Sequential simulation-based inference for cosmological initial conditions

Oleg Savchenko

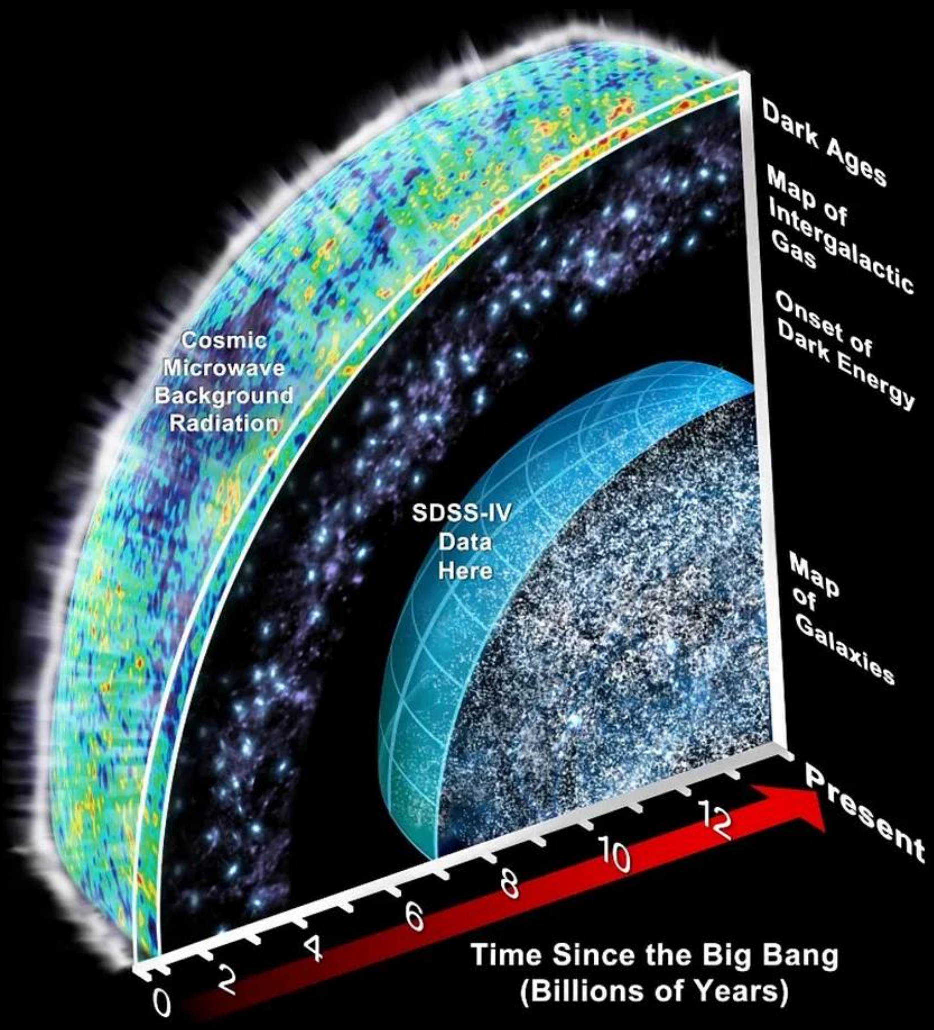
Work with: Guillermo Franco Abellán, Florian List, Noemi Anau Montel, Christoph Weniger

UNIVERSITEIT VAN AMSTERDAM

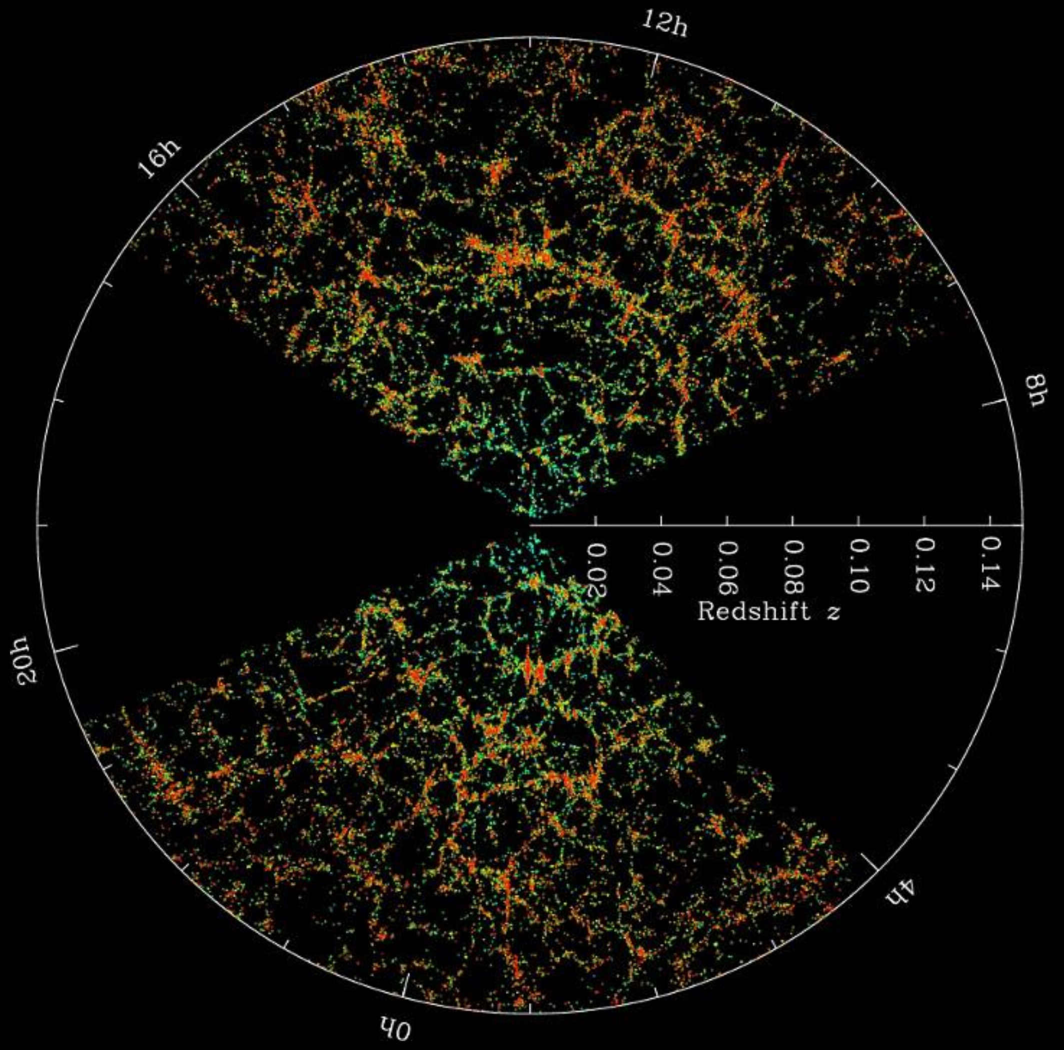
GRAPPA
GRavitation AstroParticle Physics Amsterdam

EUROPEAN AI FOR FUNDAMENTAL PHYSICS CONFERENCE
EuCAIFCon 2025

Large scale structure

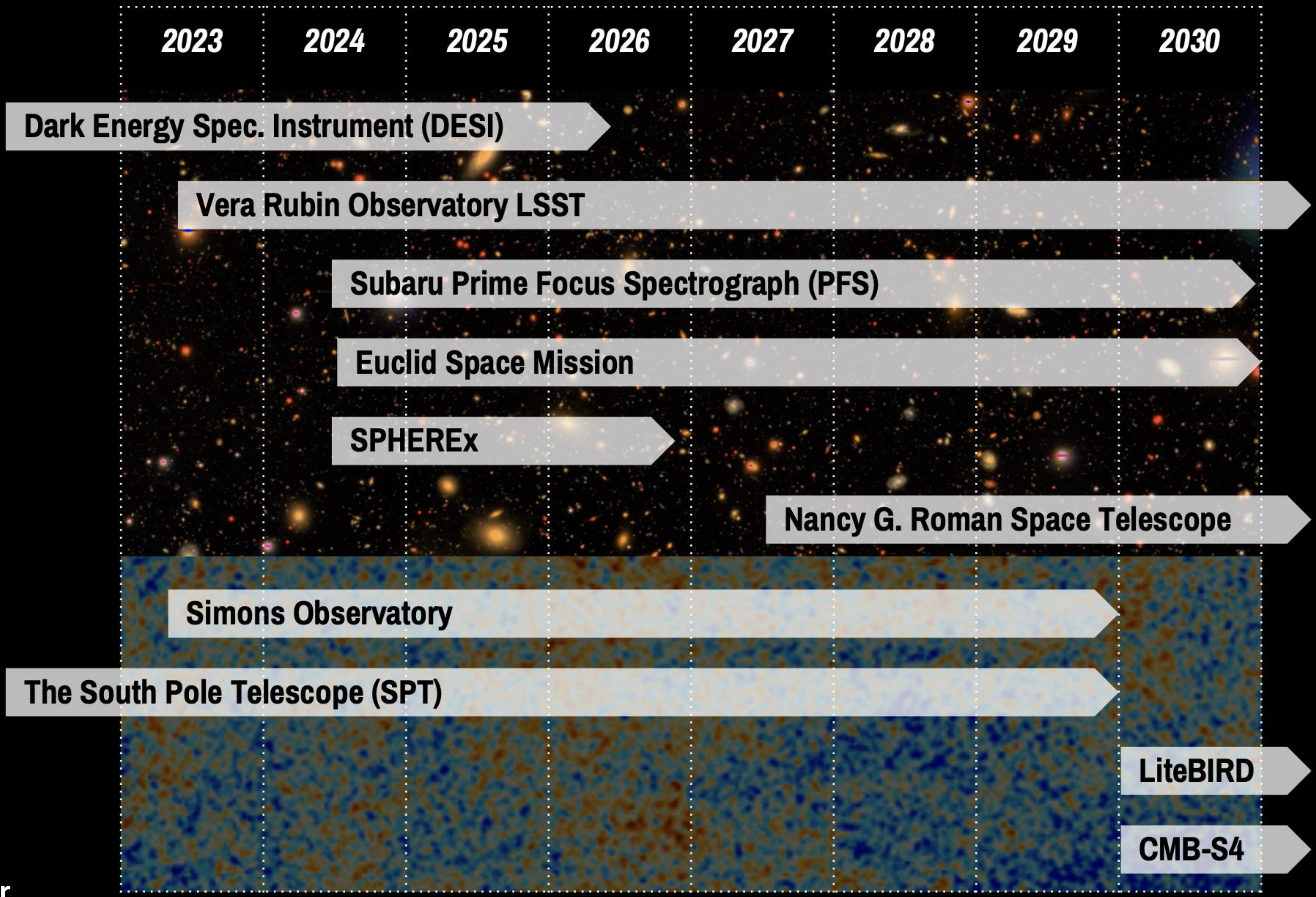


https://mapoftheuniverse.net/



SDSS collaboration

Next decade



Picture credit: **A. Bayer**

How to analyse LSS data?

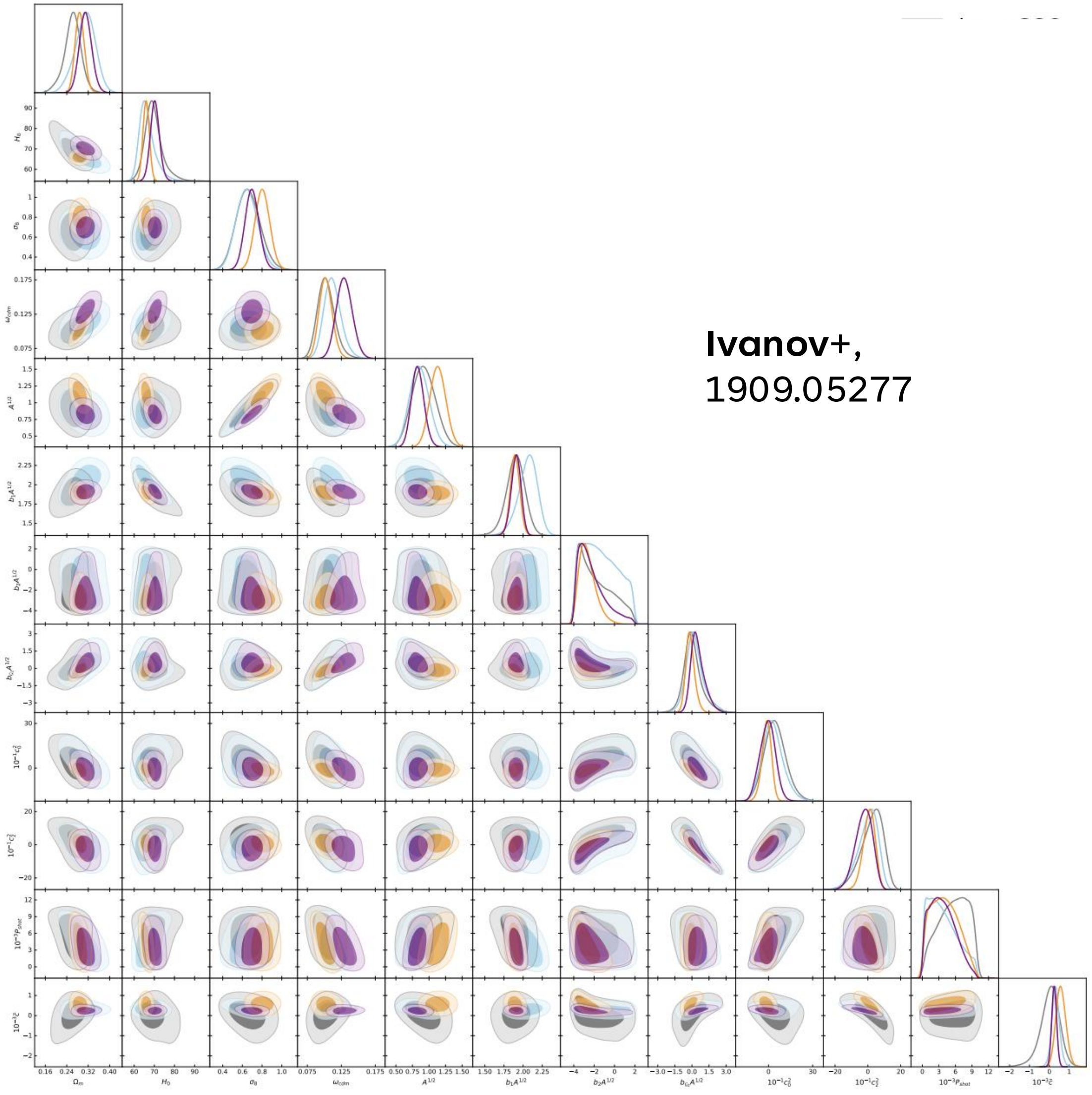


Figure 11: The triangle plot for cosmological and nuisance parameters of four independent BOSS datasets.

Ivanov+, 1909.05277

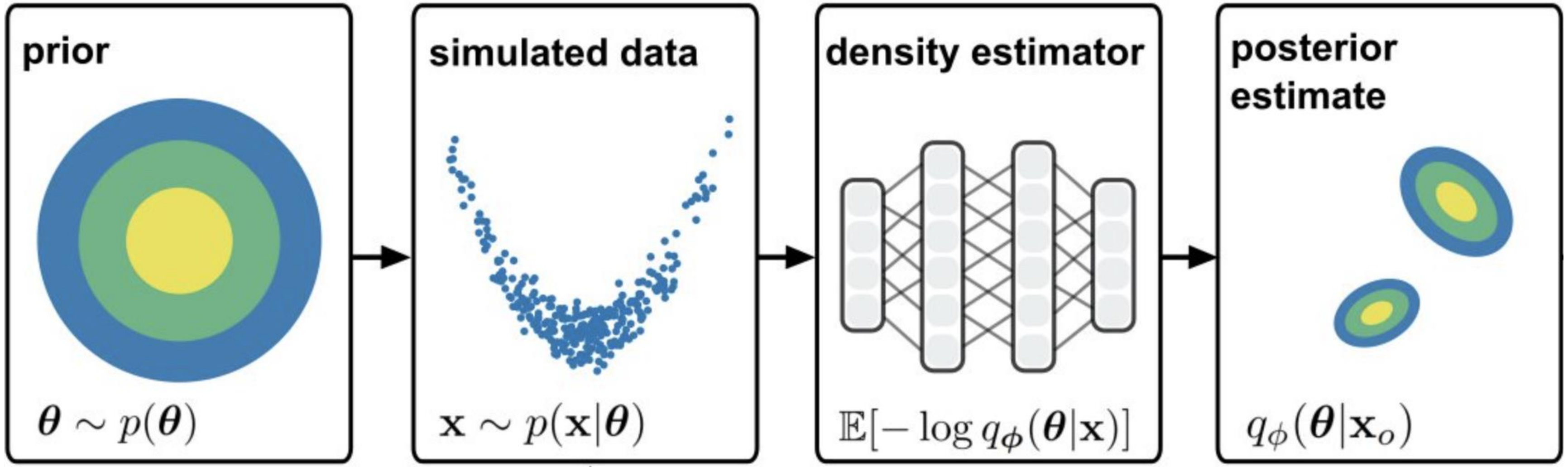
Classic approach:

- Come up with some summary statistic ***s****(data)*
- Develop theory predictions for ***s***
- Construct an analytic likelihood model (usually Gaussian)
- Use Bayes theorem and run MCMC:

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})} p(\boldsymbol{z}).$$

Simulation-based inference

How to perform inference if all that you have is a forward simulator that can generate samples?



Picture credit:
Deistler+, 2210.04815

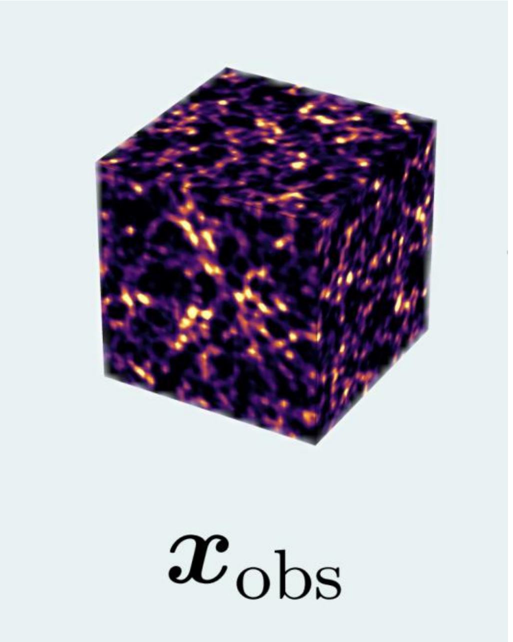
$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})} p(\boldsymbol{z}).$$

- Neural Posterior Estimation
- Neural Likelihood Estimation
- Neural Ratio Estimation

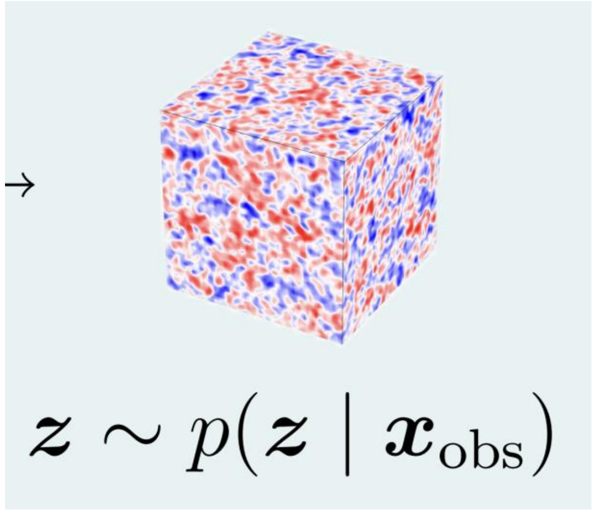
See review:
Cranmer+, 1911.01429

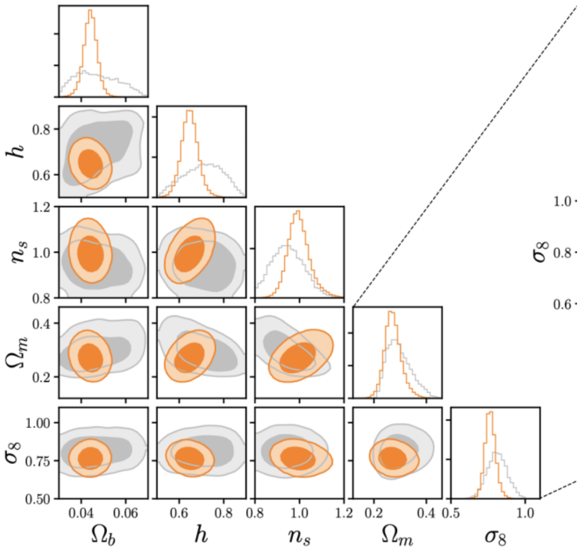
Field-level inference

The whole field contains much more information than some summary like a power spectrum!



$$p(\boldsymbol{x}_{\mathrm{IC}}, \boldsymbol{\theta} | \boldsymbol{x}_{\mathrm{obs}}) = \,?$$





Initial conditions reconstruction

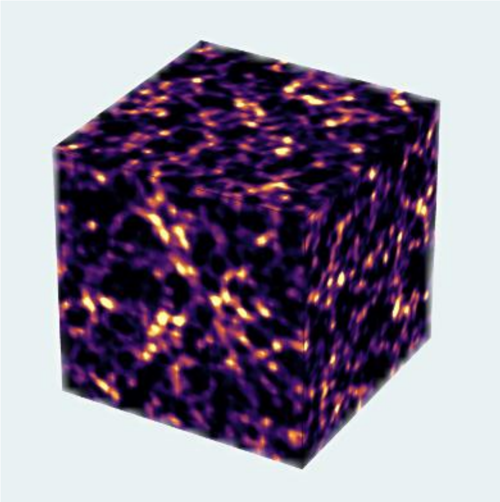
Very early universe had very simple properties!

→ feasible way to do field reconstruction is to infer these initial conditions 🙂

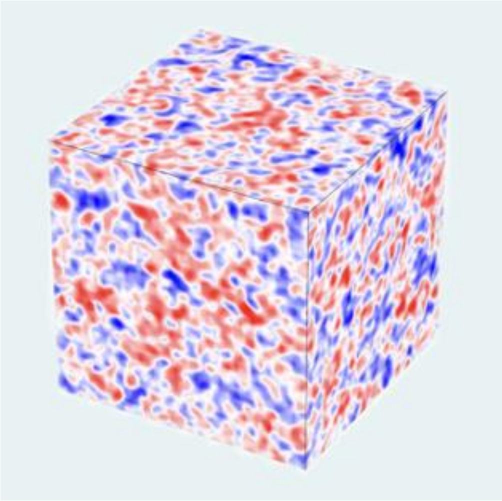
$$P(\boldsymbol{\delta}^{\mathrm{IC}} \mid \{N_i^g\}) = \frac{P(\boldsymbol{\delta}^{\mathrm{IC}})P(\{N_i^g\} \mid G_i(\boldsymbol{\delta}^{\mathrm{IC}}))}{P(\{N_i^g\})}$$

$\boldsymbol{\delta}^{\mathrm{IC}}$ - initial density field

$\{N_i^g\}$ - galaxy catalog data



z = 0

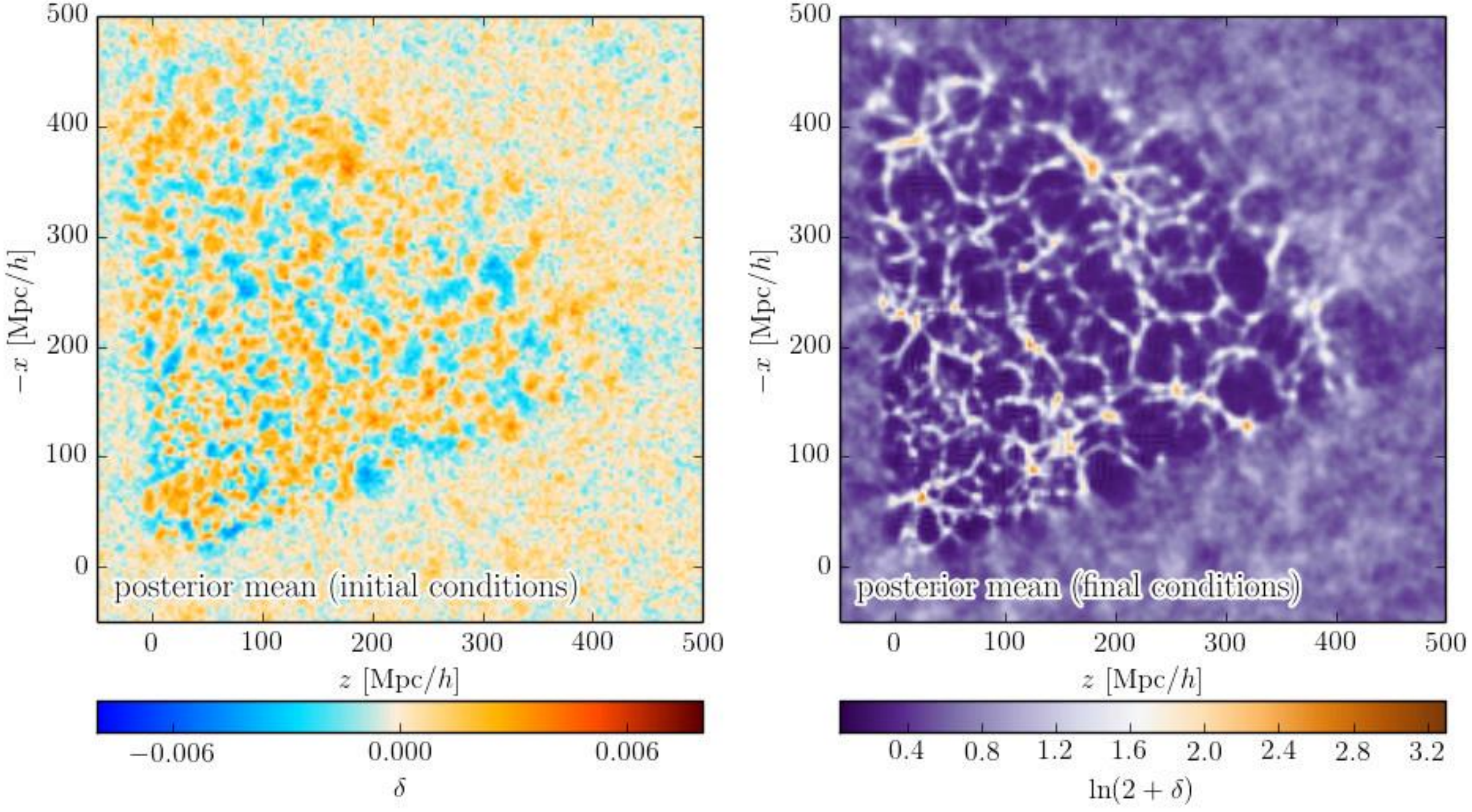


z = 1000

Gaussian initial conditions

Hamiltonian Monte Carlo

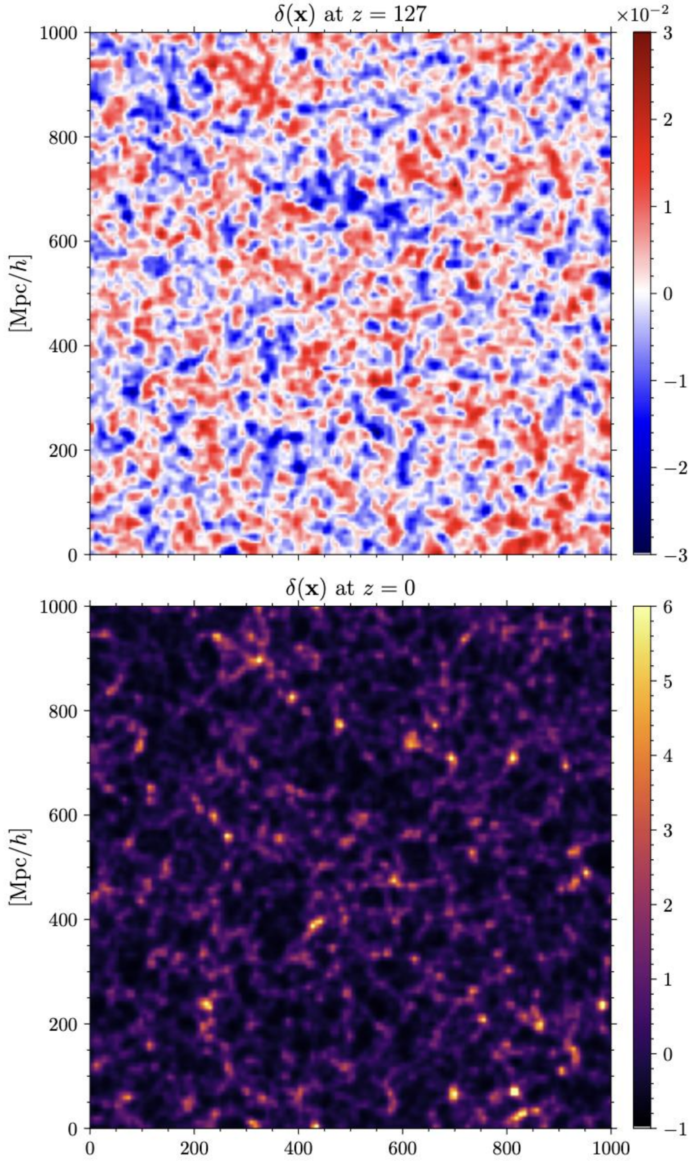
Jasche+, 1203.3639, 1806.11117



Samples produced with **Bayesian Origin Reconstruction from Galaxies** (**BORG**) algorithm

- Most developed method so far
- Explicit likelihood
- Requires gradients from the simulator
- Computationally expensive

Our setting



Training data: **2000** 128^3 (1 *Gpc/h*)^3 *Quijote* N-body simulations

128^3 resolution: ~**million**-dimensional parameter space!

- Want to explore the full posterior, not only get point estimates
- Want to keep things as simple as possible: model the likelihood as

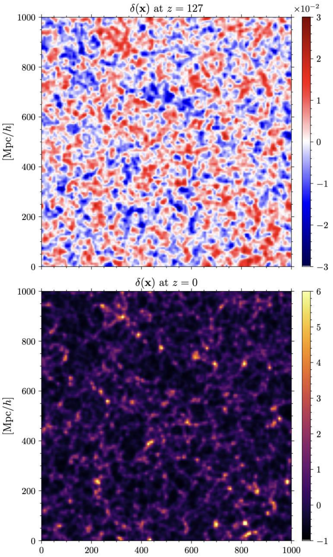
$$p(\boldsymbol{x}|\boldsymbol{z}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \hat{\mathbf{z}}_{\boldsymbol{\theta}}(\boldsymbol{x}))^T \boldsymbol{Q}^L_{\boldsymbol{\theta}} (\boldsymbol{z} - \hat{\mathbf{z}}_{\boldsymbol{\theta}}(\boldsymbol{x}))\right\}$$

- Trainable parts of the model: estimator $\boldsymbol{\mu}_{MAP}(\boldsymbol{x})$ and **Q** matrix Fourier diagonal values

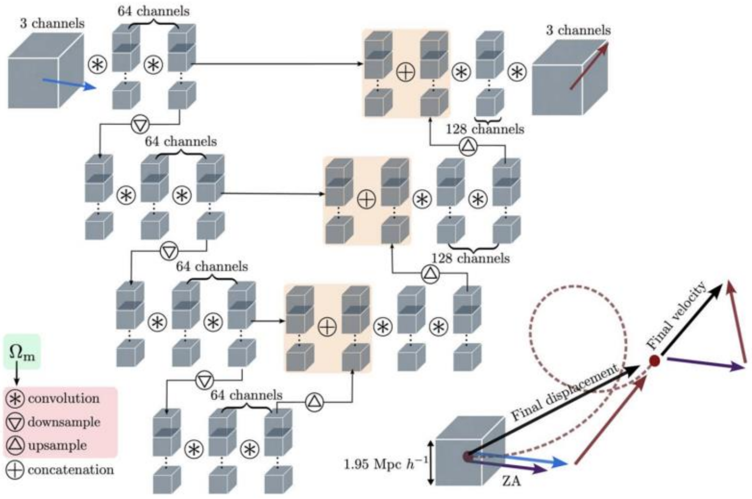
OS+, 2502.03139

Our approach

Apply a U-Net to estimate $\boldsymbol{\mu}_{MAP}(\boldsymbol{x})$



Training data



map2map U-Net

Jamieson+, 2206.04594

Train the model with a simple MAP loss function

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N}\left\{(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^T \boldsymbol{Q}_{\boldsymbol{\theta}}\,(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i))\right\} - \frac{N}{2}\,\mathrm{tr}\log \boldsymbol{Q}_{\boldsymbol{\theta}}$$

and a Fourier-diagonal $\boldsymbol{Q}$ matrix

Learn precision matrix and the embedding network simultaneously

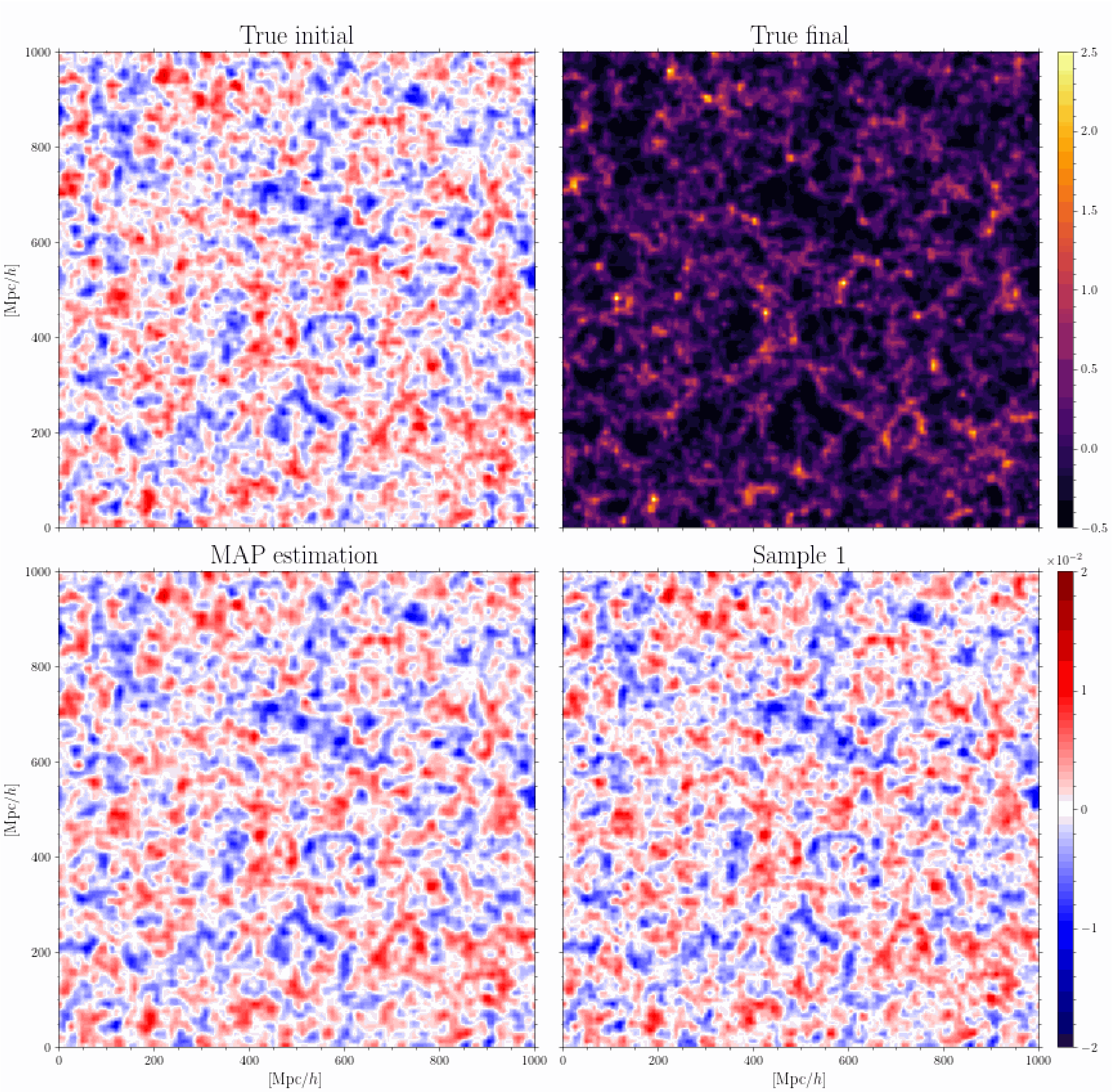
Super-fast sampling from a Gaussian

OS+, 2502.03139

Results

True

Inferred



$V = (1\ Gpc/h)^3$

$N_{grid}=128^3$

1.5 hrs of training on 1 GPU

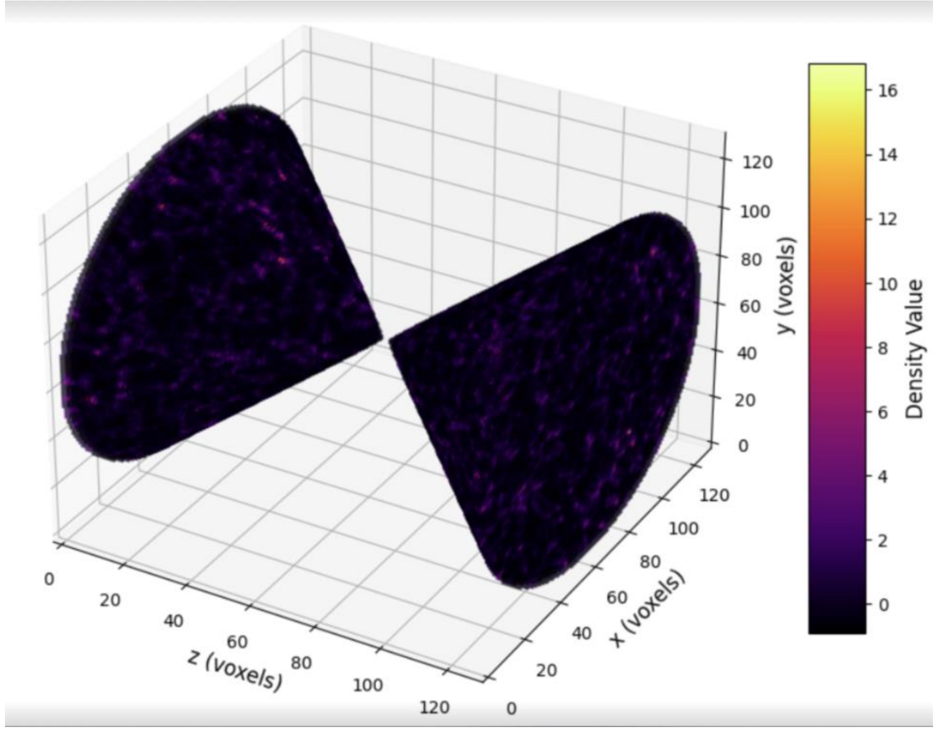
NVIDIA 40GB A100

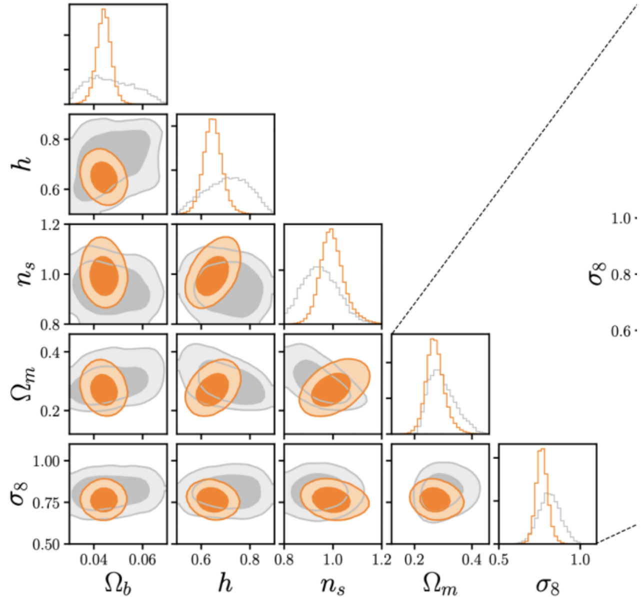
10^3 samples in < 3 s

Next steps

Moving to a more realistic setting:

- Survey effects: mask, spatially varying noise, etc.
- Varying cosmological parameters

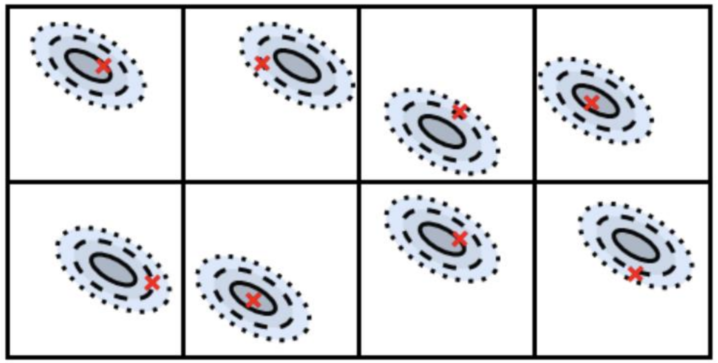




Amortized vs sequential inference

- Fast inference for new $\mathbf{x}_{obs}$
- High simulation cost for training
- Can be not very precise

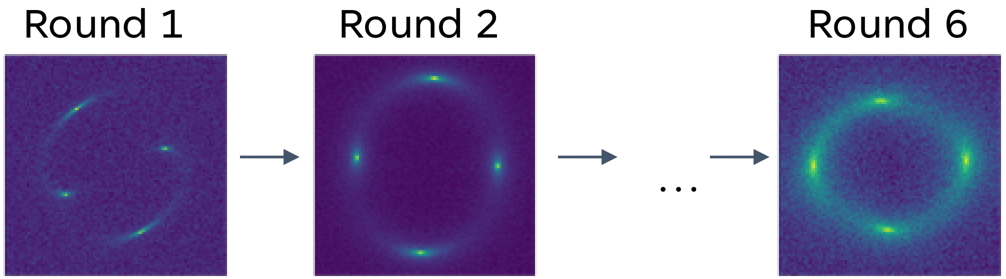
- Slow inference for new $\mathbf{x}_{obs}$
- Low simulation cost
- High precision



Cole+, 2111.08030

$p(\theta|\mathbf{x}) \quad \forall \mathbf{x} \sim p(\mathbf{x})$

simultaneously estimates the posteriors for all simulated observations



Animation credit:
N. Anau Montel

Sequential inference in Falcon

- Framework for **distributed computing** on multiple nodes
- Training set evolves during training for sequential inference: new samples are drawn from the **proposal distribution**
- Some nodes **simulate**, some nodes train to do **inference**

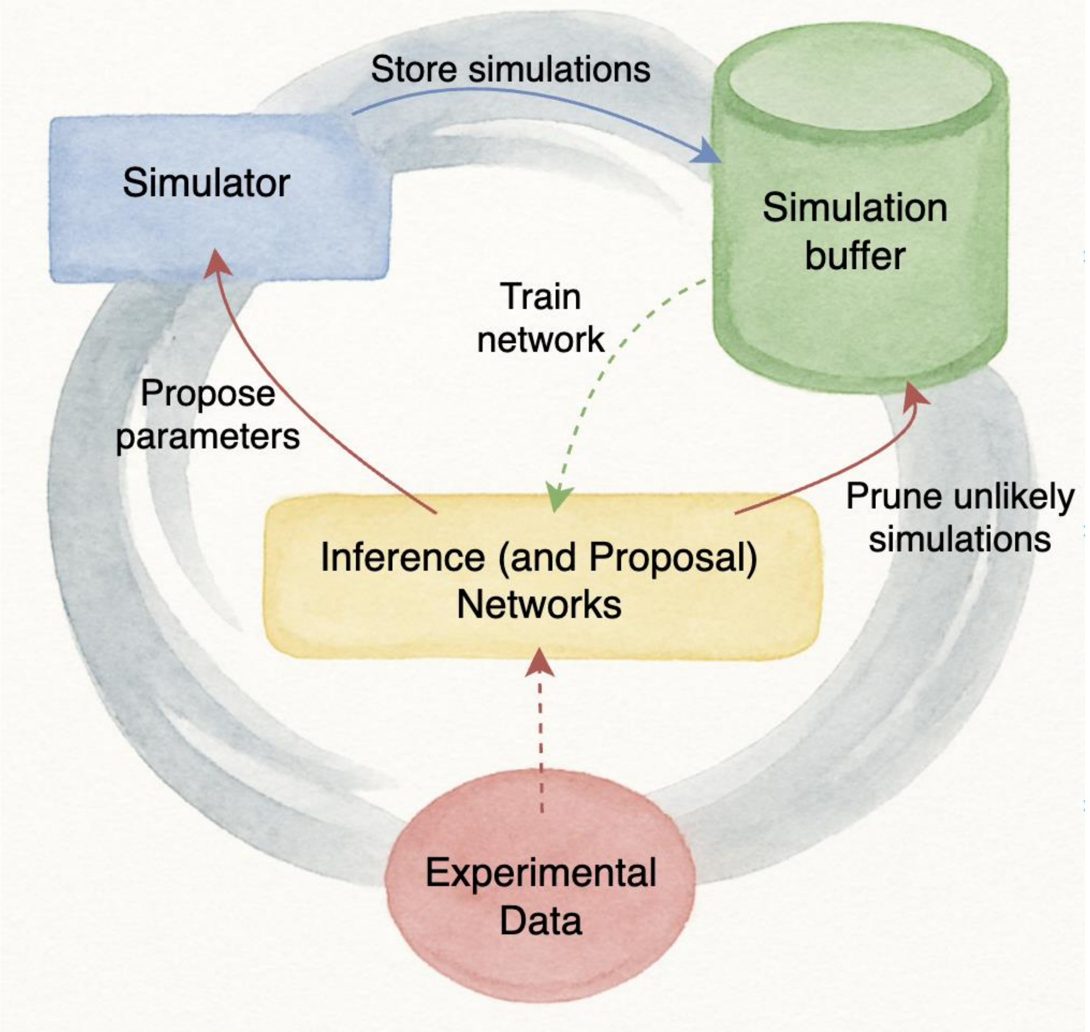
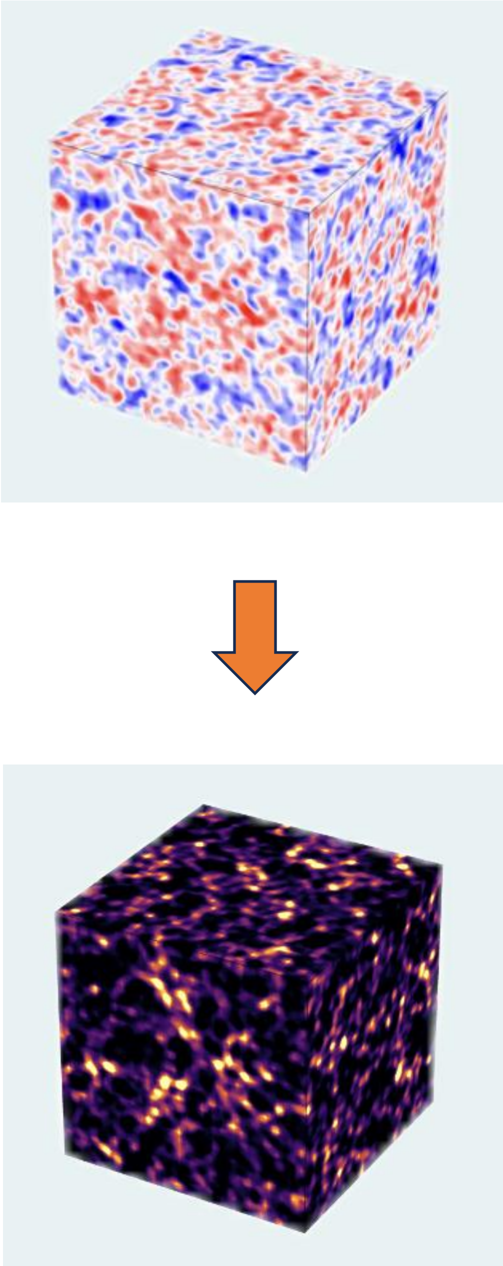


Diagram credit:
C. Weniger

Forward model

Code developed by **F. List, O. Hahn et al.**

DISCO-DJ
differentiable simulations for cosmology
done with jax



Differentiable, highly-parallelisable particle-mesh simulator written in JAX.

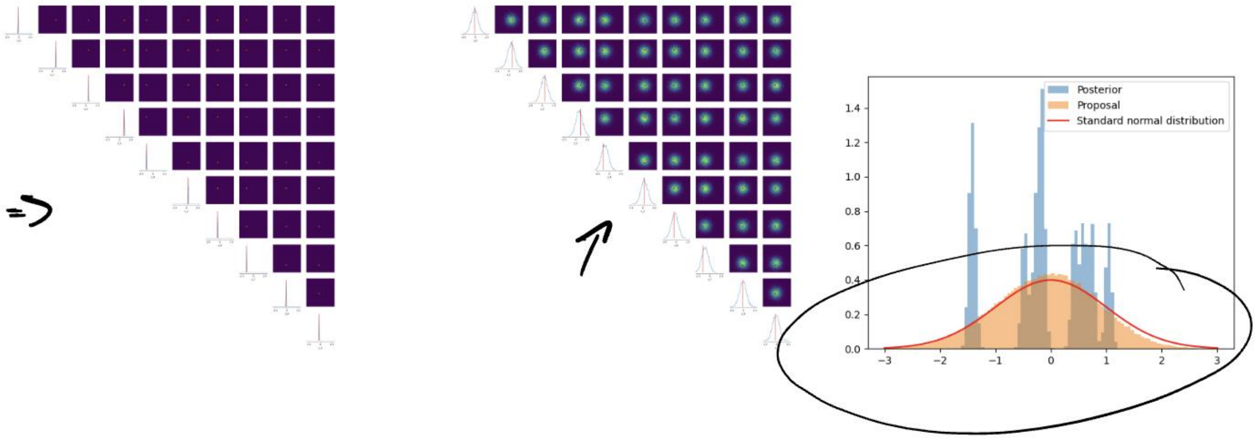
Hierarchical model implemented in Falcon:

$$\begin{tikzcd} \Omega \arrow[r] \arrow[rrr, bend left] & p(k) \arrow[r] & \delta_{ic} \arrow[r] & \delta_f \arrow[r] & \delta_{obs} \end{tikzcd}$$

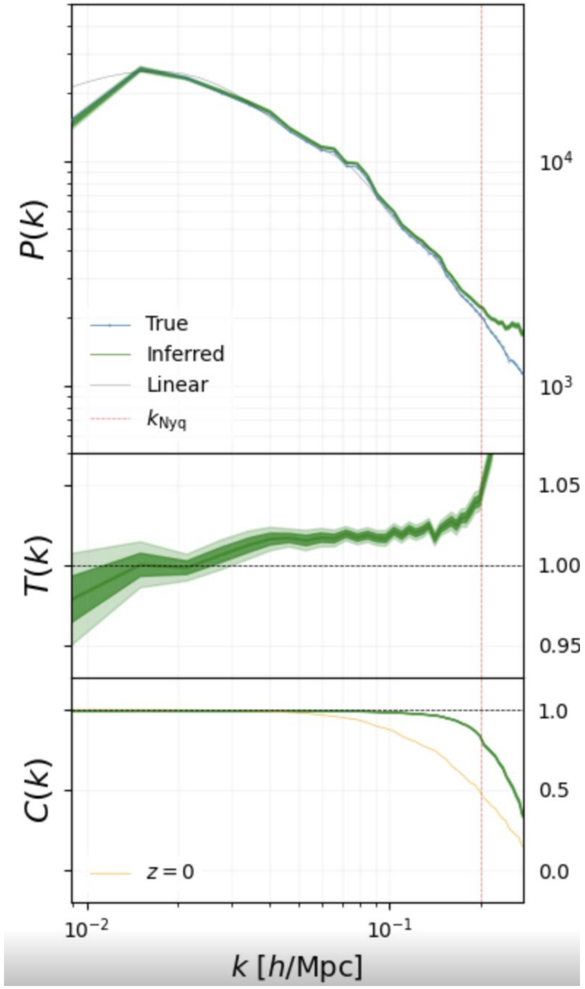
Seuential inference in Falcon

Why is this hard in high dimensions? Naïve way of tempering the likelihood fails

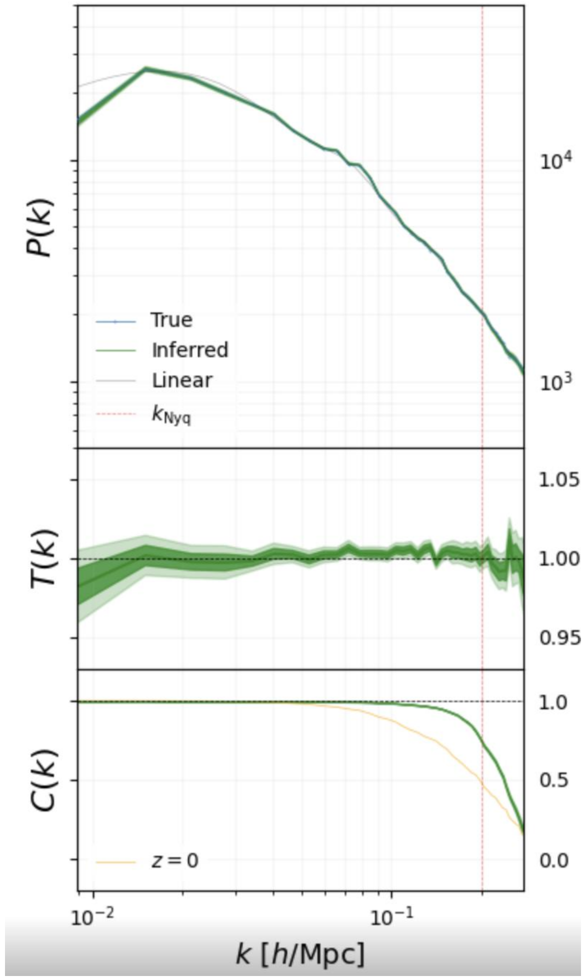
Raising likelihood to some power produces samples with incorrect summaries



Tempering the likelihood

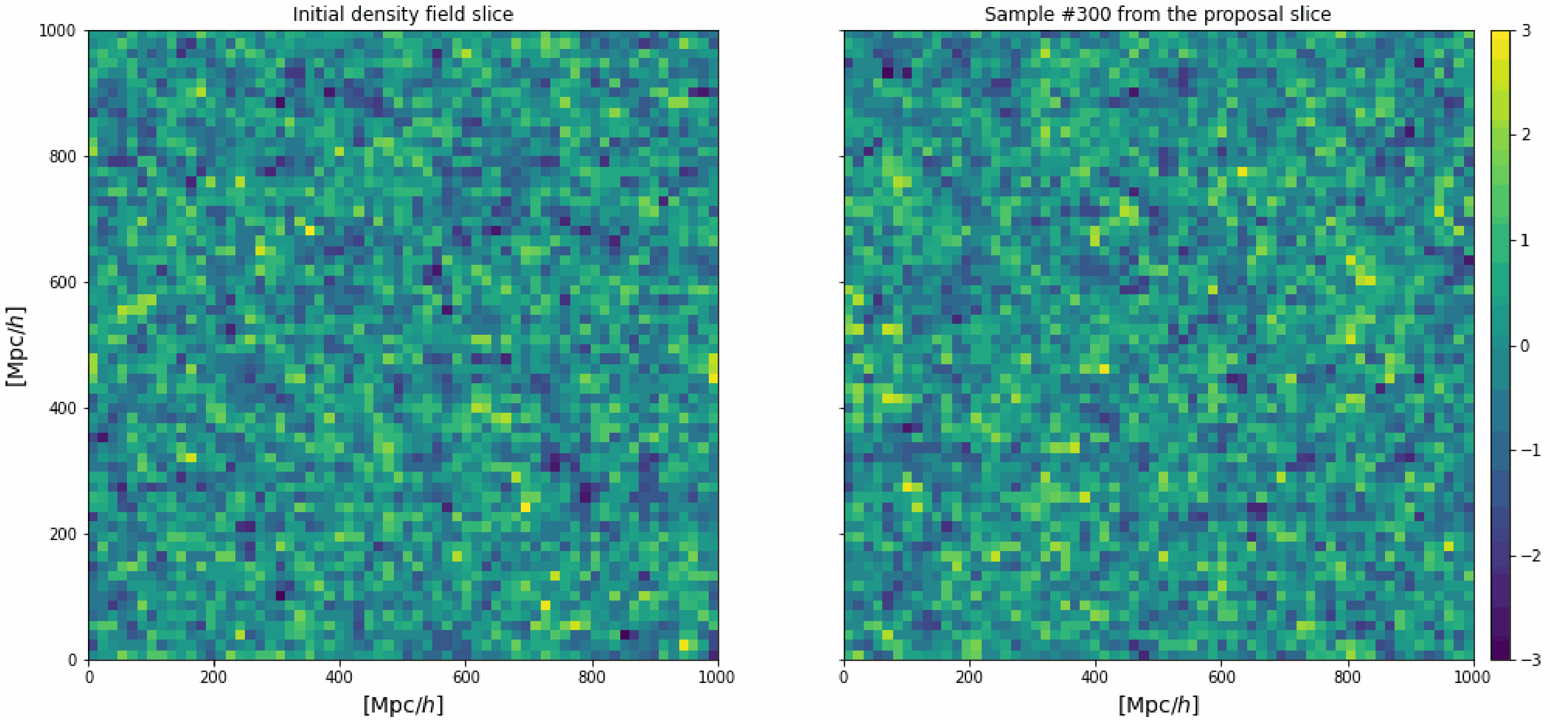


VS



- Need to have a **proposal distribution**
- Can temper the likelihood by adding more noise to the data

Tempering the likelihood

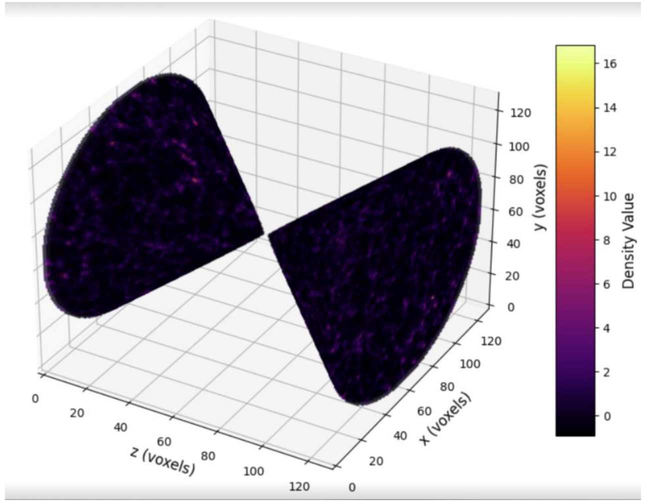


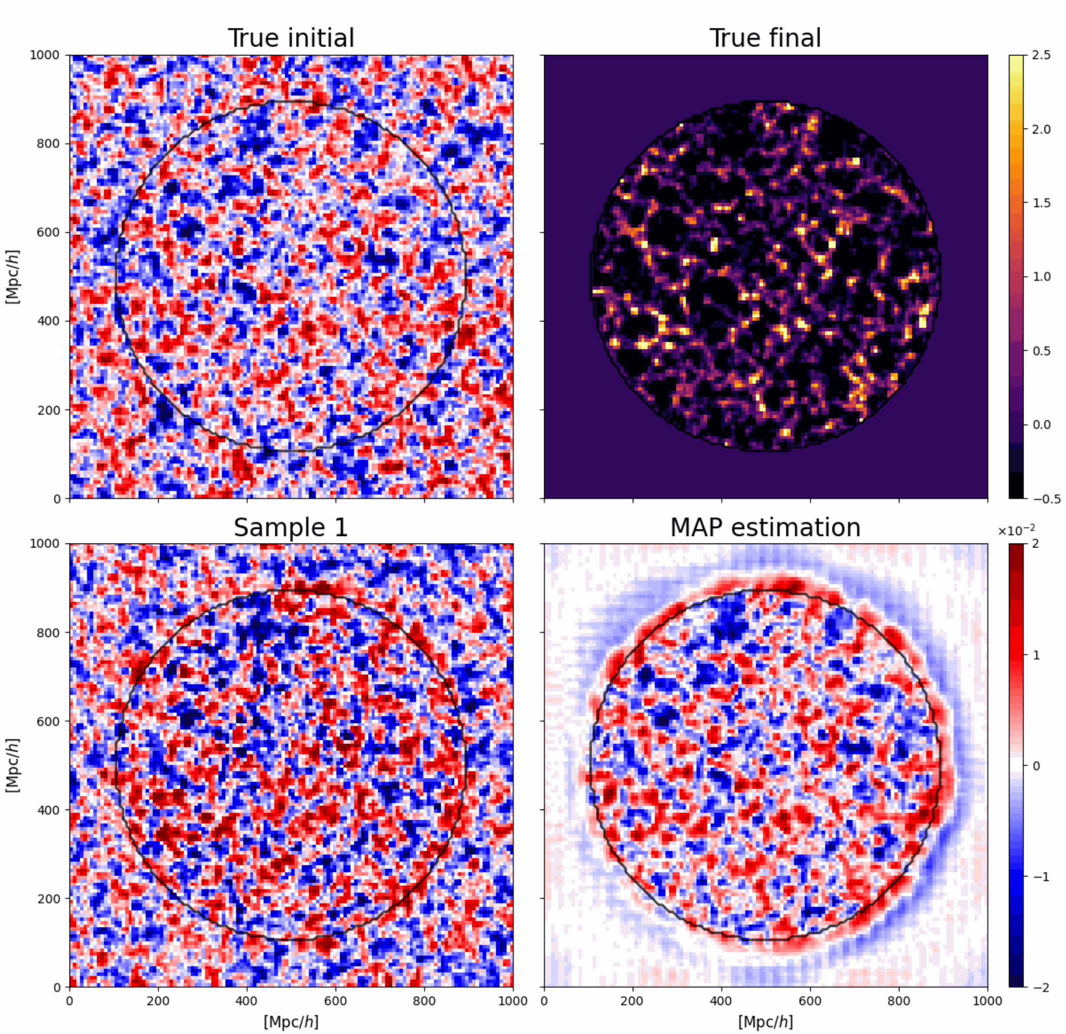
- Need to have a **proposal distribution**
- Can temper the likelihood by adding specific noise to the proposal data:

$$\boldsymbol{Q}_{\text{eff}} = \left(\boldsymbol{Q}^P + \gamma\,\boldsymbol{Q}^L_{\boldsymbol{\theta}}\right)\left[\boldsymbol{Q}^P + \left(2\gamma - \gamma^2\right)\boldsymbol{Q}^L_{\boldsymbol{\theta}}\right]^{-1}\left(\boldsymbol{Q}^P + \gamma\boldsymbol{Q}^L_{\boldsymbol{\theta}}\right)$$

Incomplete data

Field is constrained in the observed region and sampled from the prior in the unobserved region.



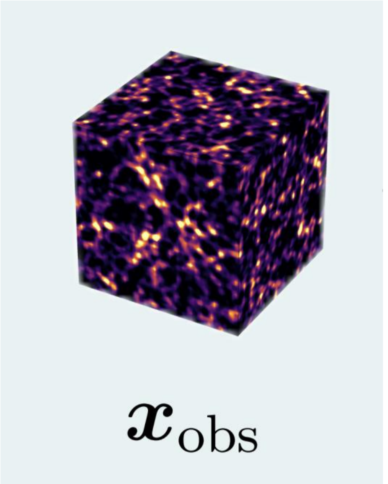


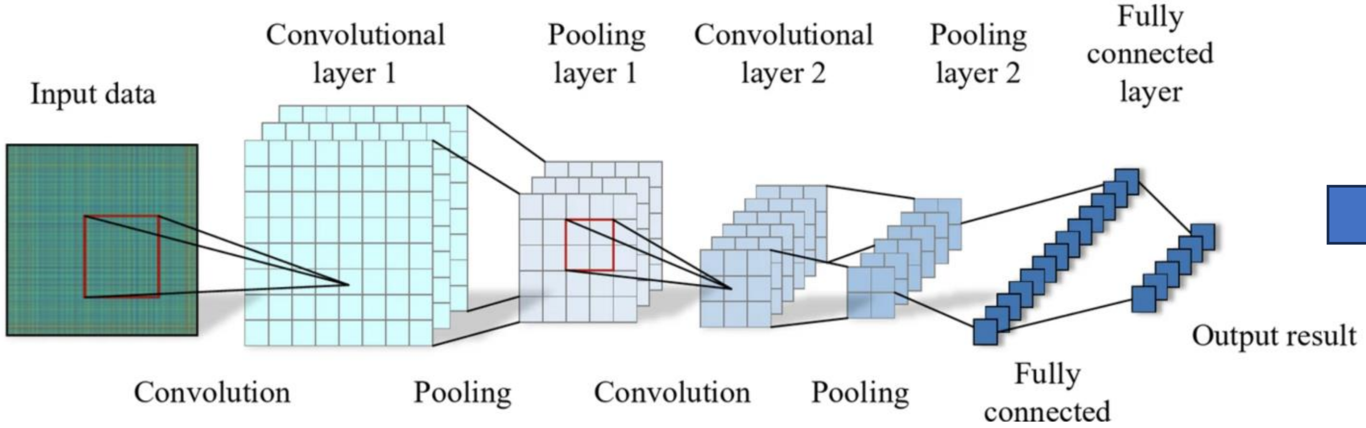
PRELIMINARY

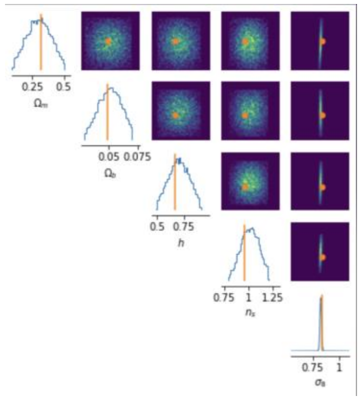
Animation credit:
A. Orban

Parameter inference

CNN trainable summary statistic







$$p(\boldsymbol{\delta}_{\rm IC}, \boldsymbol{\theta} | \boldsymbol{\delta}_{\rm obs}) = p(\boldsymbol{\delta}_{\rm IC} | \boldsymbol{\theta}, \boldsymbol{\delta}_{\rm obs}) \times p(\boldsymbol{\theta} | \boldsymbol{\delta}_{\rm obs})$$

Summary

- Reconstruction of cosmological **initial conditions** is an important problem that allows to **analise LSS data in the fullest way**
- **Falcon** framework allows for powerful **distributed computing** applications for **active learning**
- **Tempering the likelihood** in high dimensions can be done by adding a specific amount of additional noise to the proposal
- Mowing towards realistic settings **(incomplete data, varying cosmology)** requiers **sequential SBI**

BACK UP SLIDES

Cosmological simulations

Types:

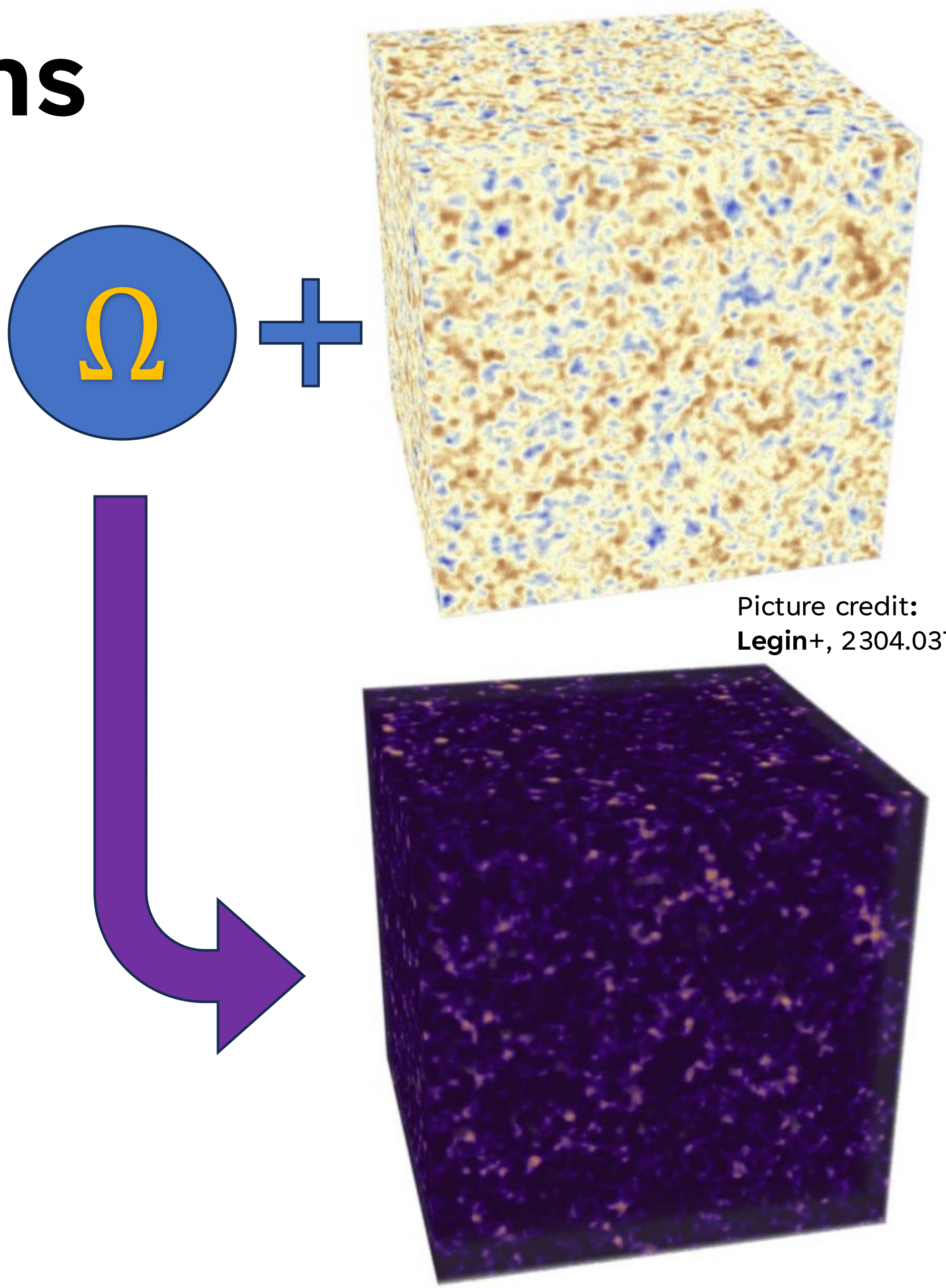
- LPT
- COLA
- Particle mesh
- N-body
- Hydrodynamical

Quijote

CAMELS

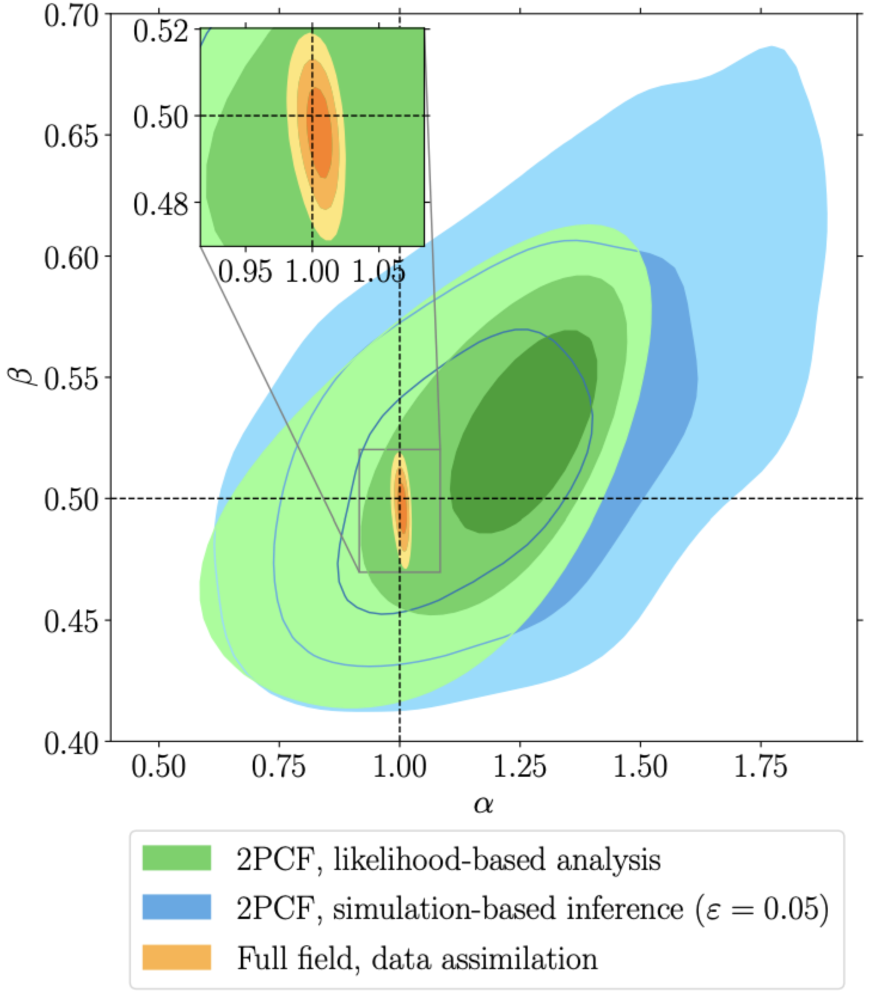
https://quijote-simulations.readthedocs.io/

https://camels.readthedocs.io/



Picture credit: **Legin**+, 2304.03788

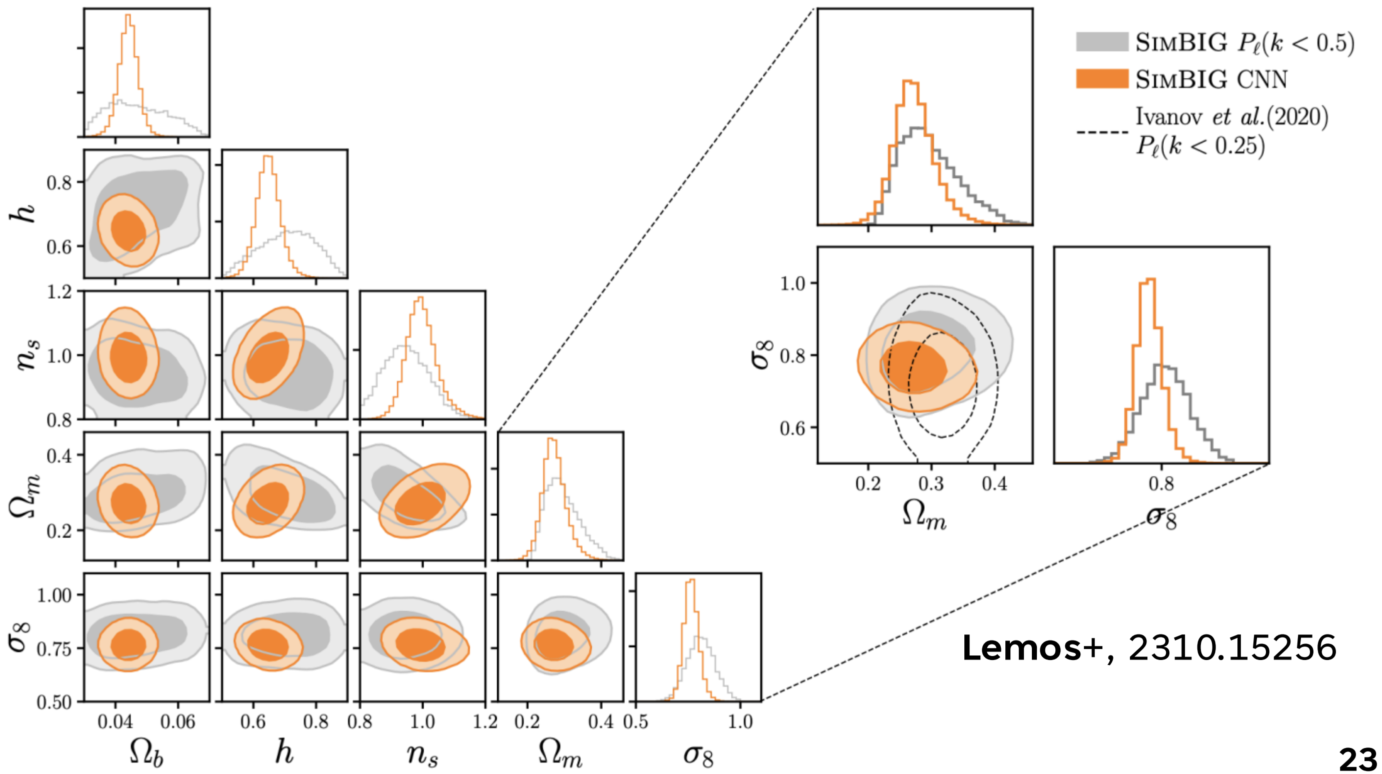
Field-level inference



Leclercq+, 2103.04158

The whole field contains much more information than some summary like a power spectrum!

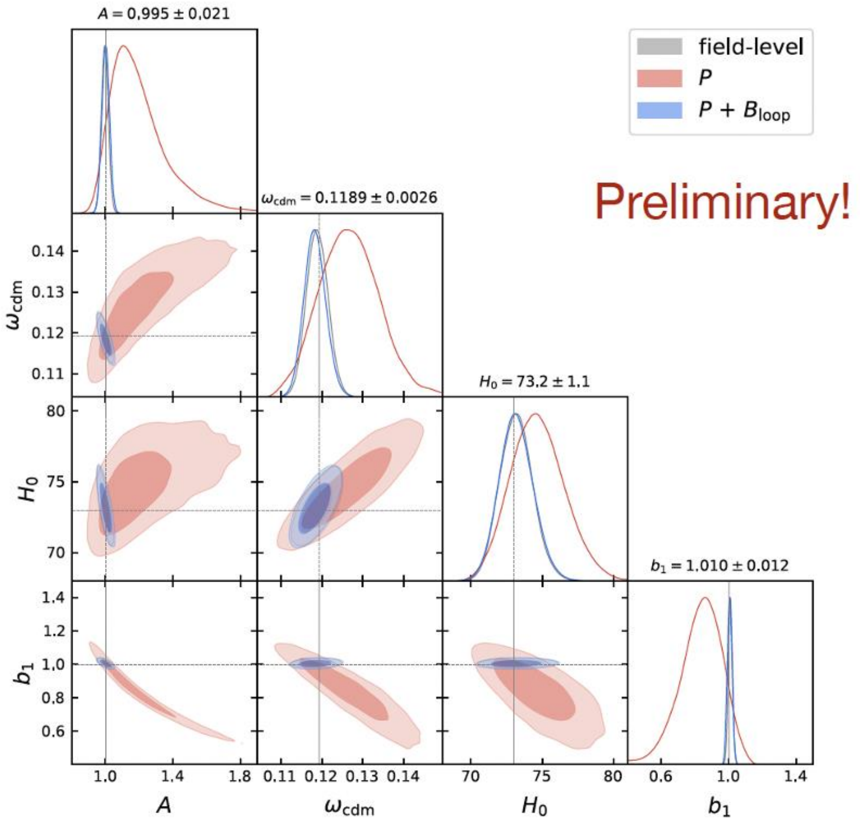
$$\langle \delta_\mathrm{m}(\mathbf{k})\delta_\mathrm{m}(\mathbf{k}')\rangle = (2\pi)^3\delta_\mathrm{D}^3(\mathbf{k}+\mathbf{k}')P_\mathrm{mm}(k)$$



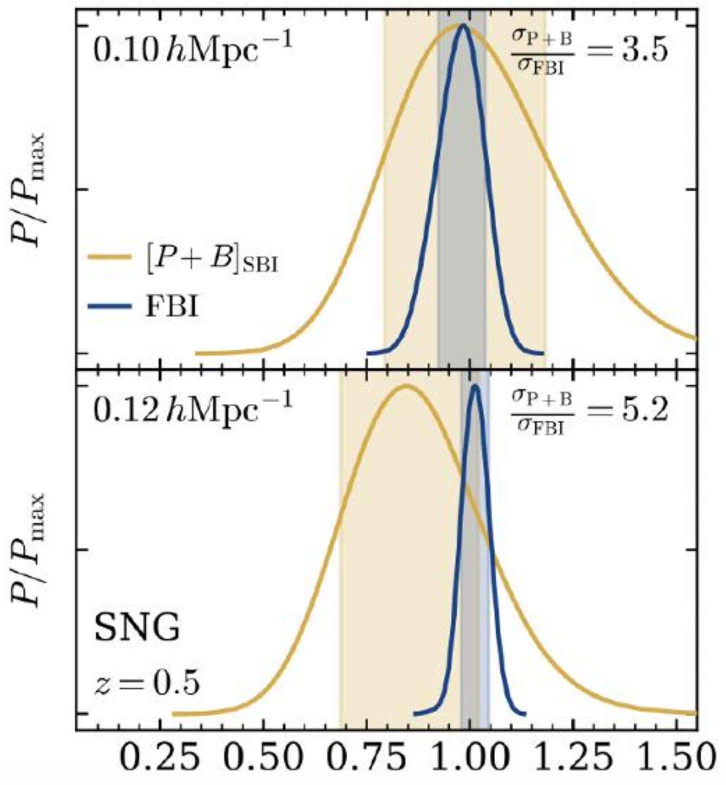
Lemos+, 2310.15256

Ongoing debate

Akitsu et al. (to appear)



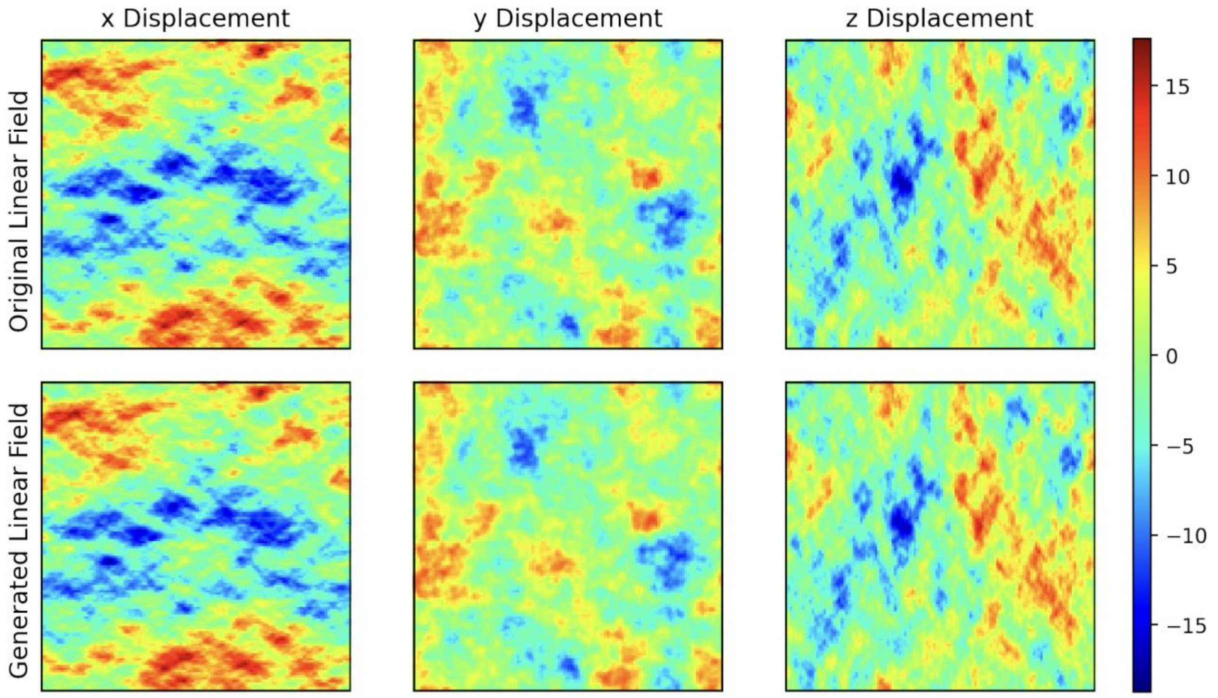
Nguyen, Schmidt, Tucci, Reinecke, Kostic (2024)



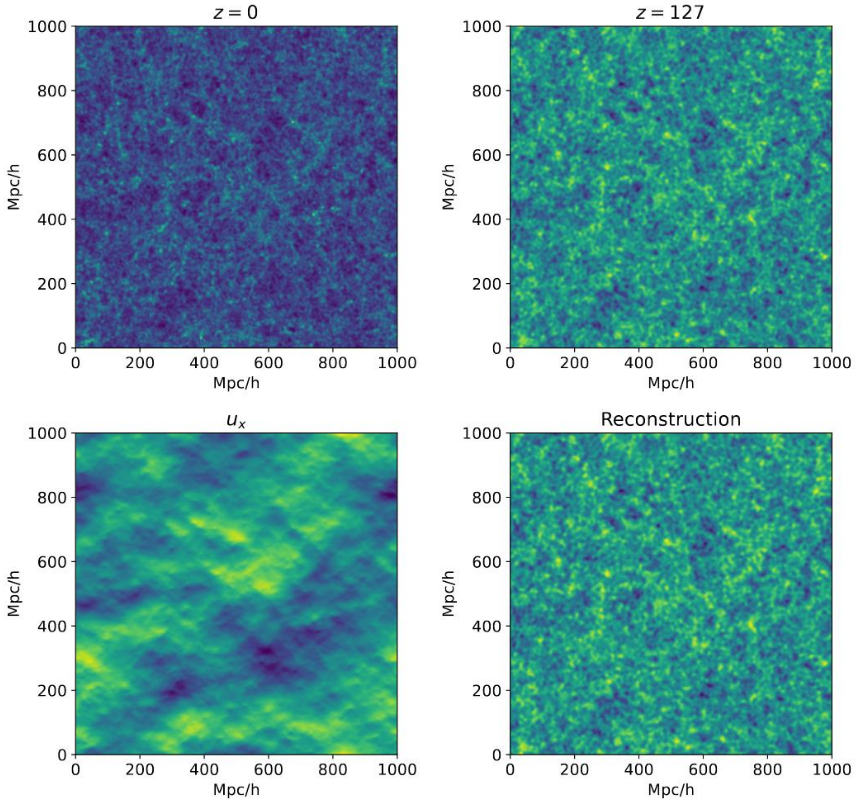
Different results from different groups!

ML approaches

- **Point estimates**: train a neural net to give single deterministic prediction (e.g. via MSE loss)



Jindal+, 2303.13056



Flöss+, 2305.07018

Diffusion models

- Train a network to approximate the **score:** $\nabla_x \log p(x|y)$
- Generate samples via reverse-diffusion process.

24 hrs of training on 4 80GB NVIDIA A100 GPU's

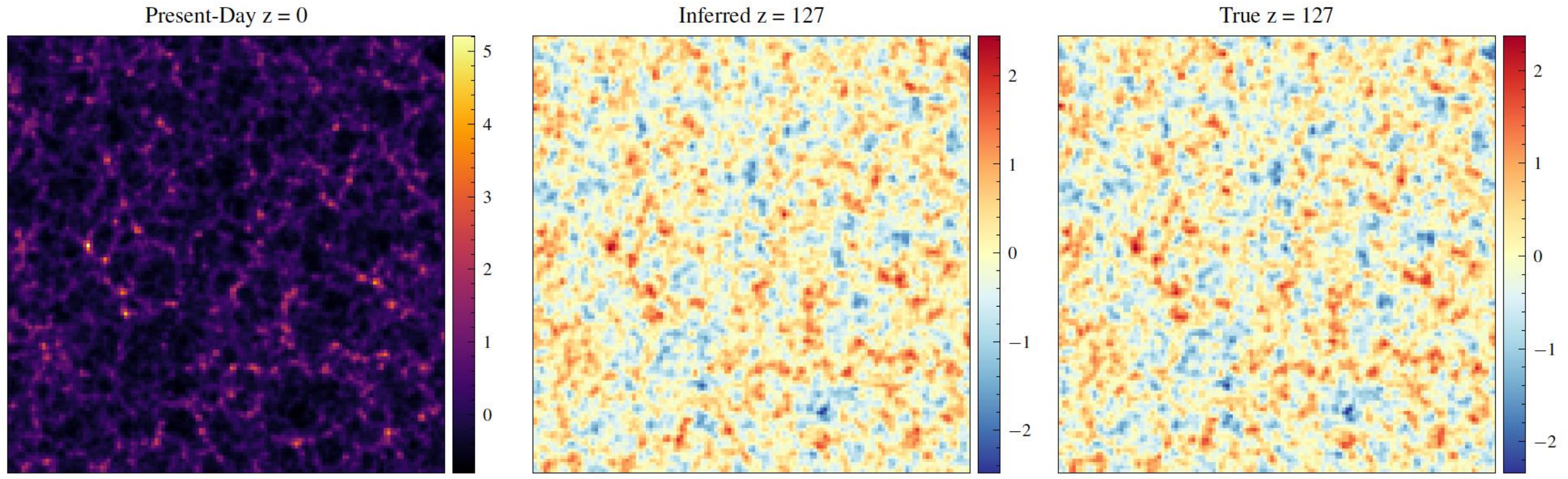
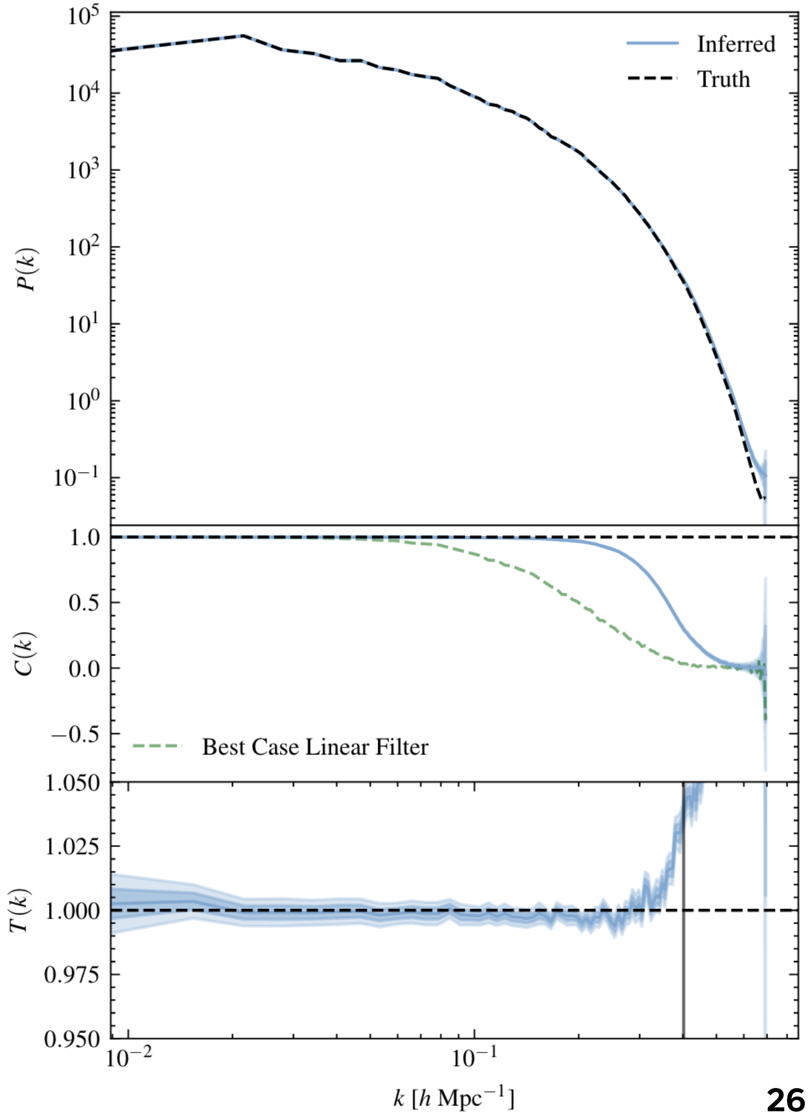


Figure 2. Left: The density field at redshift $z = 0$ for the fiducial Planck cosmology. Center: Initial conditions sampled from the posterior $p(\boldsymbol{x}|\boldsymbol{y})$. Right: The true initial conditions. All three density fields span a $1000 \times 1000 \times 125 (h^{-1}\mathrm{Mpc})^3$ region averaged over the third axis. This example demonstrates the capability of score-based generative models to sample highly detailed initial conditions consistent with the ground truth. See Figure 3 for quantification of uncertainty.

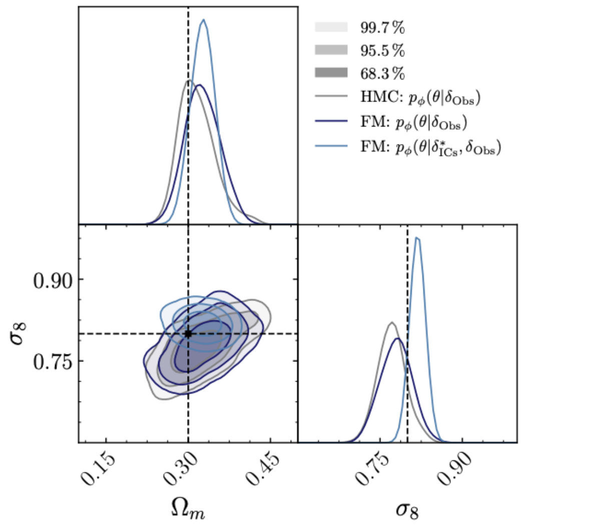
Legin+, 2304.03788



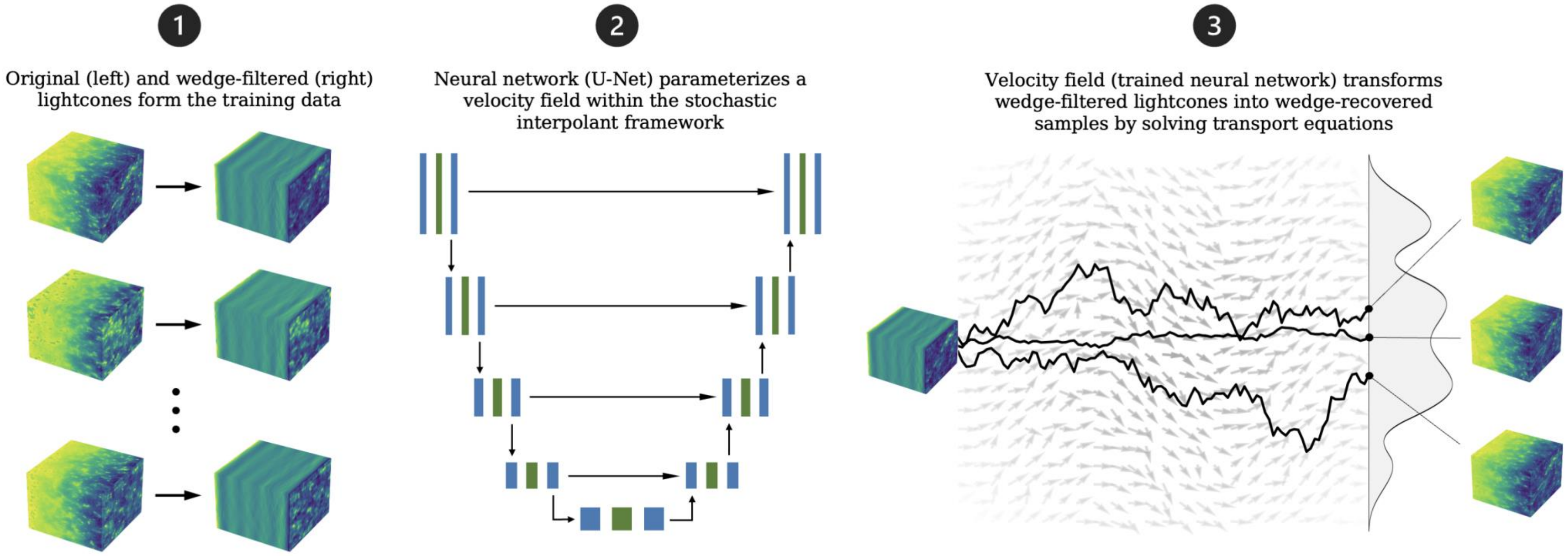
Flow-based models (stochastic interpolants)

- Train a network to approximate the **flow velocity field**
- Velocity field then stochastically transforms the samples

Cuesta-Lazaro+, **NeurIPS** 2024



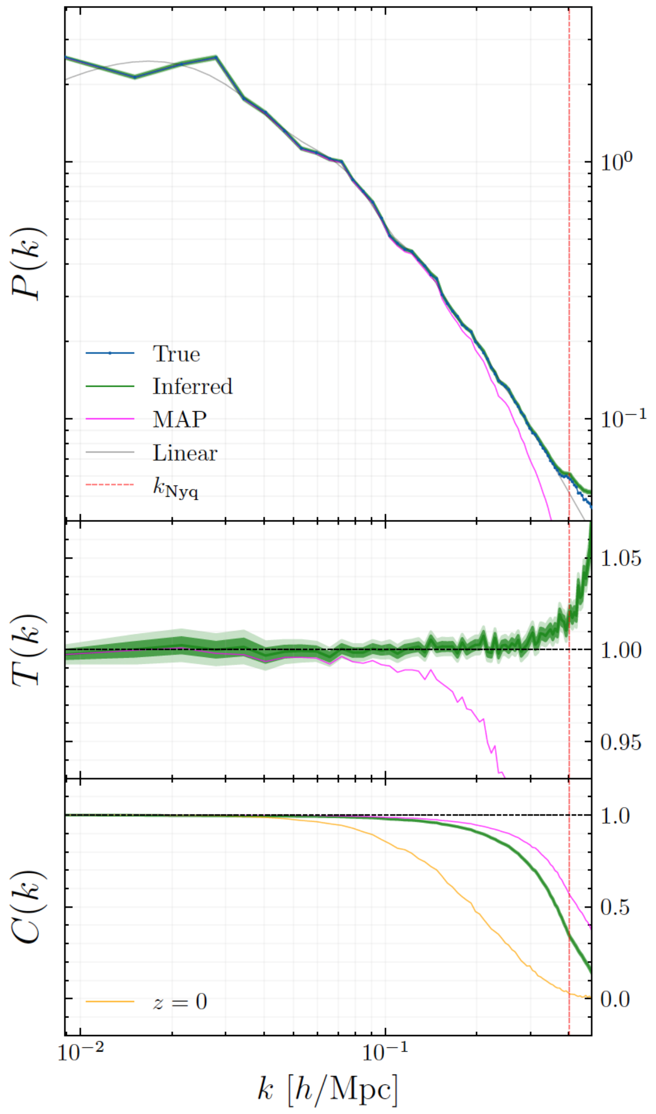
(a) Posteriors over cosmological parameters obtained through Stochastic Interpolants and HMC in a random test simulation. The true value is highlighted with dashed lines.



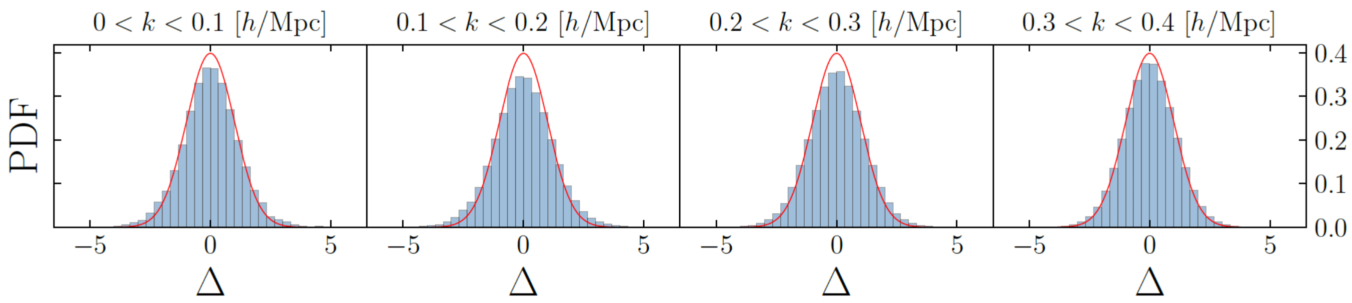
Sabti+, **2407.21097**

3 days of training on an 80GB NVIDIA A100 GPU, sampling in minutes

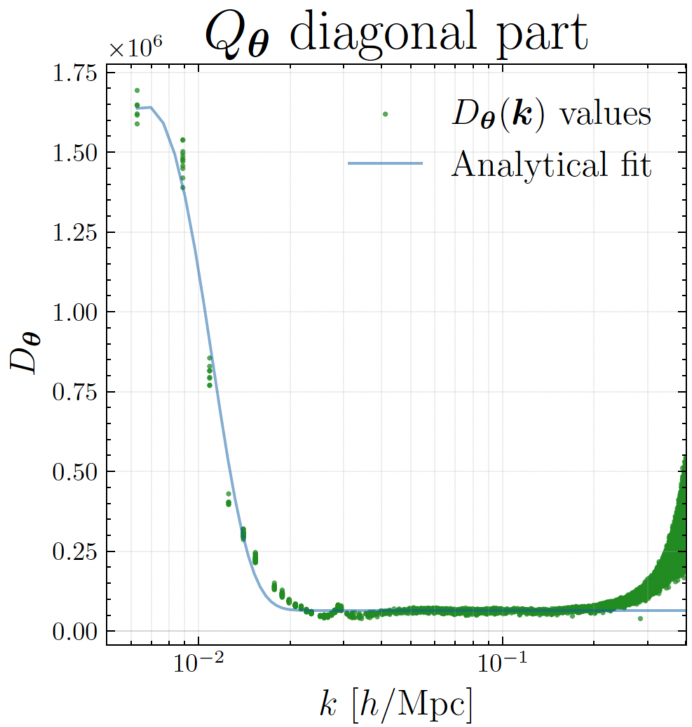
Summary statistics comparison



1-2% agreement in the power spectrum



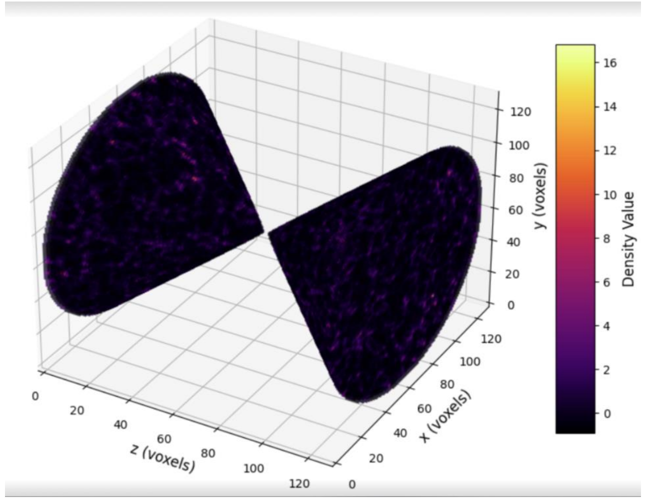
Coverage test shows that samples follow the correct distribution



Knowledge of the $\boldsymbol{Q}(|\boldsymbol{k}|)$ dependence allows to turn any point estimator into a fast sampler

[OS+, 2502.03139]

Incomplete data



- Modify $\mathbf{Q_L}$ matrix with real-space factors P:

$$\boldsymbol{Q}_{\boldsymbol{\theta}}^{L} = \mathcal{F}^{\dagger} \boldsymbol{D}_{\boldsymbol{\theta}}^{L} \mathcal{F} \quad \Rightarrow \quad \boldsymbol{Q}_{\boldsymbol{\theta}}^{L} = \mathcal{P} \mathcal{F}^{\dagger} \boldsymbol{D}_{\boldsymbol{\theta}}^{L} \mathcal{F} \mathcal{P}$$

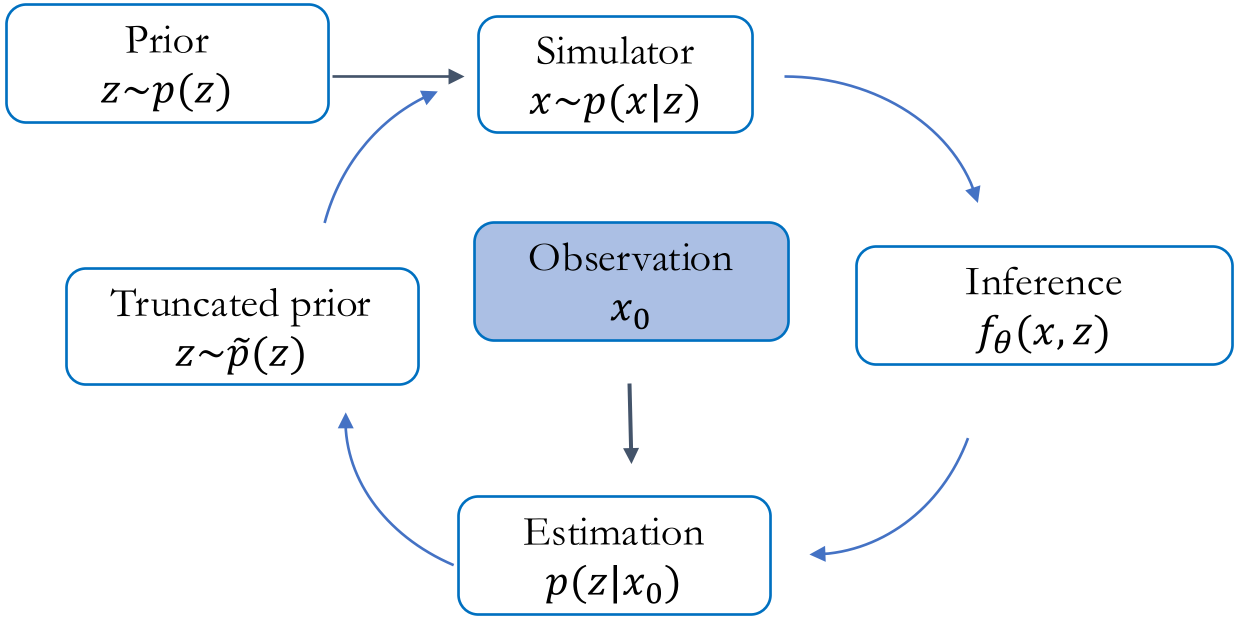
- Use Conjugate Gradient to estimate tr log $\boldsymbol{Q}$:

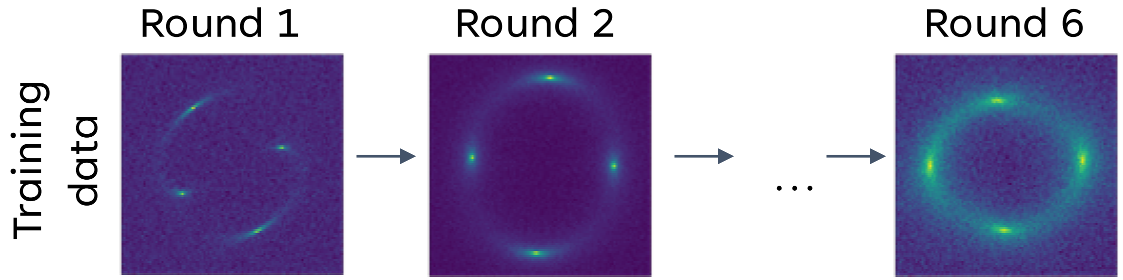
$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^{N} \left\{ (\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^T \boldsymbol{Q}_{\boldsymbol{\theta}} \left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \right) \right\} - \frac{N}{2} \operatorname{tr} \log \boldsymbol{Q}_{\boldsymbol{\theta}}$$

- Then sampling requires GEDA

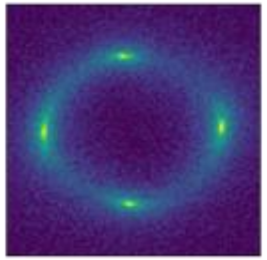
Sequential inference & adaptive learning

- Our parameter space is too vast to explore
- Want to ‘zoom in’ into it and obtain precise results with a low number of simulations





Target observation

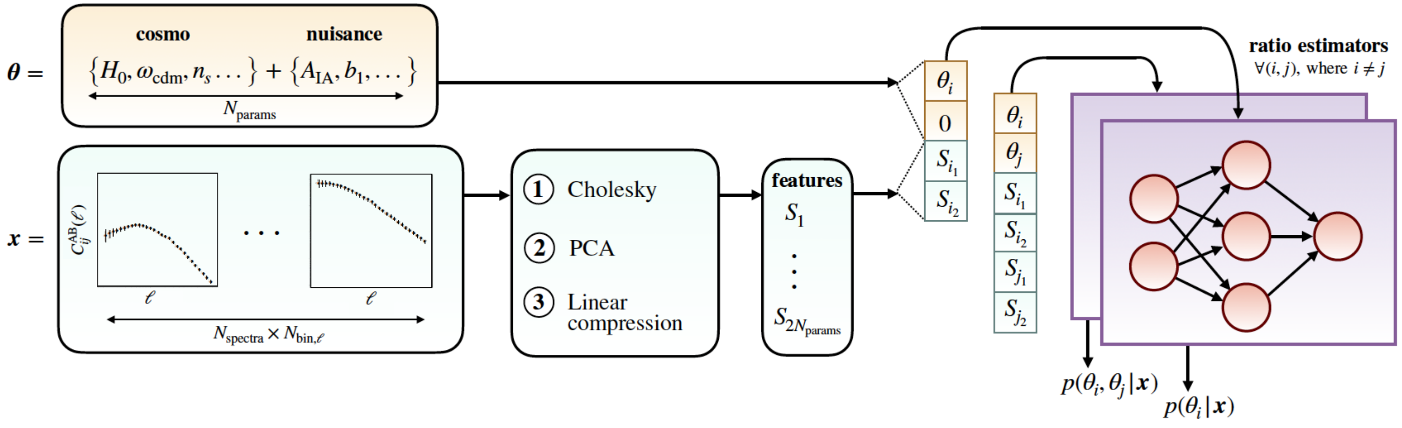


Learning with **high-precision**

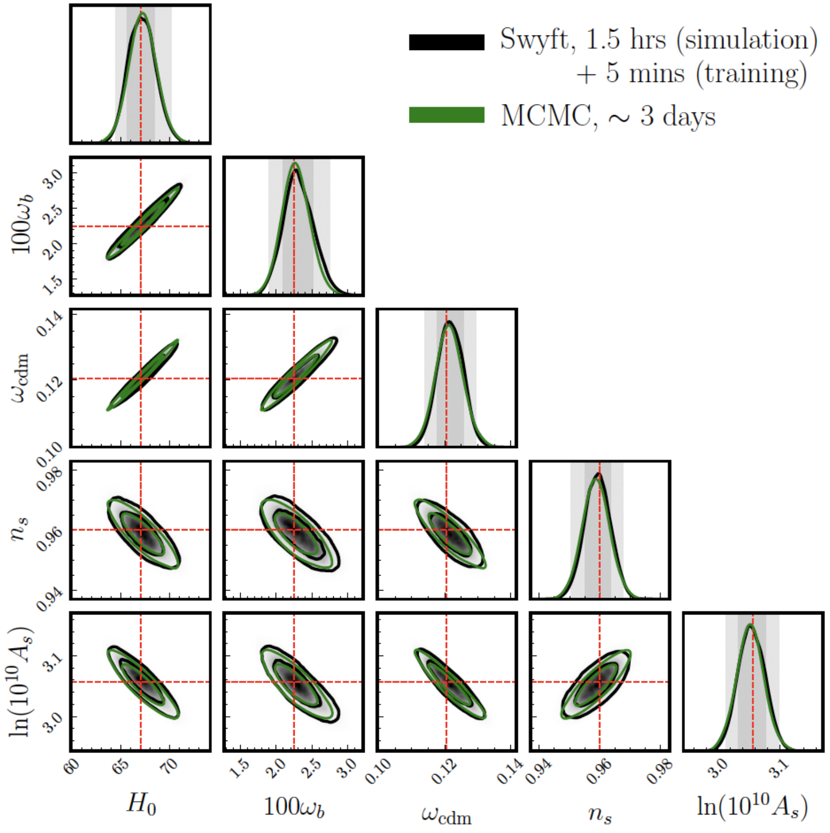
Figures credit:
N. Anau Montel

Parameter inference applied to Euclid

Apply **Marginal Neural Ratio Estimation** algorithm via **swyft** code to pre-compressed **3x2pt statistics**



G.F. Abellán+, 2403.14750



Excellent agreement with MCMC and a dramatic reduction in CPU time