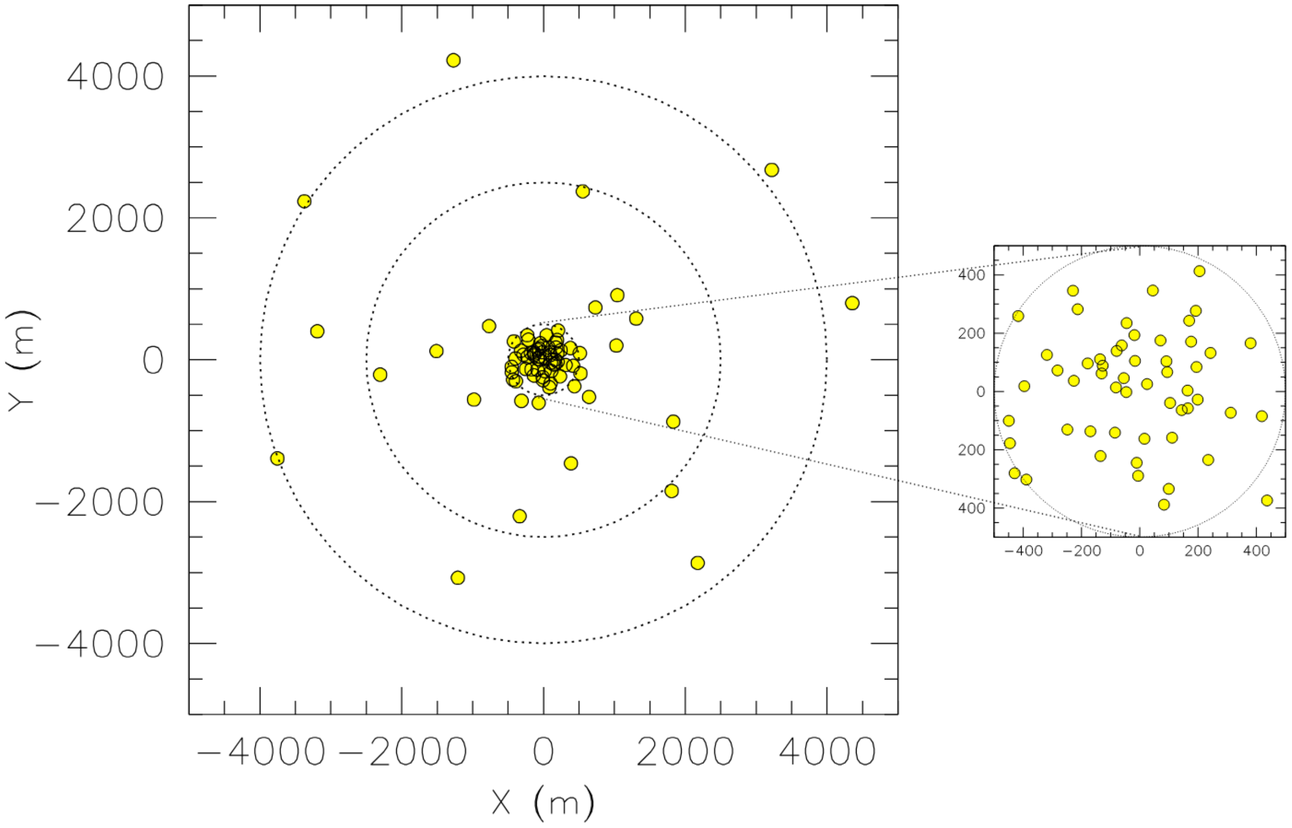
Each telescope i at position $\vec{r}_i$ measures a voltage $E(\vec{r}_p, t)$

We correlate the voltages between different pairs of antennas p and q to construct visibilities:

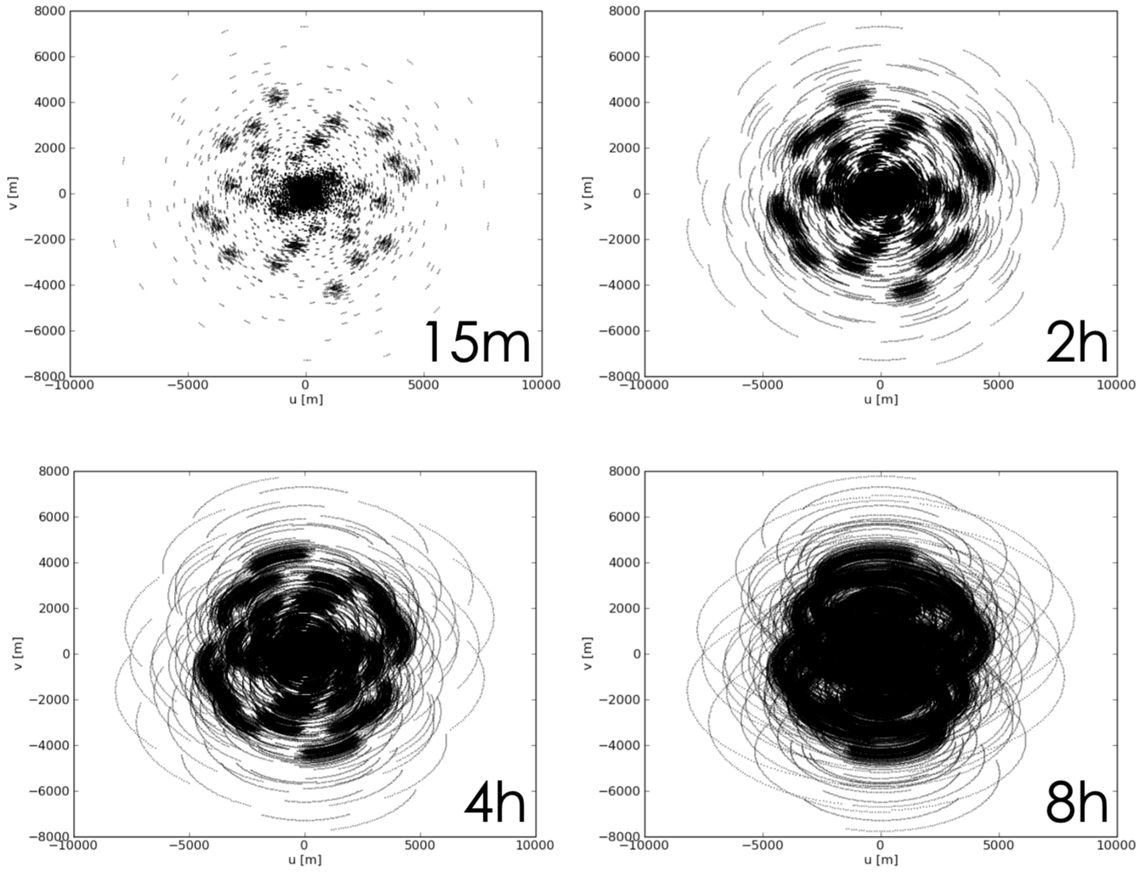
$$V_{pq} = \langle E(\vec{r}_p, t) E(\vec{r}_q, t)^* \rangle_t$$

Inverse problems in radio astronomy

Can also express visibilities as $V(u, v, w)$, where u, v, w are components of the vector $\vec{b} = \vec{r}_p - \vec{r}_q$



MeerKAT telescope layout



MeerKAT telescope uv coverage

https://arxiv.org/pdf/0910.2935

Inverse problems in radio astronomy

Visibilities are related to the radio sky image $I(l, m)$
through a Fourier transform:

$$V(u, v, w) = \iint I(l, m) e^{-2\pi i [ul+vm+wn]} \frac{dl\, dm}{n+1} + z(u, v, w)$$

Radio Interferometry

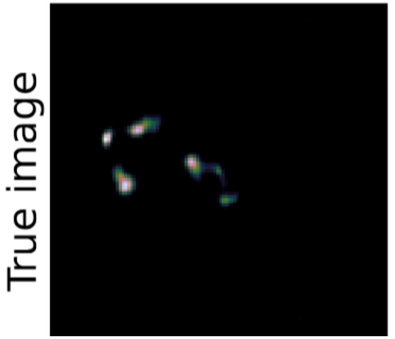
Visibilities are related to the radio sky image $I(l,m)$ through a Fourier transform:

$$V(u,v,w) = \iint I(l,m) e^{-2\pi i[ul+vm+wn]} \frac{dl\,dm}{n+1} + z(u,v,w)$$

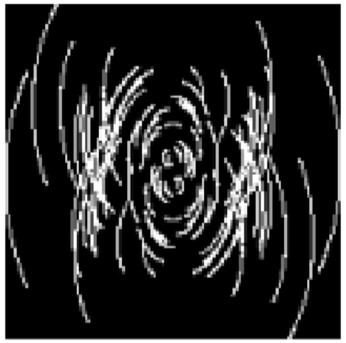
Can write this as a linear inverse problem:

$$\mathbf{y} = \mathbf{S}\mathcal{F}\mathbf{x} + \mathbf{z}$$

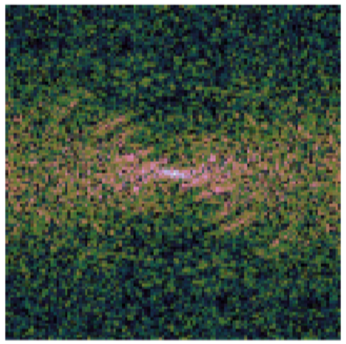
Because $\mathbf{S}$ removes data, this is an **ill-posed inverse problem**



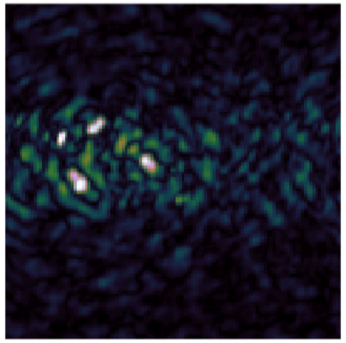
$\mathbf{x}$



$\mathbf{S}$

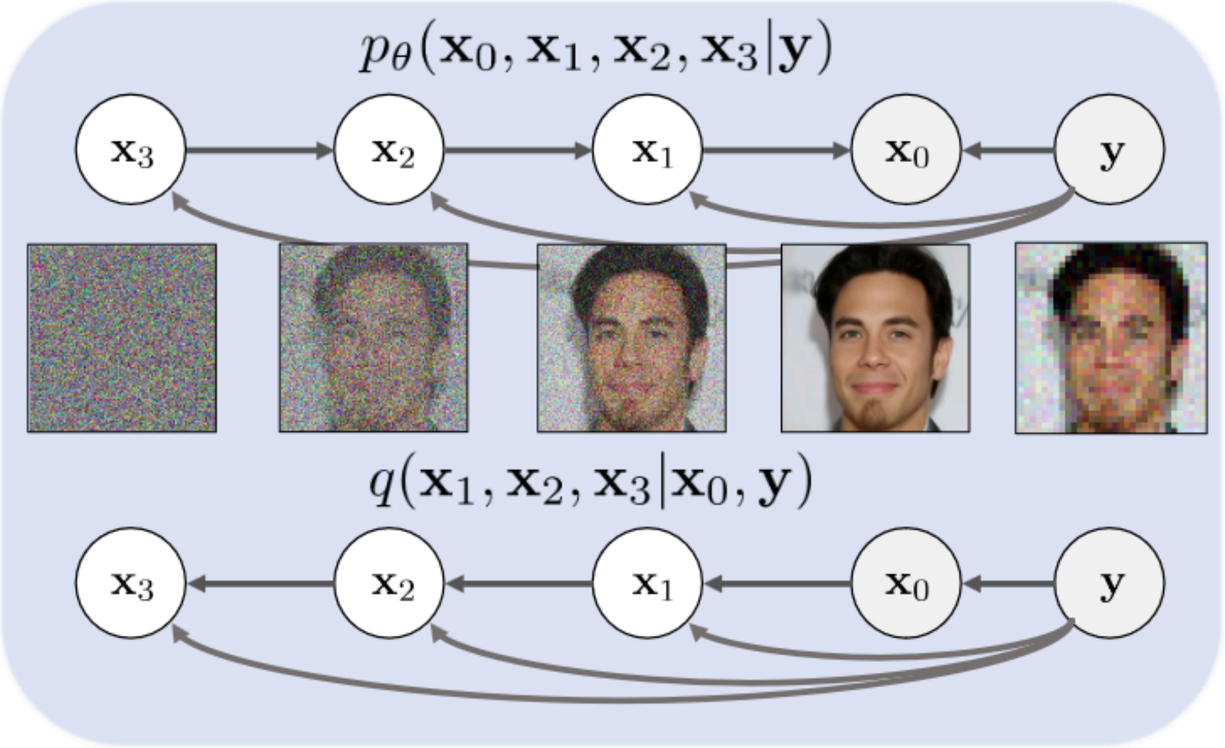


$\mathcal{F}^{-1}\mathbf{S}$



$\mathcal{F}^{-1}\mathbf{S} * \mathbf{x}$

Denoising Diffusion Restoration Models (DDRM)



Denoising Diffusion Restoration Models
(Dependent on inverse problem)

DDRM is a Markov chain conditioned on $\mathbf{y}$ which corresponds to the evidence lower bound:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \mathbf{y} \sim q(\mathbf{y}|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{y})] \\
\geq\ &-\mathbb{E}\left[\sum_{t=1}^{T-1} D_{\mathrm{KL}}(q^{(t)}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) \| p_\theta^{(t)}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}))\right] \\
&+\mathbb{E}\left[\log p_\theta^{(0)}(\mathbf{x}_0|\mathbf{x}_1, \mathbf{y})\right] \\
&-\mathbb{E}[D_{\mathrm{KL}}(q^{(T)}(\mathbf{x}_T|\mathbf{x}_0, \mathbf{y}) \| p_\theta^{(T)}(\mathbf{x}_T|\mathbf{y}))]
\end{aligned}$$

Uses a pre-trained DDPM as a prior on $\mathbf{x}$

$\mathbf{x}$ and $\mathbf{y}$ are related by a linear forward model:

$$\mathbf{y} = \boldsymbol{H}\mathbf{x} + \mathbf{z}$$

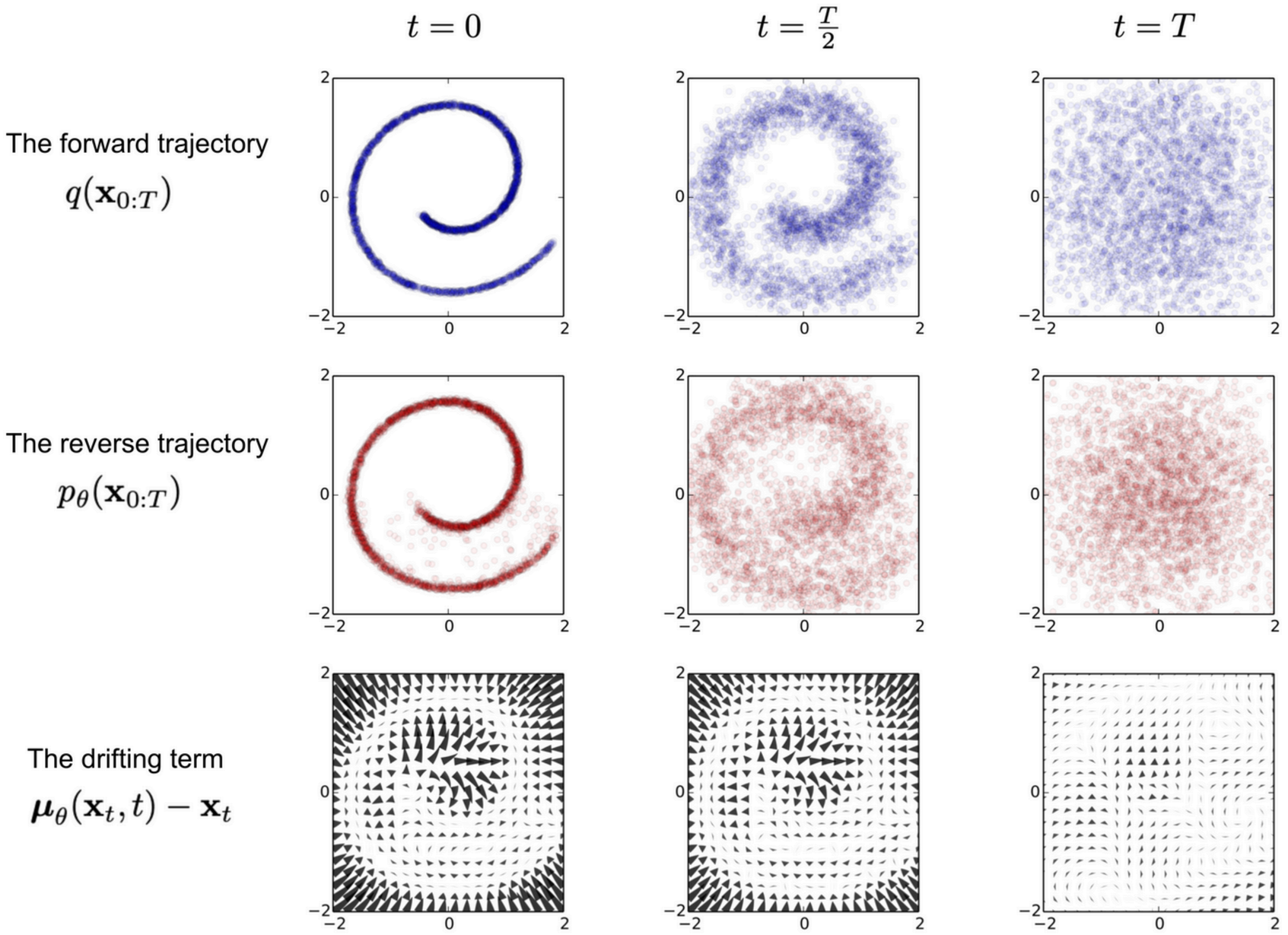


Fig. 3. An example of training a diffusion model for modeling a 2D swiss roll data. (Image source: Sohl-Dickstein et al., 2015)

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

The loss term L_t is parameterized to minimize the difference from $\tilde{\boldsymbol{\mu}}$:

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\|_2^2} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2} \| \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_t \right) - \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2} \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2} \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2 \right]
\end{aligned}$$

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/