Sampling high-dimensional posterior with a simulation based prior

astro-ph.CO arXiv:2201.05561 stat.ML arXiv:2011.08698

Benjamin Remy

With : Francois Lanusse, Zaccharie Ramzi, Niall Jeffrey, Jia Liu, J.-L. Starck

CosmoStat

cnes

université PARIS-SACLAY

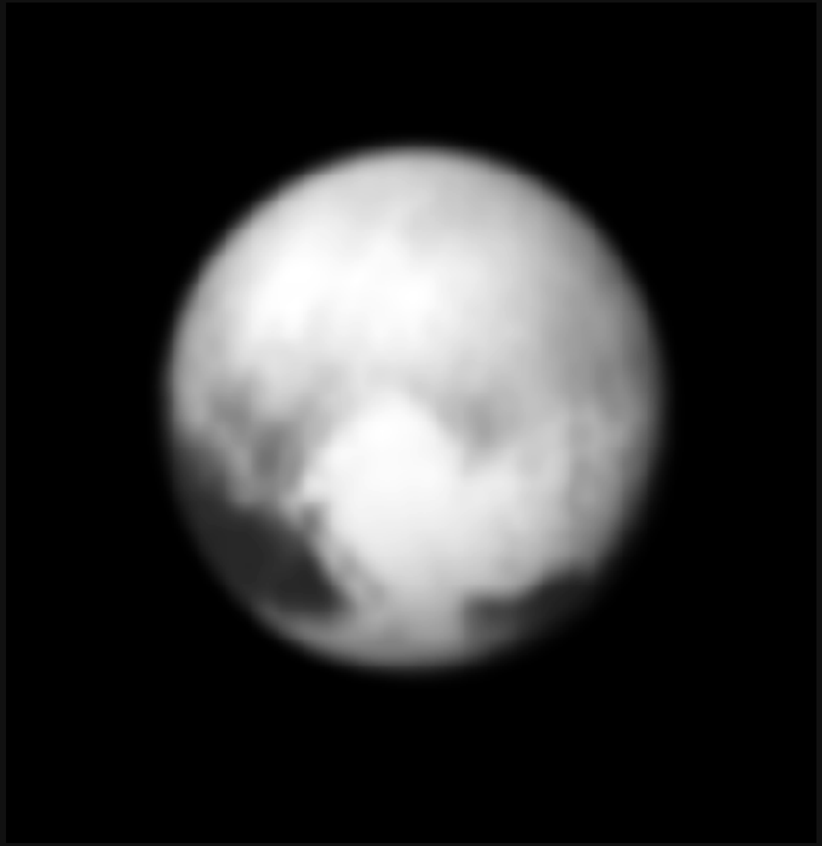
slides at b-remy.github.io/talks/Paris2022

Linear inverse problems

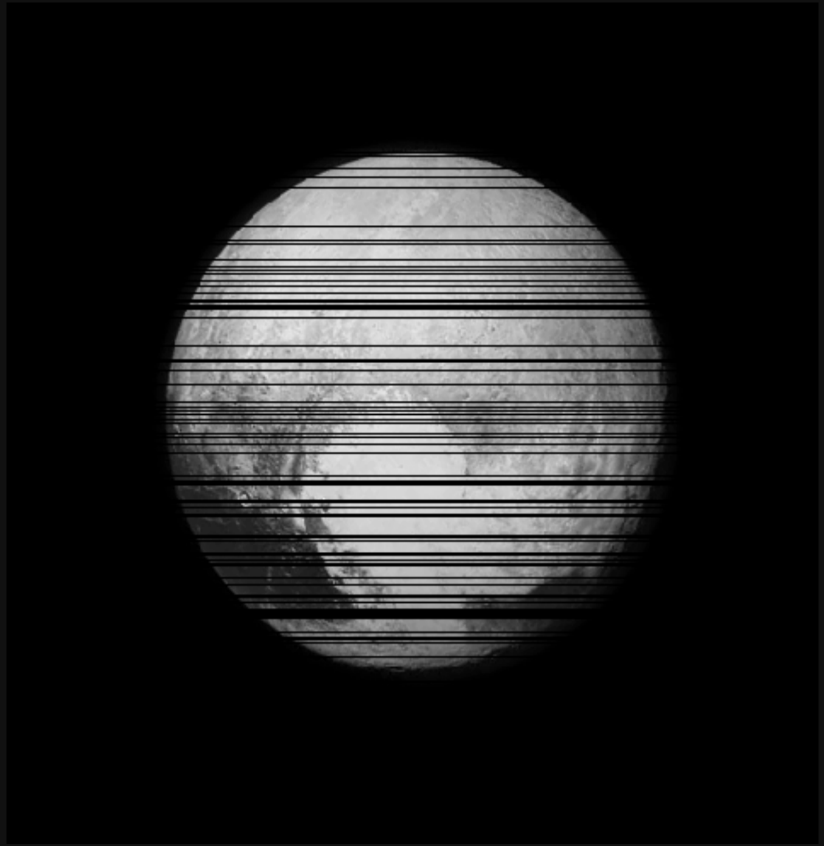
$$y = \mathbf{A}x + n$$

$\mathbf{A}$ is known and encodes our physical understanding of the problem.

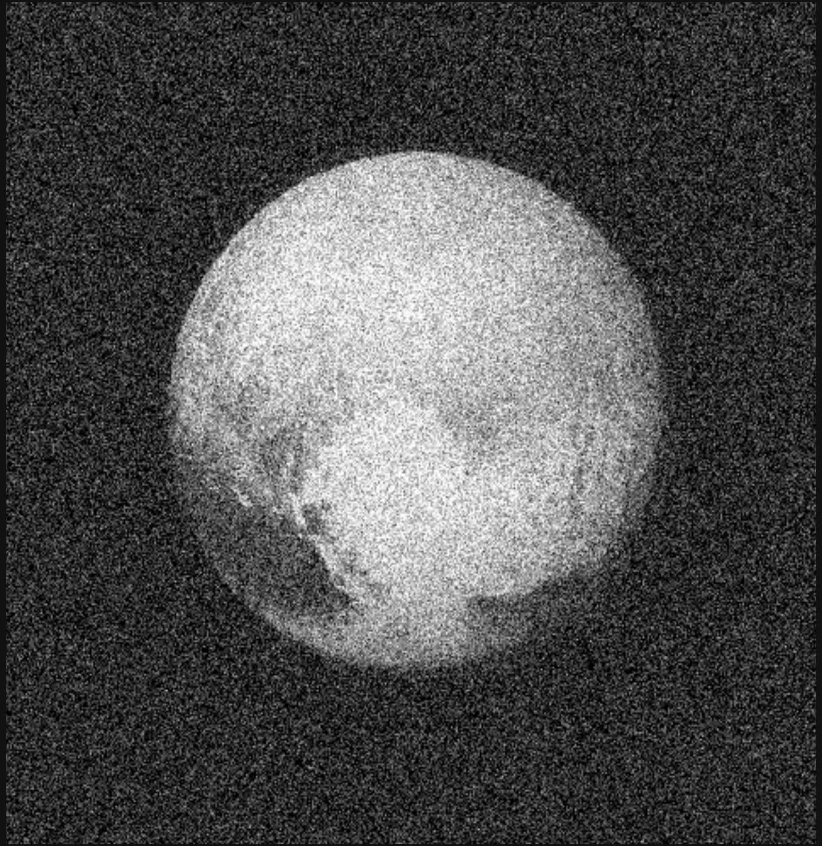
$\implies$ When non-invertible or ill-conditioned, the inverse problem is ill-posed **with no unique solution** x



Deconvolution



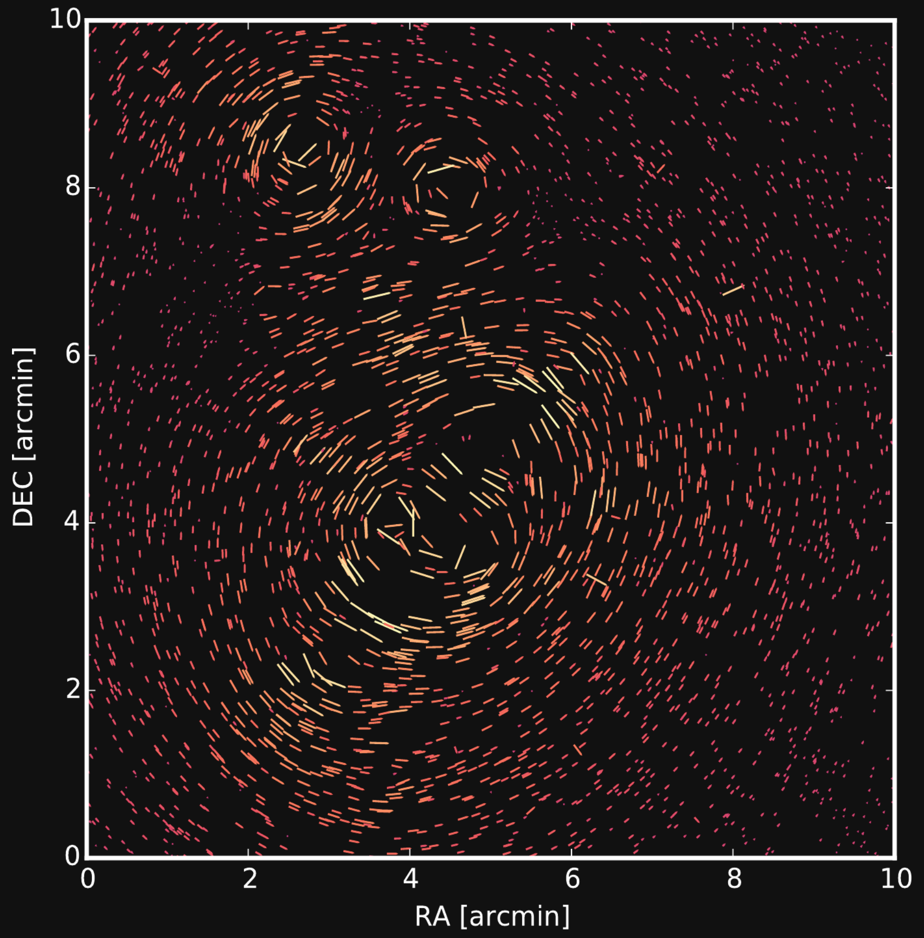
Inpainting



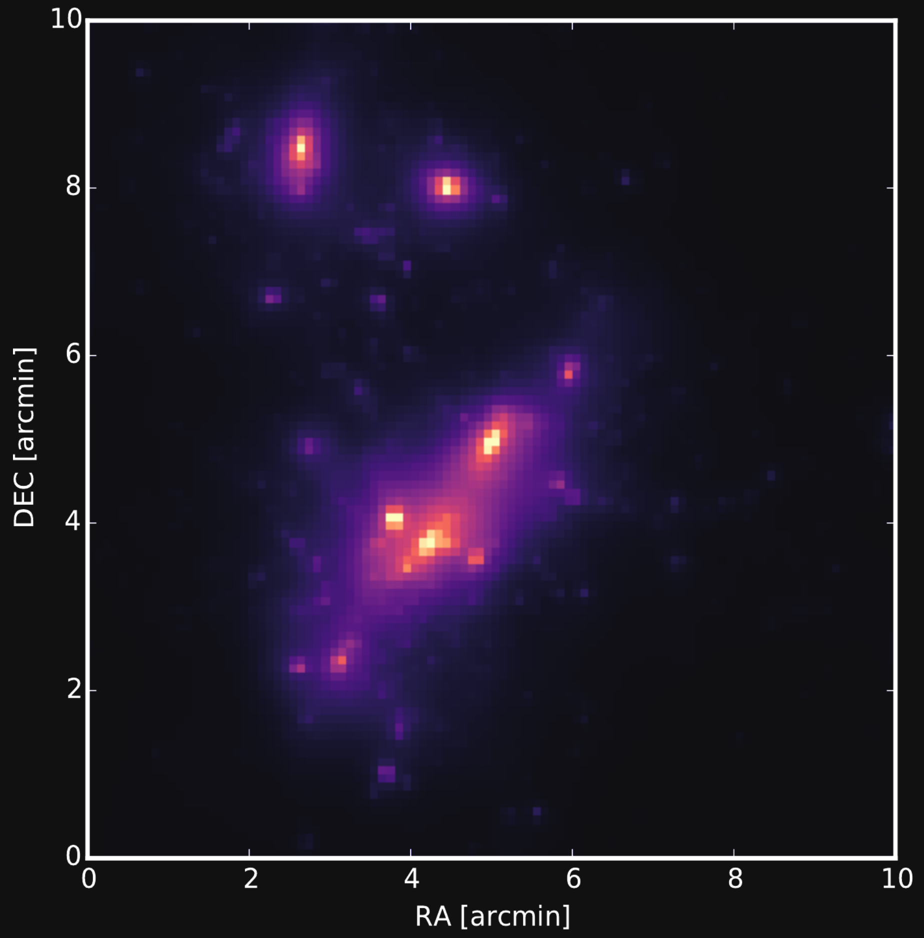
Denoising

The Weak Lensing Mass-Mapping as an Inverse Problem

Shear γ



Convergence κ

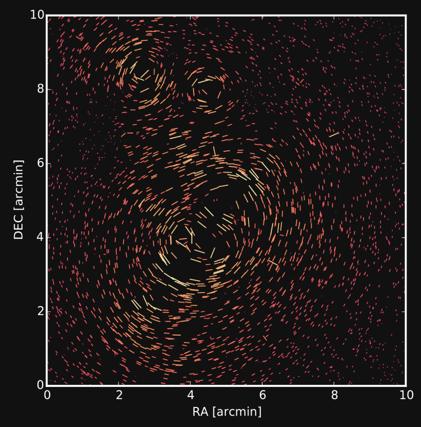


$$\gamma = \mathbf{P}\kappa + n$$

Bayesian Modeling

$$\gamma = \mathbf{P}\kappa + n$$

$\mathbf{P}$ is known and encodes our physical understanding of the problem
$\implies$ Non-invertible (*survey mask*, *shape noise*), the inverse problem is ill-posed **with no unique solution κ**

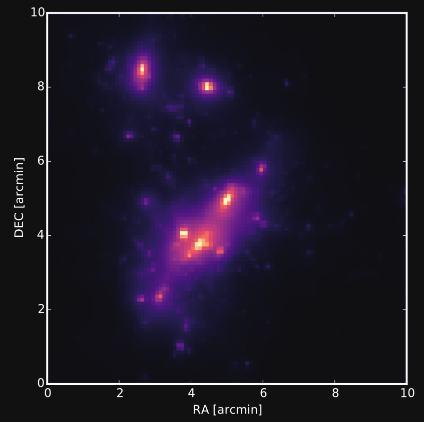


γ

The Bayesian view of the problem:

$$p(\kappa \mid \gamma) \propto p(\gamma \mid \kappa)\, p(\kappa)$$

- $p(\gamma \mid \kappa)$ is the data likelihood, which **contains the physics**
- $p(\kappa)$ is the prior knowledge on the solution.



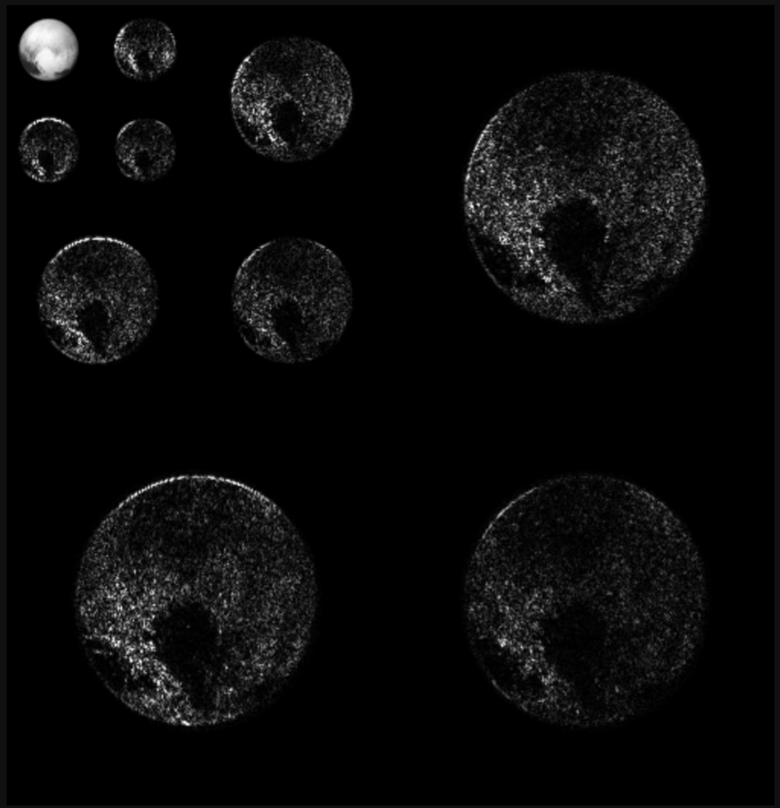
κ

In this perspective we can provide point estimates: **Posterior Mean**, **Max**, **Median**, etc.
and **the full posterior $p(\kappa \mid \gamma)$** with **Markov Chain Monte Carlo** or Variational Inference methods

How do you choose the prior ?

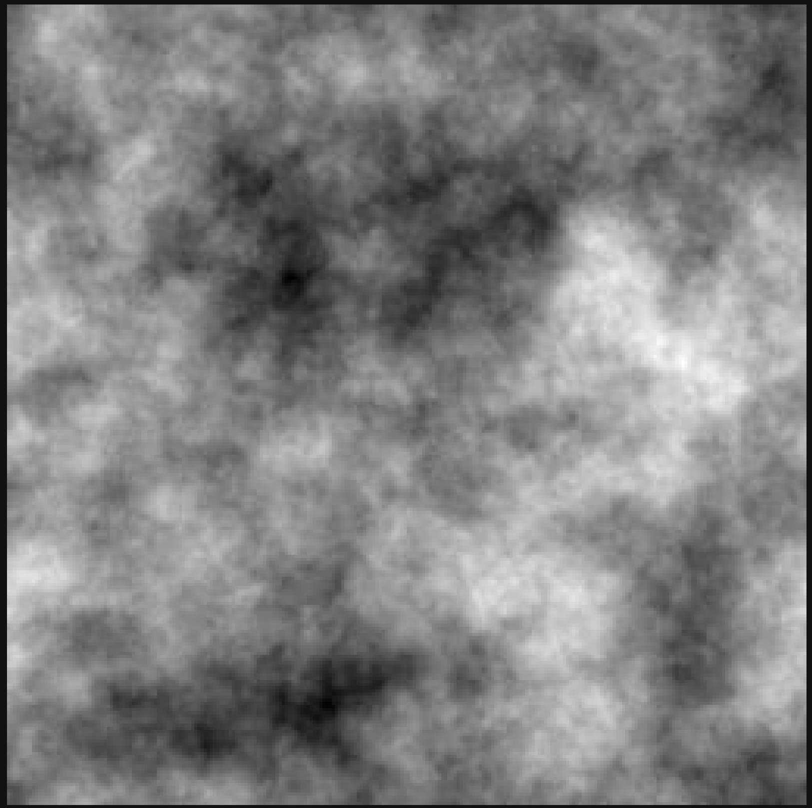
Classical examples of signal priors

Sparse



$$\log p(x) = \| \mathbf{W}x \|_1$$

Gaussian



$$\log p(x) = x^t \mathbf{\Sigma}^{-1} x$$

Total Variation

$$\log p(x) = \| \nabla x \|_1$$

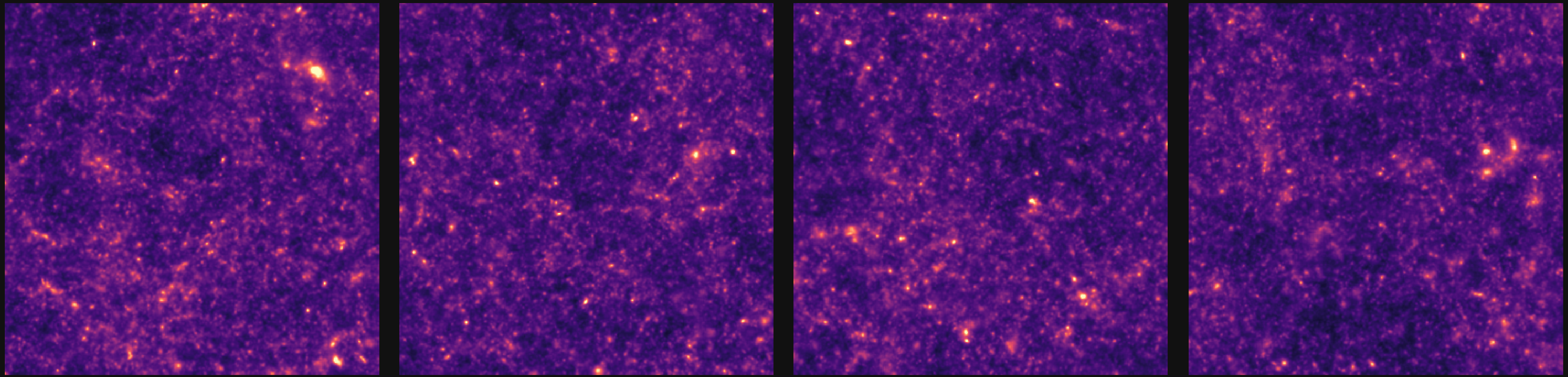
But what about learning the prior
with **deep generative models**?

Writing down the convergence map log posterior

$$\log p(\kappa \mid e) = \underbrace{\log p(e \mid \kappa)}_{\simeq -\frac{1}{2} \parallel e - P\kappa \parallel_\Sigma^2} + \log p(\kappa) + cst$$

- The likelihood term is **known analytically.**
- There is **no close form expression for the full non-Gaussian prior** of the convergence.
  However:
  - **We do have access to samples of full implicit prior** through simulations: $X = \{x_0, x_1, \ldots, x_n\}$ with $x_i \sim \mathrm{P}$

κTNG (Osato et al. 2021)



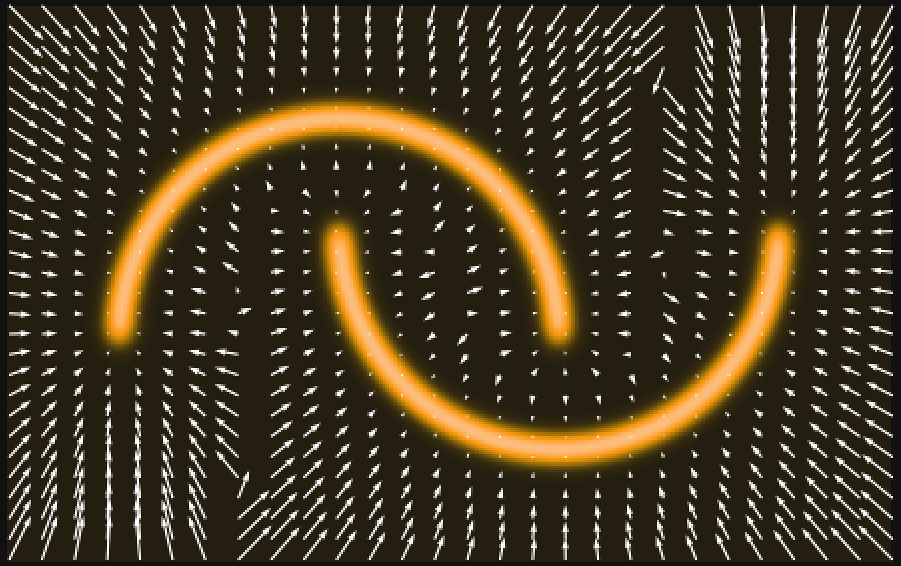
$\Longrightarrow$ Our strategy: **Learn the prior from simulation**, and then **sample the full posterior**.

The score is all you need!

- Whether you are looking for the MAP or sampling with HMC or MALA, you **only need access to the score** of the posterior:

$$\frac{\partial \log p(x|y)}{\partial x}$$

  - Gradient descent: $x_{t+1} = x_t + \tau \nabla_x \log p(x_t|y)$
  - Langevin algorithm: $x_{t+1} = x_t + \tau \nabla_x \log p(x_t|y) + \sqrt{2\tau} n_t$



- The score of the full posterior is simply:

$$\nabla_x \log p(x|y) = \nabla_x \underbrace{\log p(y|x)}_{\text{known}} + \nabla_x \underbrace{\log p(x)}_{\text{can be learned}}$$

$\Longrightarrow$ all we have to do is **model/learn the score of the prior**.

Neural Score Estimation by Denoising Score Matching

- **Denoising Score Matching:** An optimal **Gaussian denoiser learns the score** of a given distribution.
  - If $x \sim \mathrm{P}$ is corrupted by additional Gaussian noise $u \in \mathrm{N}(0, \sigma^2)$ to yield

$$x' = x + u$$

  - Let's consider a denoiser r_θ trained under an ℓ_2 loss:

$$\mathrm{L} = \| x - r_\theta(x', \sigma) \|_2^2$$

  - The optimal denoiser $r_{\theta^\star}$ verifies:

$$\boldsymbol{r}_{\theta^\star}(\boldsymbol{x}', \sigma) = \boldsymbol{x}' + \sigma^2 \nabla_{\boldsymbol{x}} \log p_{\sigma^2}(\boldsymbol{x}')$$

Sampling method: Annealed Hamiltonian Monte Carlo

Sampling convergence maps $\kappa \sim p(\kappa|\gamma)$ **is very difficult** due to the high dimensionality of the space ($360 \times 360 \approx 10^5$ parameters).

Especially for MCMC algorithms because of *curse of dimensionality* leading to *highly correlated chains*.

We need to design an efficient sampler. $\kappa_1, \kappa_2, \ldots, \kappa_N \sim p(\kappa|\gamma)$

- **Hamiltonial Monte Carlo** proposal for a step size α:

$$
\begin{aligned}
\boldsymbol{m}_{t+\frac{\alpha}{2}} &= \boldsymbol{m}_t + \frac{\alpha}{2} \nabla_\kappa \log p(\boldsymbol{\kappa}_t|\gamma) \\
\boldsymbol{\kappa}_{t+\alpha} &= \boldsymbol{\kappa}_t + \alpha \mathbf{M}^{-1} \boldsymbol{\kappa}_{t+\frac{\alpha}{2}} \\
\boldsymbol{m}_{t+\alpha} &= \boldsymbol{m}_{t+\frac{\alpha}{2}} + \frac{\alpha}{2} \nabla_\kappa \log p(\boldsymbol{\kappa}_{t+\alpha}|\gamma)
\end{aligned}
$$

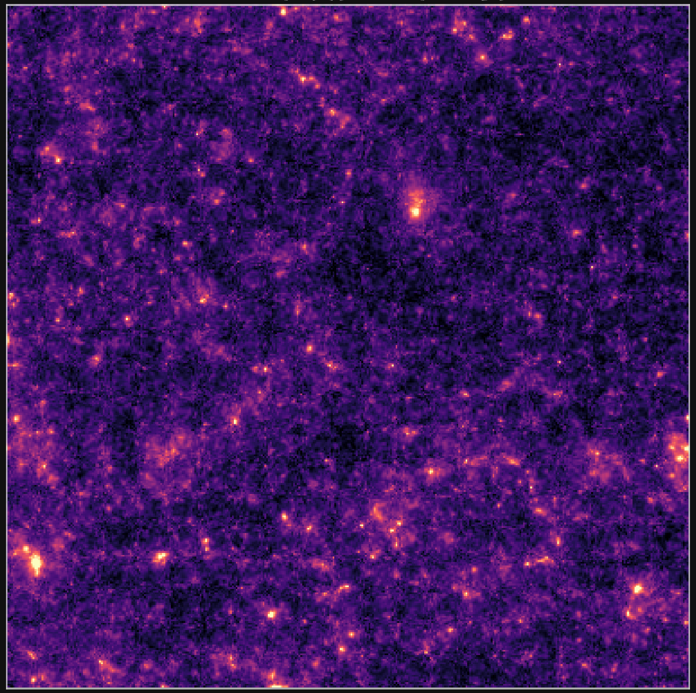
- **Annealing:** convolve the posterior with a wide gaussian to always remain on high probability density.

$$
p_\sigma(x) = \int p_{\text{data}}(x')\mathrm{N}(x|x', \sigma^2)dx', \qquad \sigma_1 > \sigma_2 > \sigma_3 > \sigma_4
$$

Sampling method: Annealed Hamiltonian Monte Carlo

Target $\sigma_0 \approx 0$

$x_t \sim p(\kappa|\gamma) * \mathcal{N}(0, \sigma_t^2)$



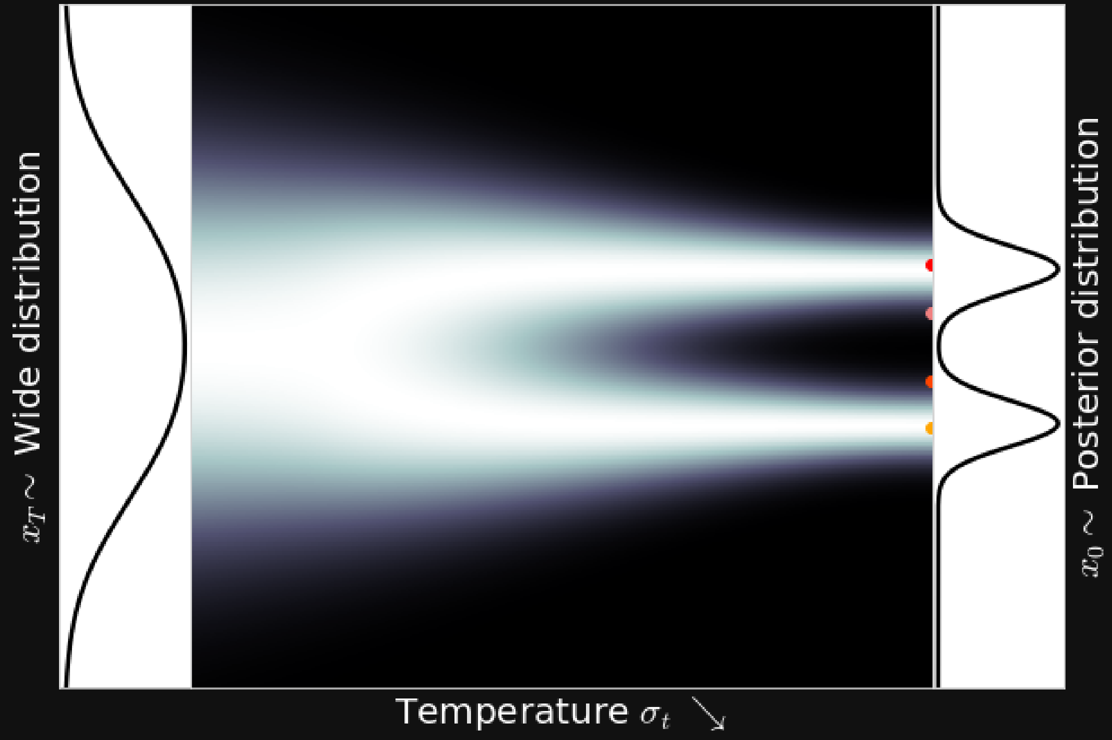


Illustration on κ-TNG simulations

$$\nabla_\kappa \log p_\sigma(\kappa|\gamma) = \nabla_\kappa \log p_\sigma(\gamma|\kappa) \quad + \quad \nabla_\kappa \log p_\sigma(\kappa)$$

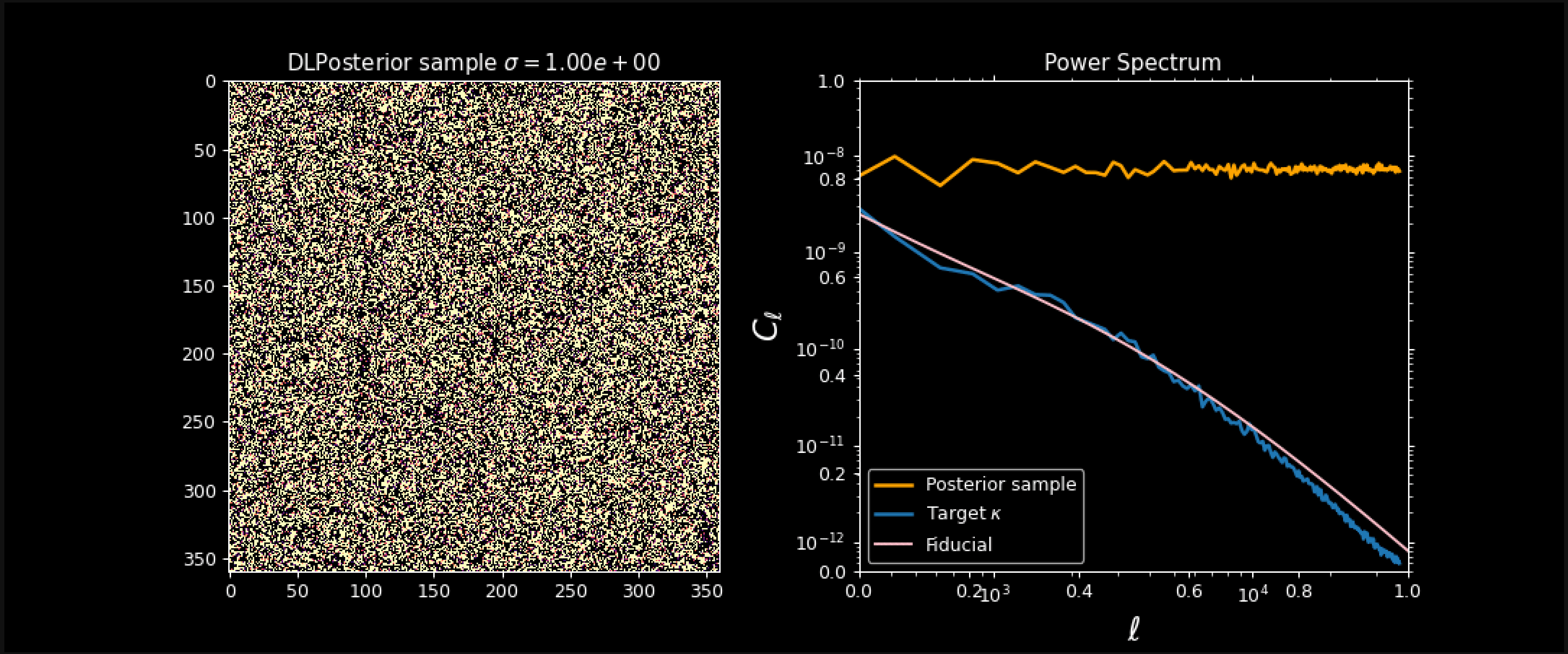
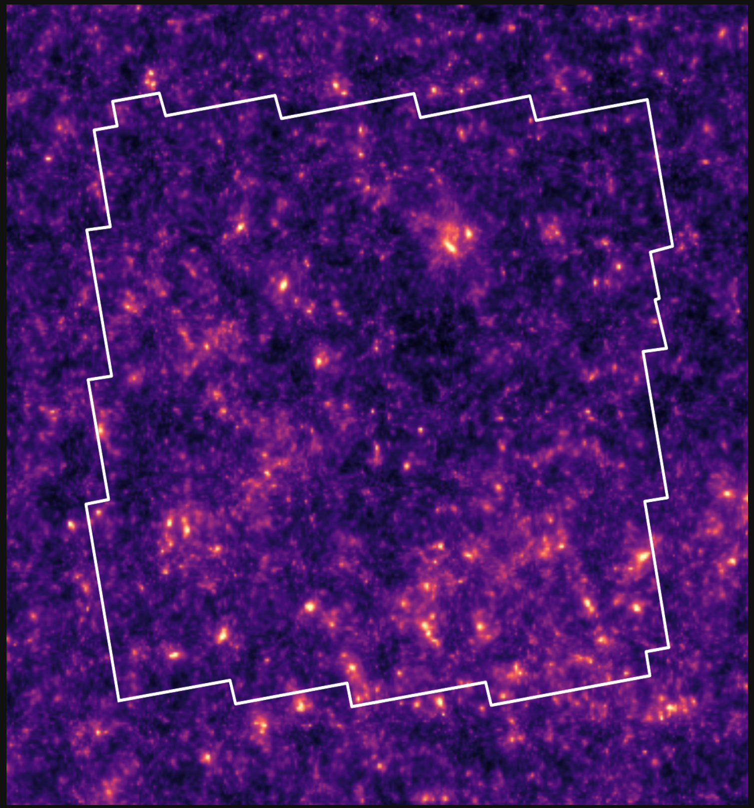
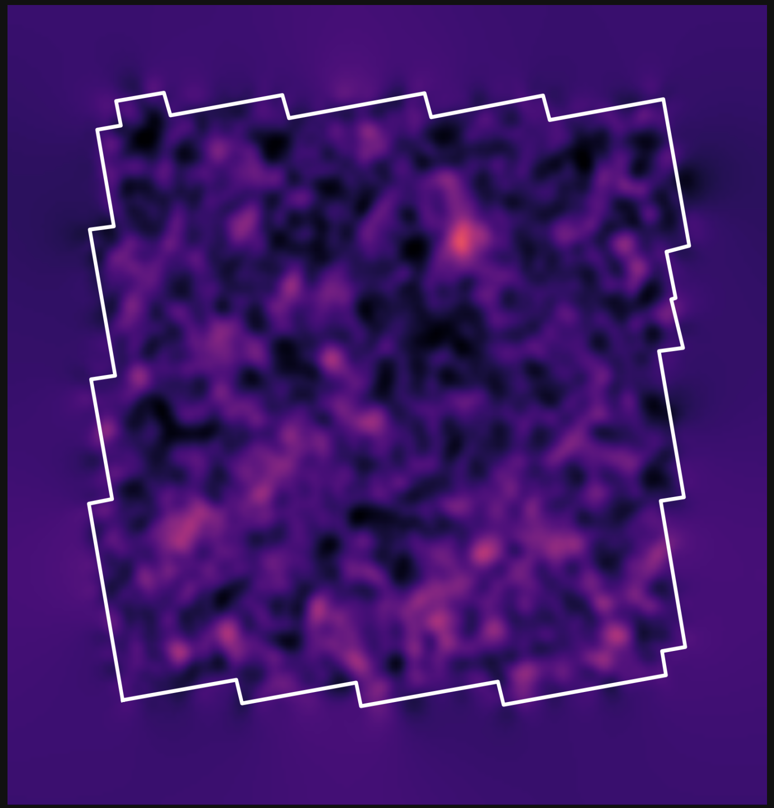


Illustration on κ-TNG simulations

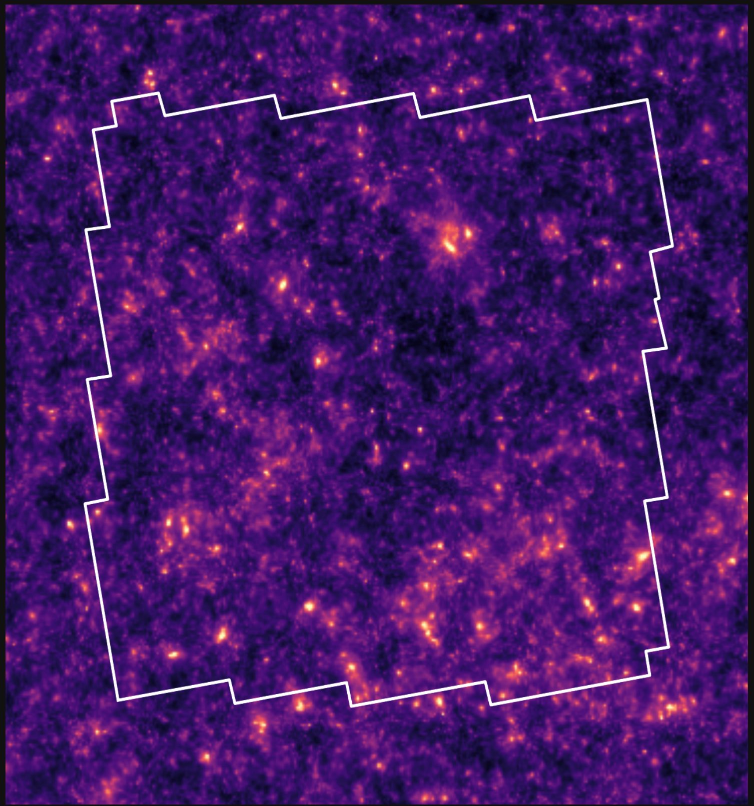


True convergence map

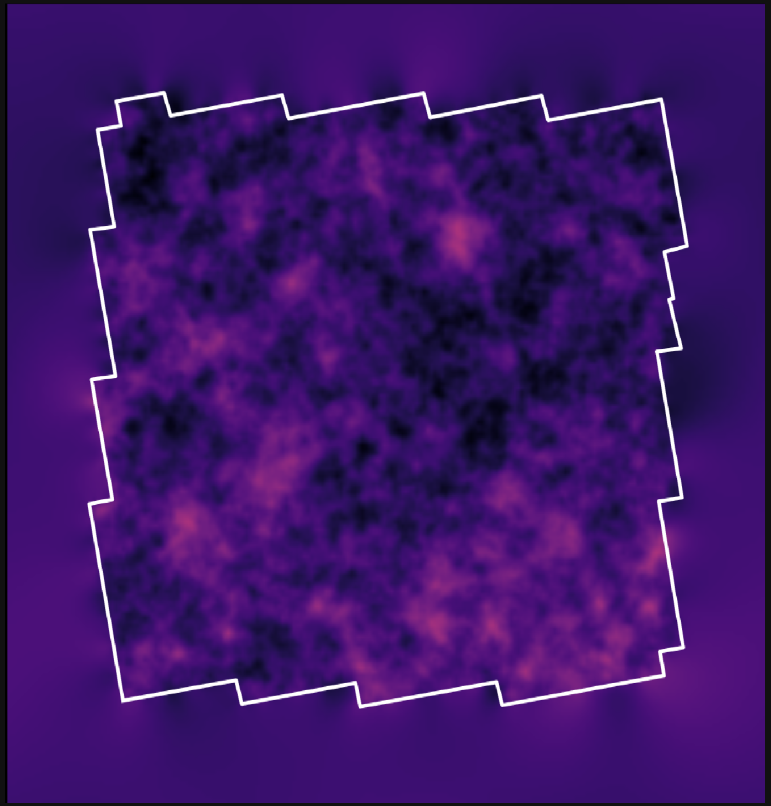


Traditional Kaiser-Squires

Illustration on κ-TNG simulations

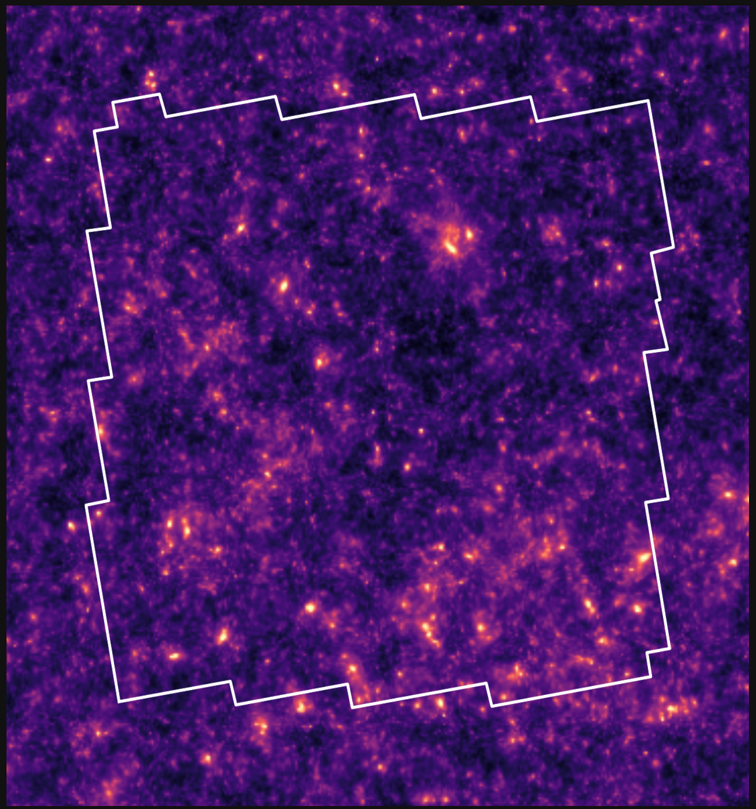


True convergence map

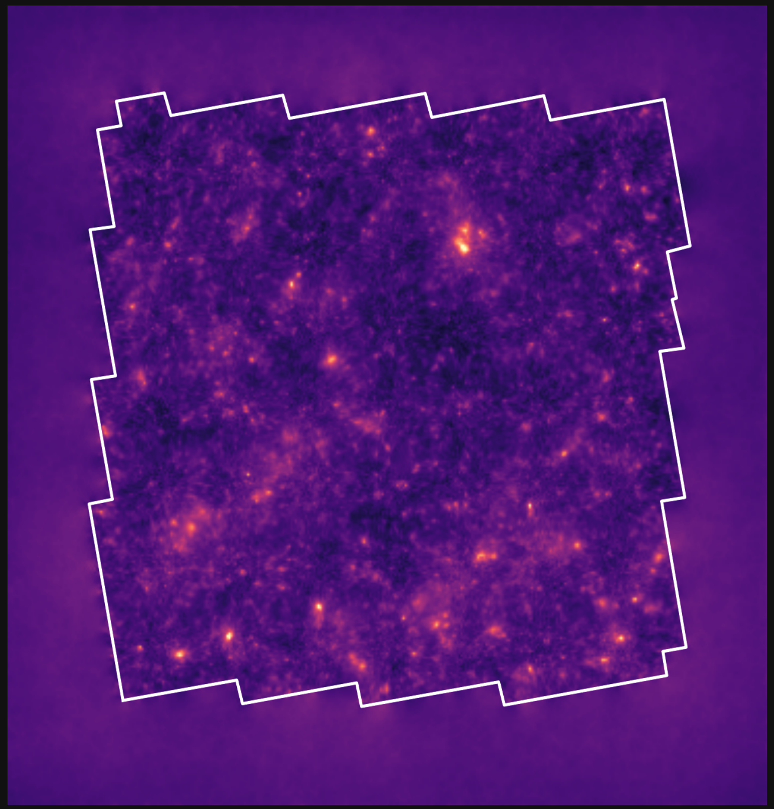


Wiener Filter

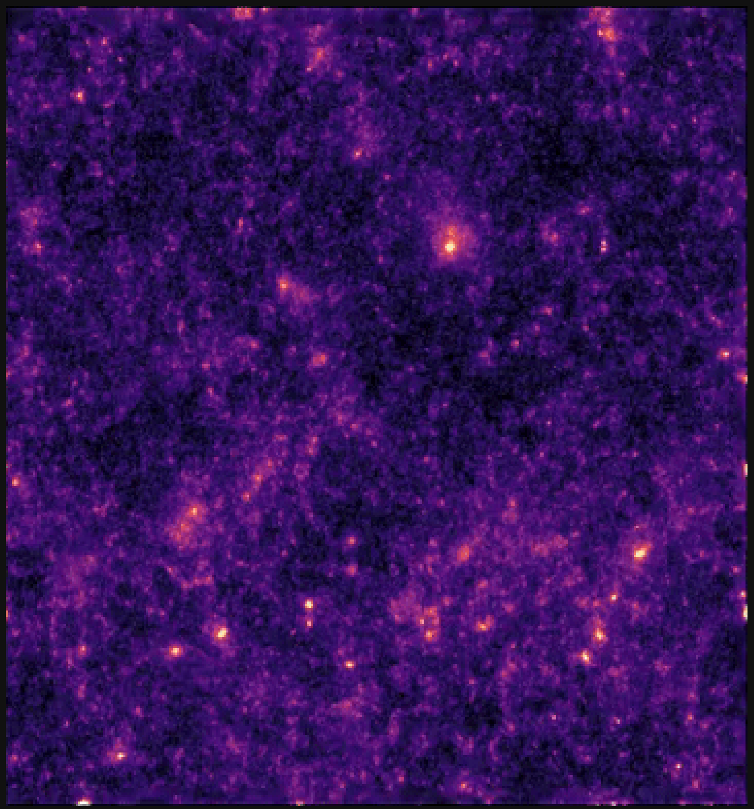
Illustration on κ-TNG simulations



True convergence map



Posterior Mean (ours)

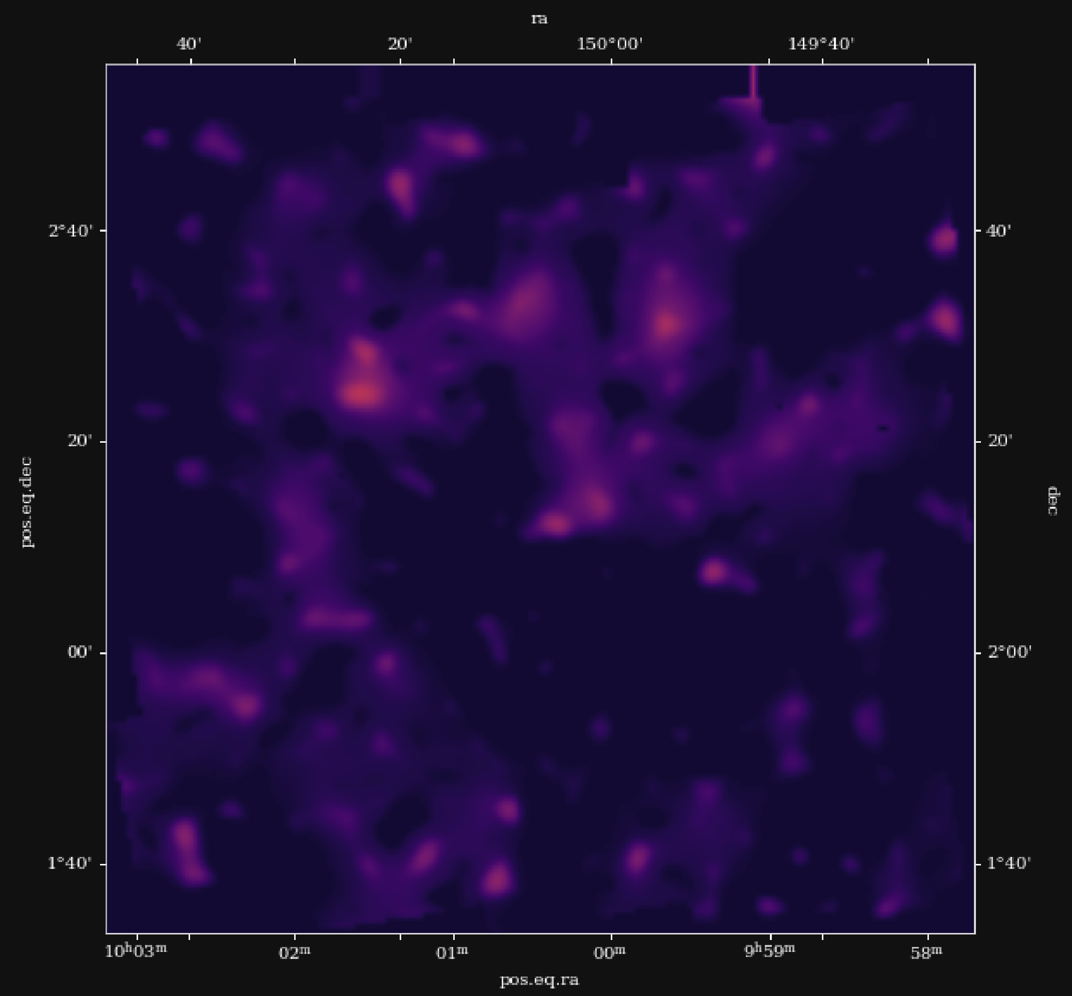


Posterior samples

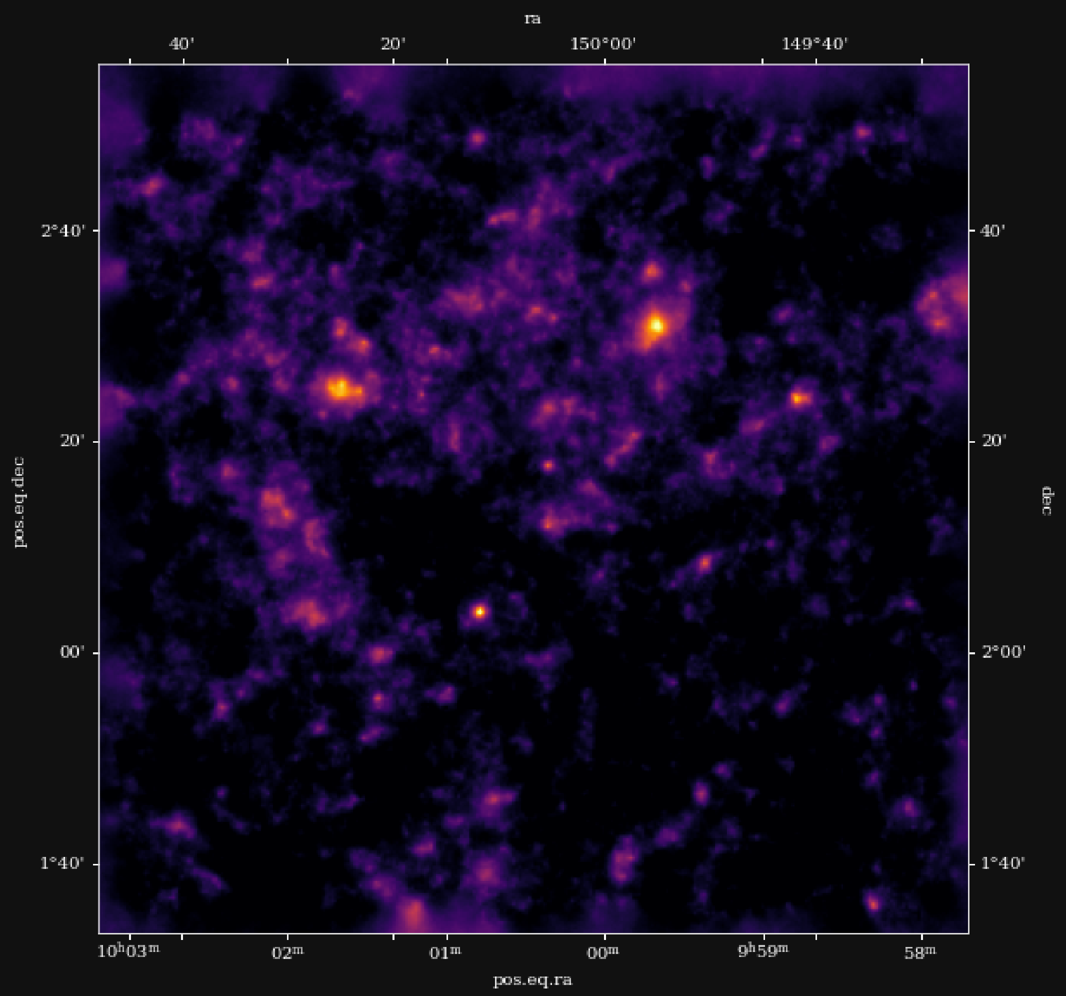
Reconstruction of the HST/ACS COSMOS field

1.637 square degree, 64.2 gal/arcmin^2

Massey et al. (2007)



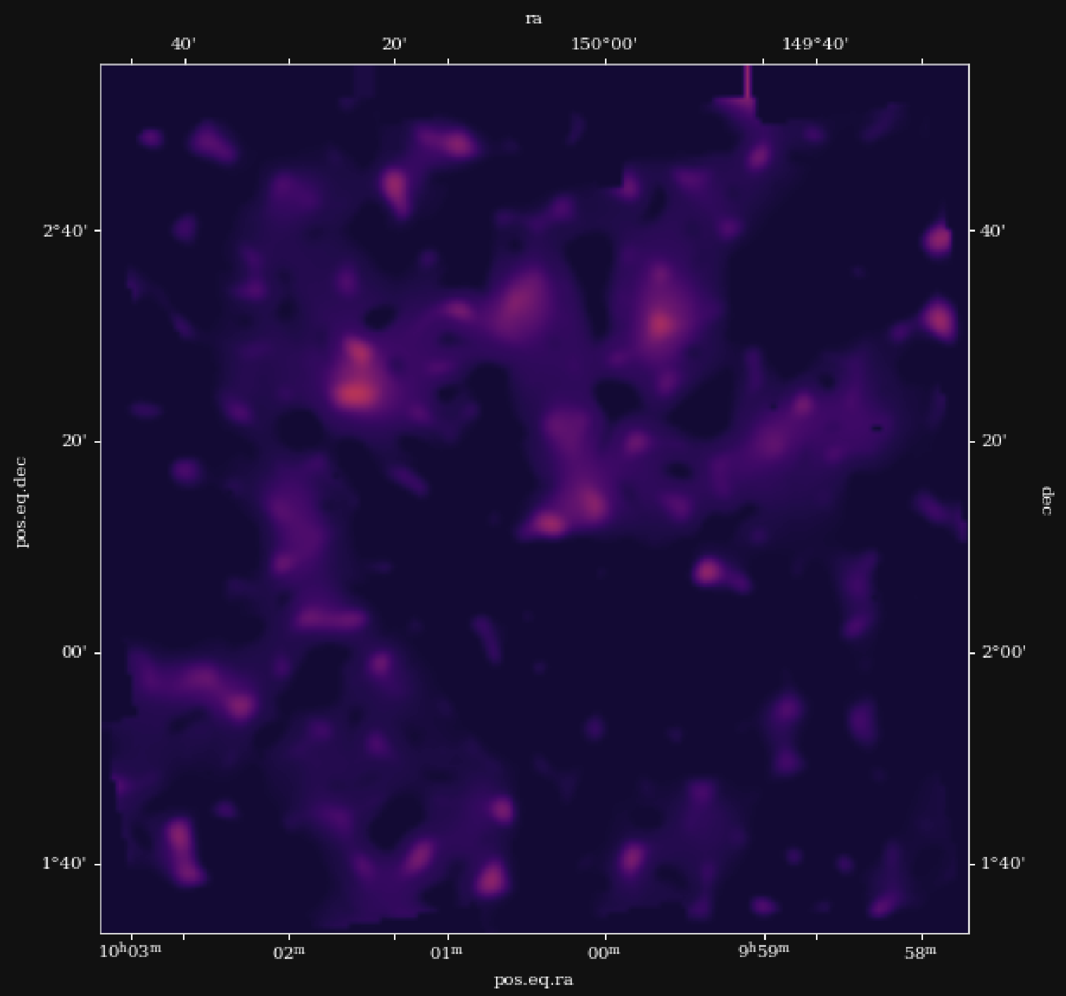
Remy et al. (2022) Posterior mean



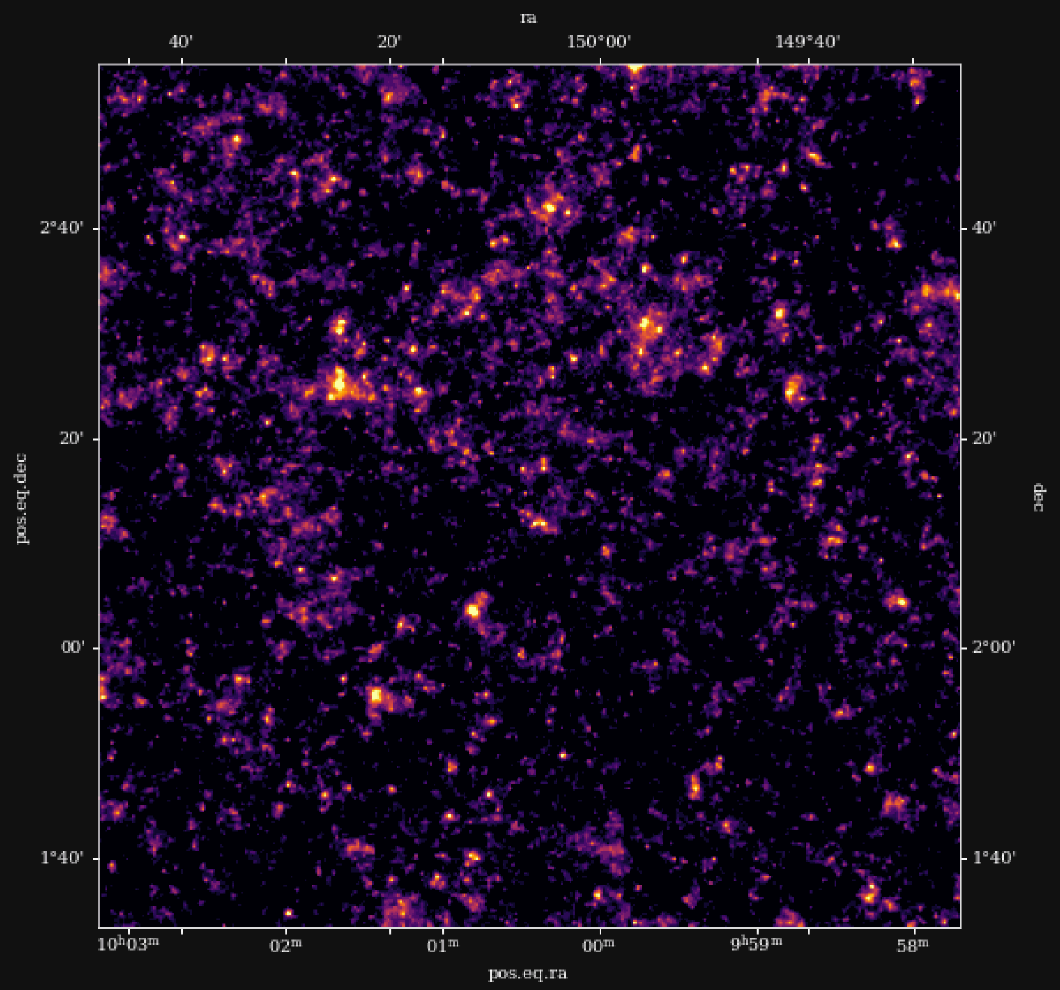
Reconstruction of the HST/ACS COSMOS field

1.637 square degree, 64.2 gal/arcmin^2

Massey et al. (2007)



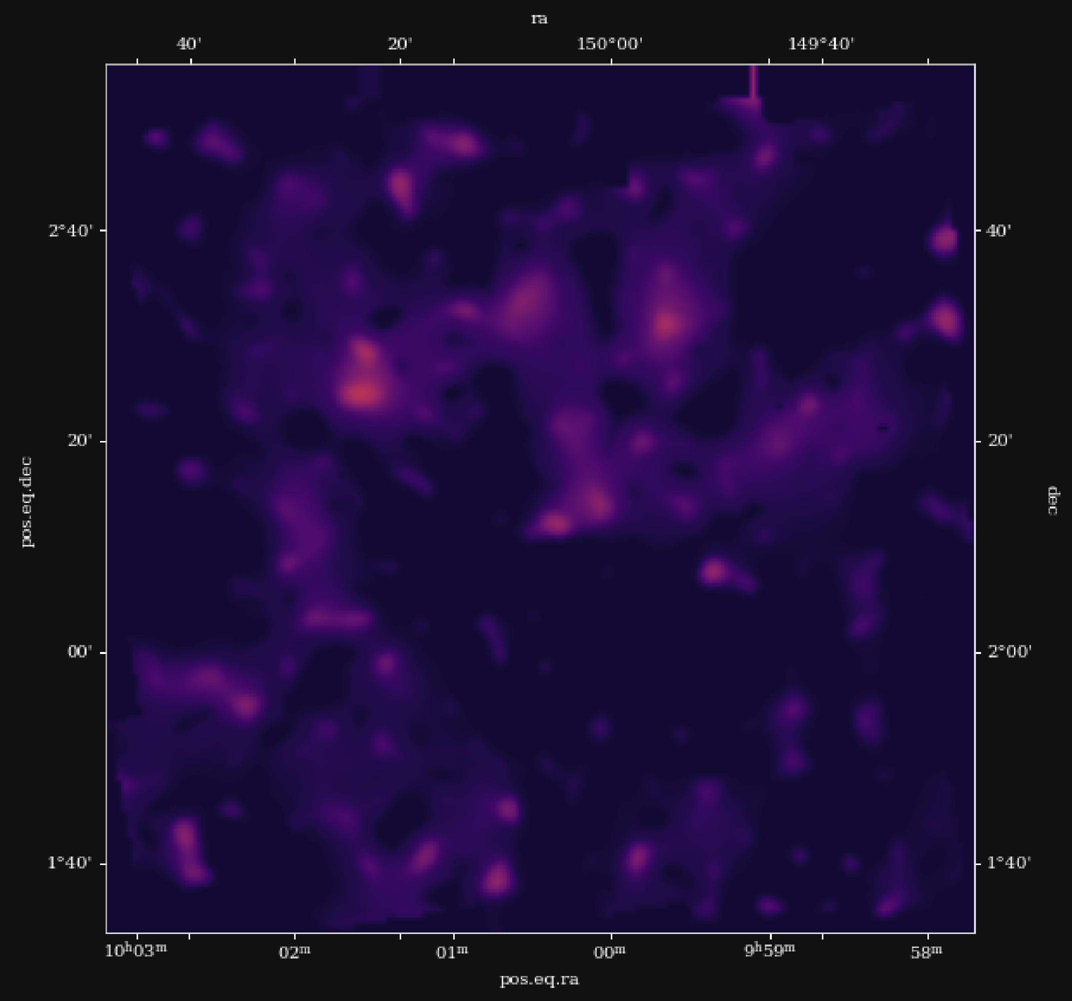
Remy et al. (2022) Posterior samples



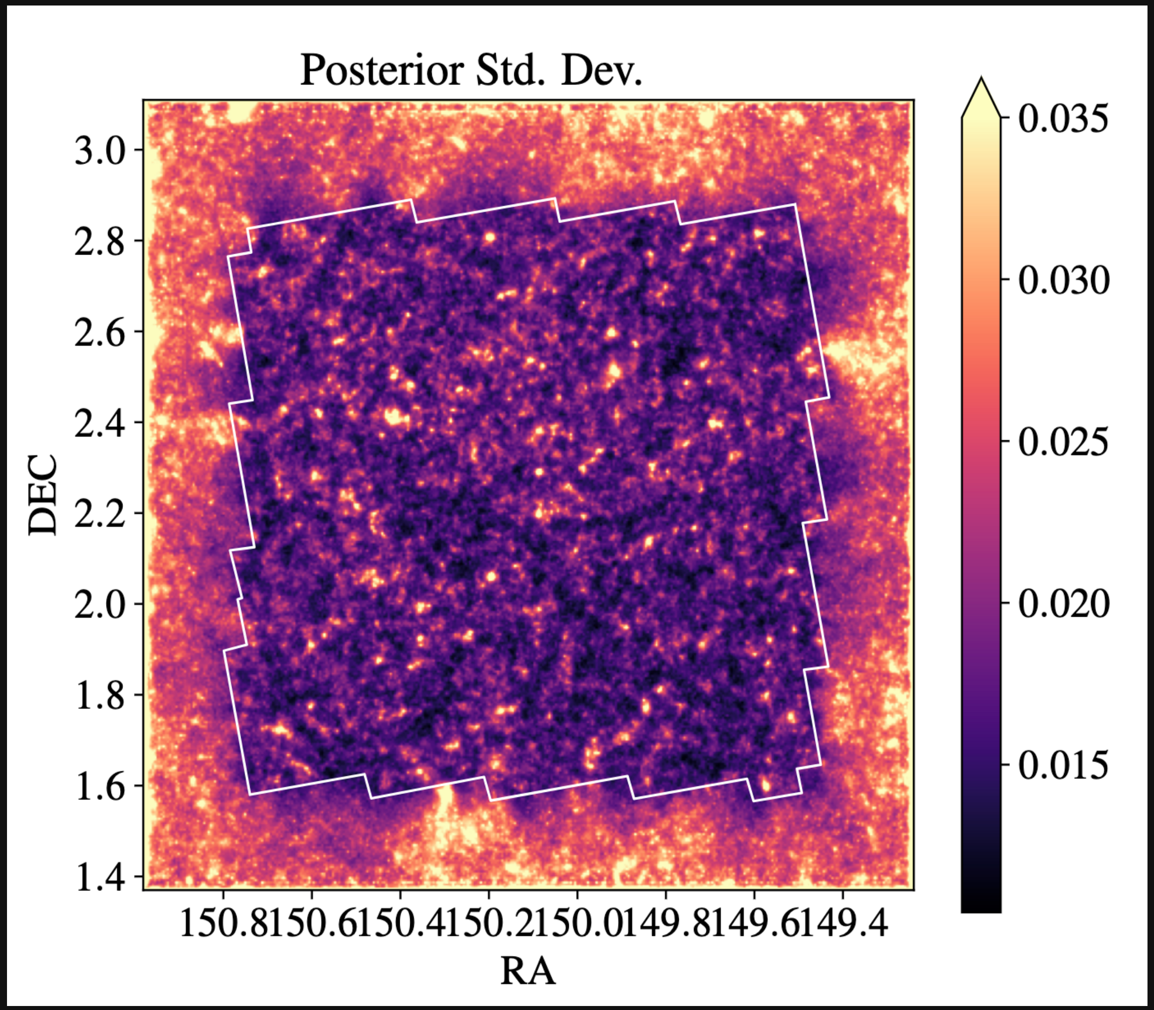
Reconstruction of the HST/ACS COSMOS field

1.637 square degree, 64.2 gal/arcmin^2

Massey et al. (2007)



Remy et al. (2022)

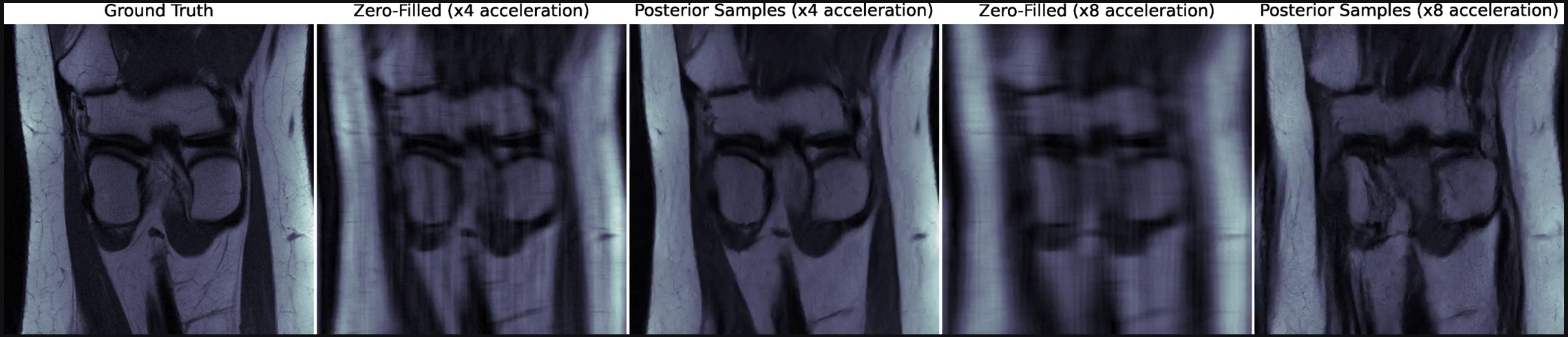


Uncertainty quantification in Magnetic Resonance Imaging (MRI)

stat.ML arXiv:2011.08698

Ramzi, Remy, **Lanusse** et al. 2020

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



$\Longrightarrow$ We can see which parts of the image are well constrained by data, and which regions are **uncertain**.

Takeaways

- Hybrid physical/deep learning modeling:
  - Deep generative models can be used to provide **data driven (simulation based) priors**.

  - **Explicit likelihood**, uses of all of our physical knowledge.
    $\Longrightarrow$ The method can be applied for varying PSF, noise, or even different instruments!

- Demonstrated the efficiency of annealing Hamiltoninan Monte Carlo for high dimensional posterior sampling.

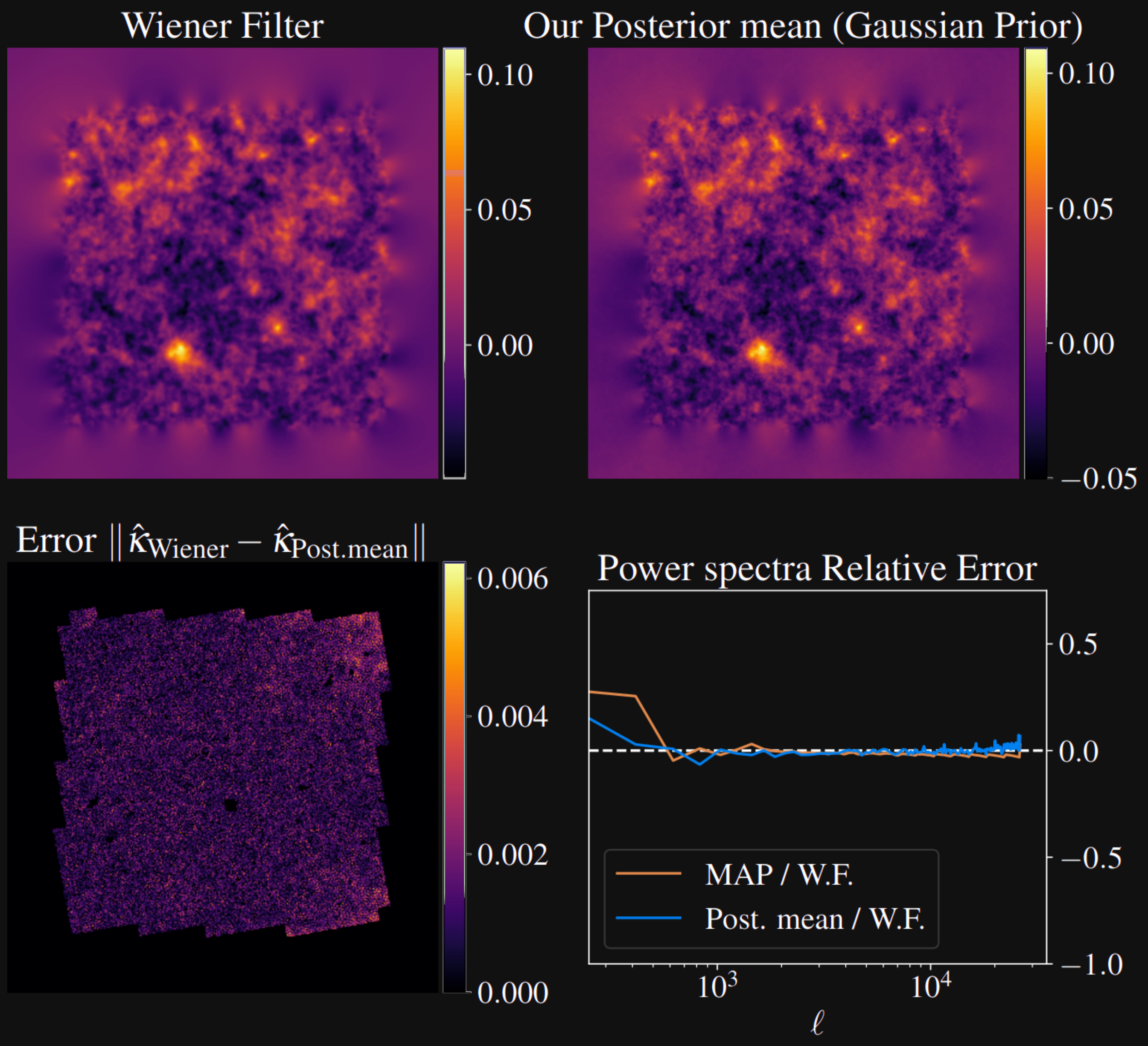
- We implemented a new class of mass mapping method, providing the full posterior
  $\Longrightarrow$ Find the highest quality convergence map of the COSMOS field online: https://zenodo.org/record/5825654

astro-ph.CO arXiv:2201.05561

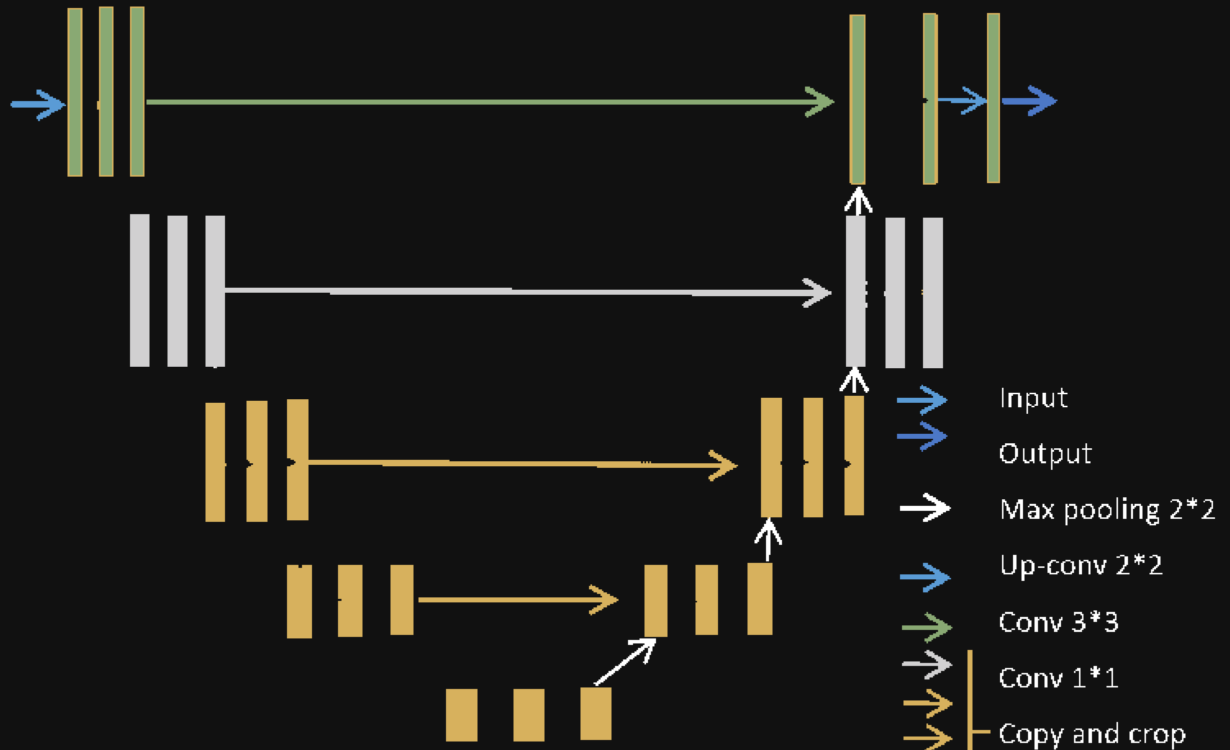
github.com/CosmoStat/jax-lensing (JAX & TFP!)

Thank you!

Validating Bayesian Posterior in Gaussian case



Training a Neural Score Estimator in practice



A standard UNet

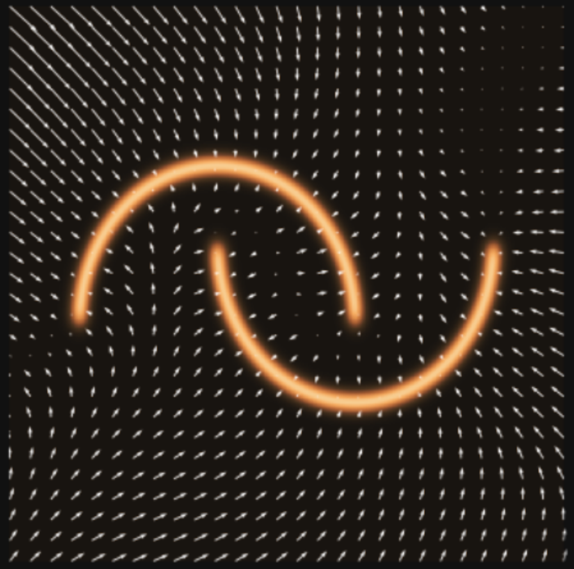
- We use a very standard residual UNet, and we adopt a residual score matching loss:

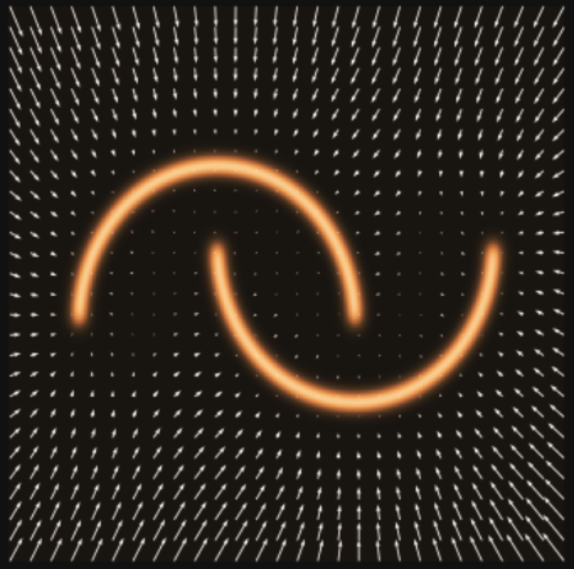
$$L_{DSM} = \underset{\boldsymbol{x} \sim P}{\mathbb{E}} \underset{\substack{\boldsymbol{u} \sim \mathcal{N}(0, I) \\ \sigma_s \sim \mathcal{N}(0, s^2)}}{\mathbb{E}} \parallel \boldsymbol{u} + \sigma_s \boldsymbol{r}_\theta(\boldsymbol{x} + \sigma_s \boldsymbol{u}, \sigma_s) \parallel_2^2$$

$\Longrightarrow$ direct estimator of the score $\nabla \log p_\sigma(x)$

- Lipschitz regularization to improve robustness:

Without regularization　　With regularization





- The likelihood term is **known analytically**.
- There is **no close form expression for the full non-Gaussian prior** of the convergence.
- We learn a **hybrid Denoiser**: theoretical Gaussian on large scale, data-driven on small scales using N-body simulations.

$$\underbrace{\nabla_{\boldsymbol{\kappa}} \log p(\boldsymbol{\kappa})}_{\text{full prior}} = \underbrace{\nabla_{\boldsymbol{\kappa}} \log p_{th}(\boldsymbol{\kappa})}_{\text{gaussian prior}} + \underbrace{\boldsymbol{r}_{\theta}(\boldsymbol{\kappa}, \nabla_{\boldsymbol{\kappa}} \log p_{th}(\boldsymbol{\kappa}))}_{\text{learned residuals}}$$