Lossless, Scalable ILI for Cosmological Fields

https://arxiv.org/abs/2107.07405

T. Lucas Makinen, *Imperial College London + Sorbonne University*

With Tom Charnock (IAP), Justin Alsing (U Stockholm), and Ben Wandelt (IAP)

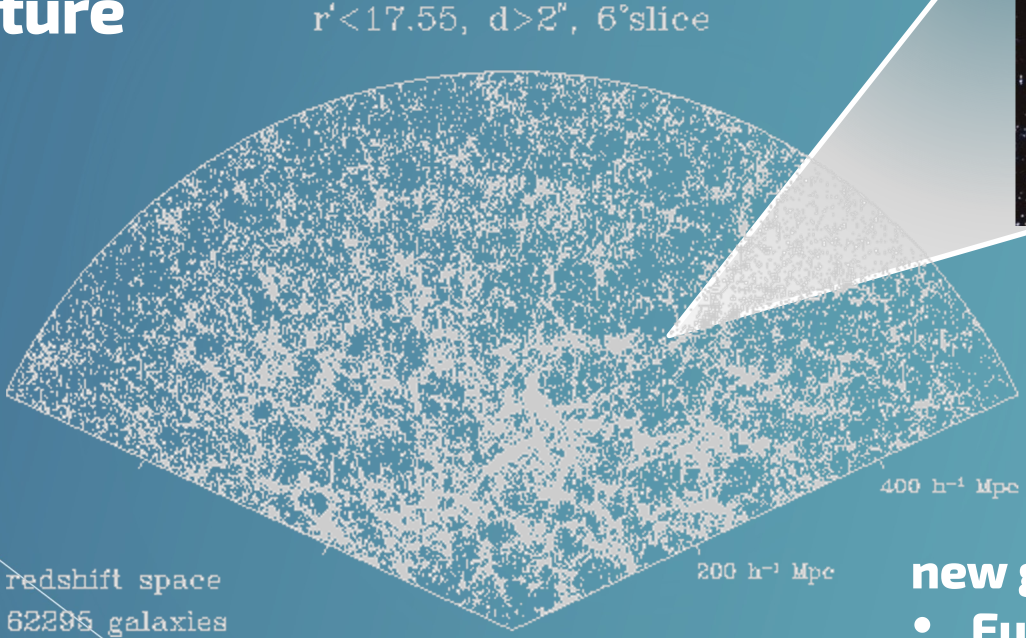
how to compress a universe into a few numbers (and what to do with them)

https://arxiv.org/abs/2107.07405

T. Lucas Makinen, *Imperial College London + Sorbonne University*

With Tom Charnock (IAP), Justin Alsing (U Stockholm), and Ben Wandelt (IAP)

Cosmological Inference from large-scale structure

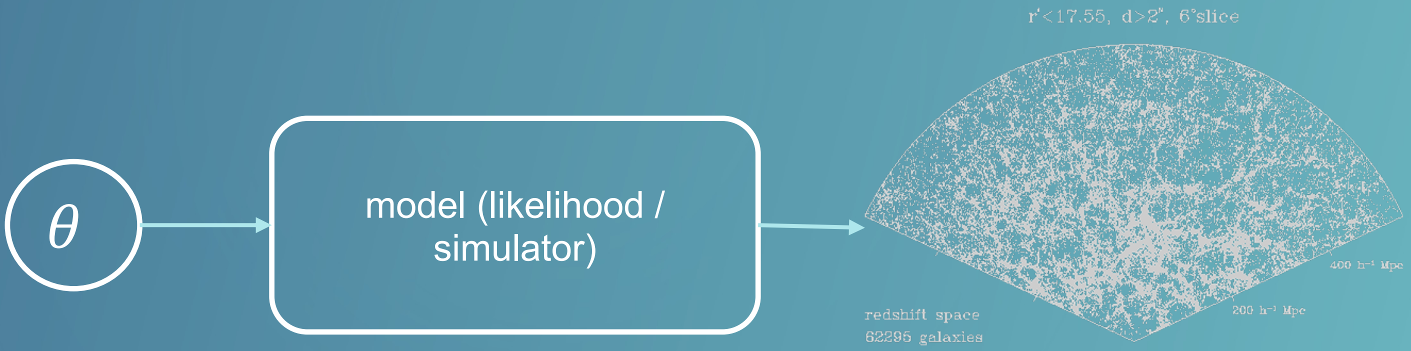


new galaxy / redshift surveys:

- Euclid + SKA ~1 billion galaxies each

Cosmological Inference from large-scale structure

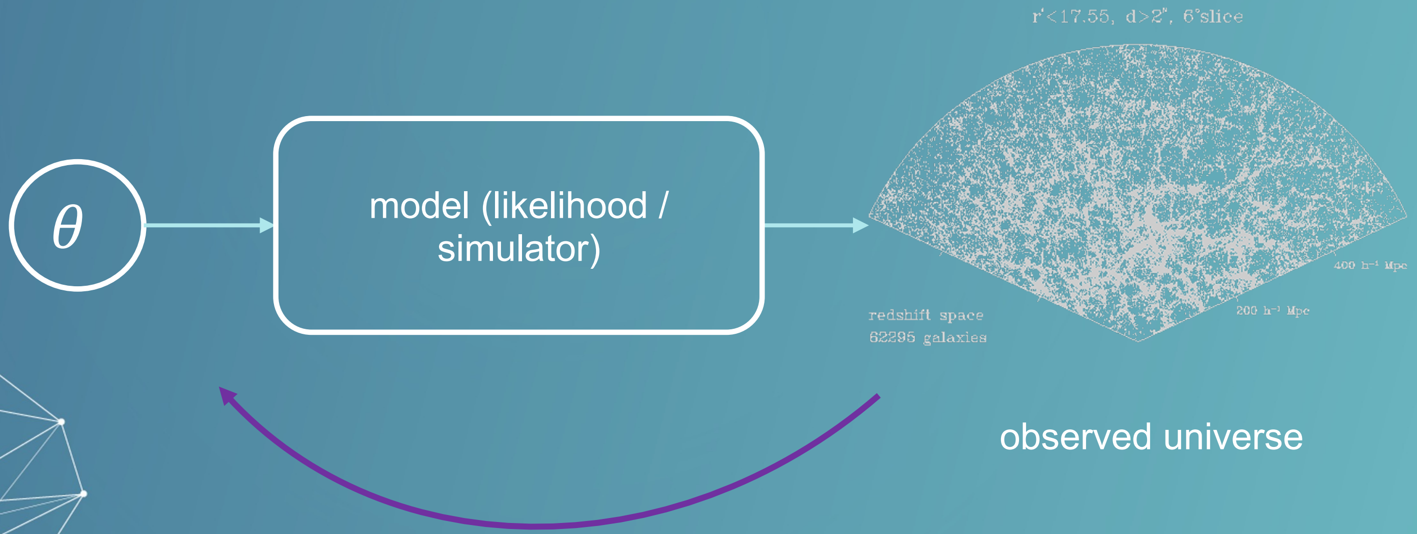
What is the probability of a given parameter, θ, being a good descriptor of observed large scale structure ?



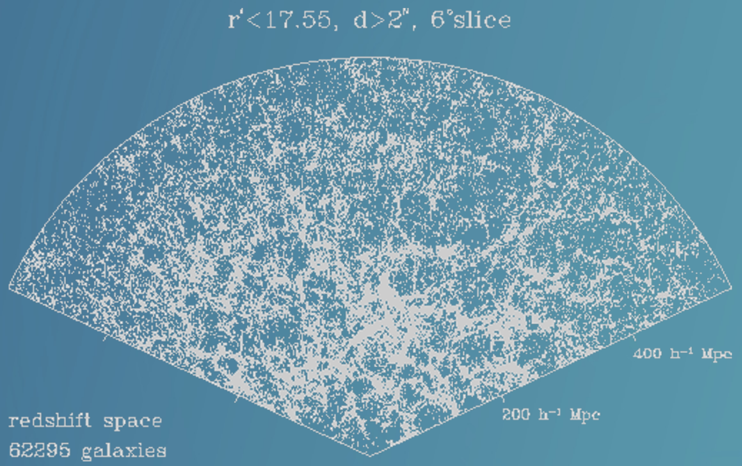
observed universe

Cosmological Inference from large-scale structure

Inverse problem: **What is the probability of a given parameter, θ, being a good descriptor of observed large scale structure ?**



Cosmological Inference from large-scale structure

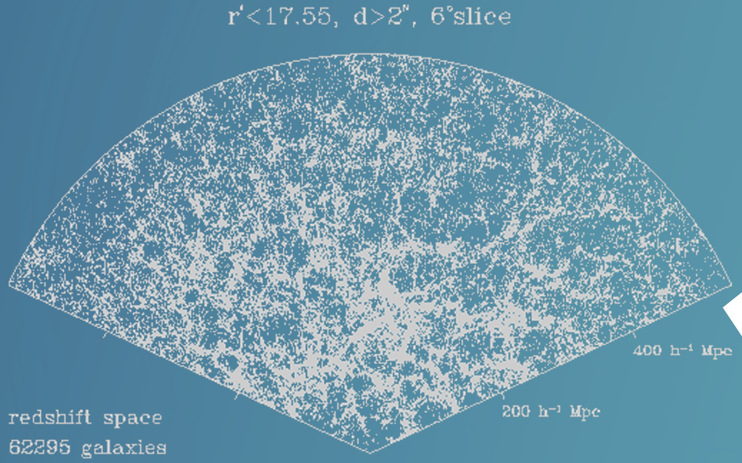


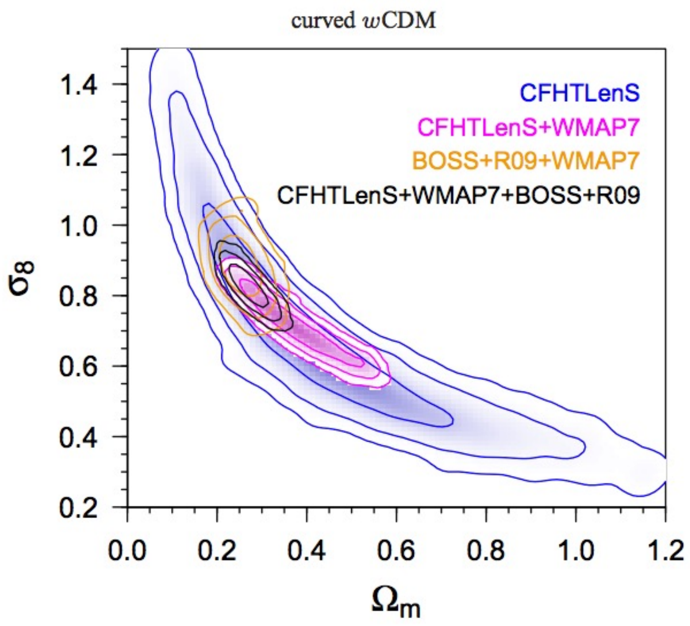
Questions:

1) Can we estimate distributions for cosmological parameters using the full overdensity field ?

2) Can we quantify the information content of the field ?

Cosmological Inference from large-scale structure ?





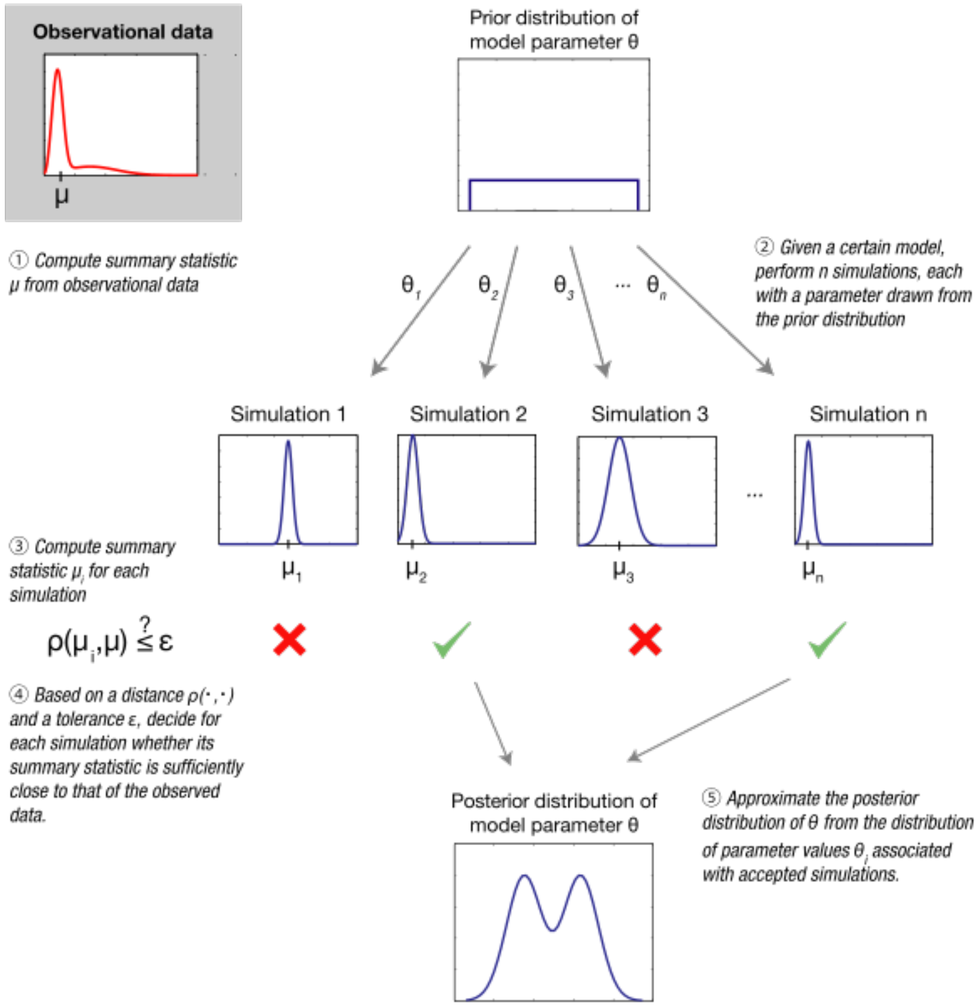
ILI: Implicit Likelihood Inference

$$p(\boldsymbol{\theta}|\mathbf{d}) \propto p(\mathbf{d}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

$\mathcal{L}(\mathbf{d}|\boldsymbol{\theta}) = p(\mathbf{d}|\boldsymbol{\theta})$: **likelihood** (simulator)

$p(\boldsymbol{\theta})$: prior

Pros	Cons
Can forward-simulate everything ! Universe + dust + telescope effects … No analytic description needed	Sims are huge ! How do we compare the distance from one simulation to a target observation ?



Approximate Bayesian Computation

1) Compress observed $\mathbf{d}$ to μ

2) For i simulation, compute distance

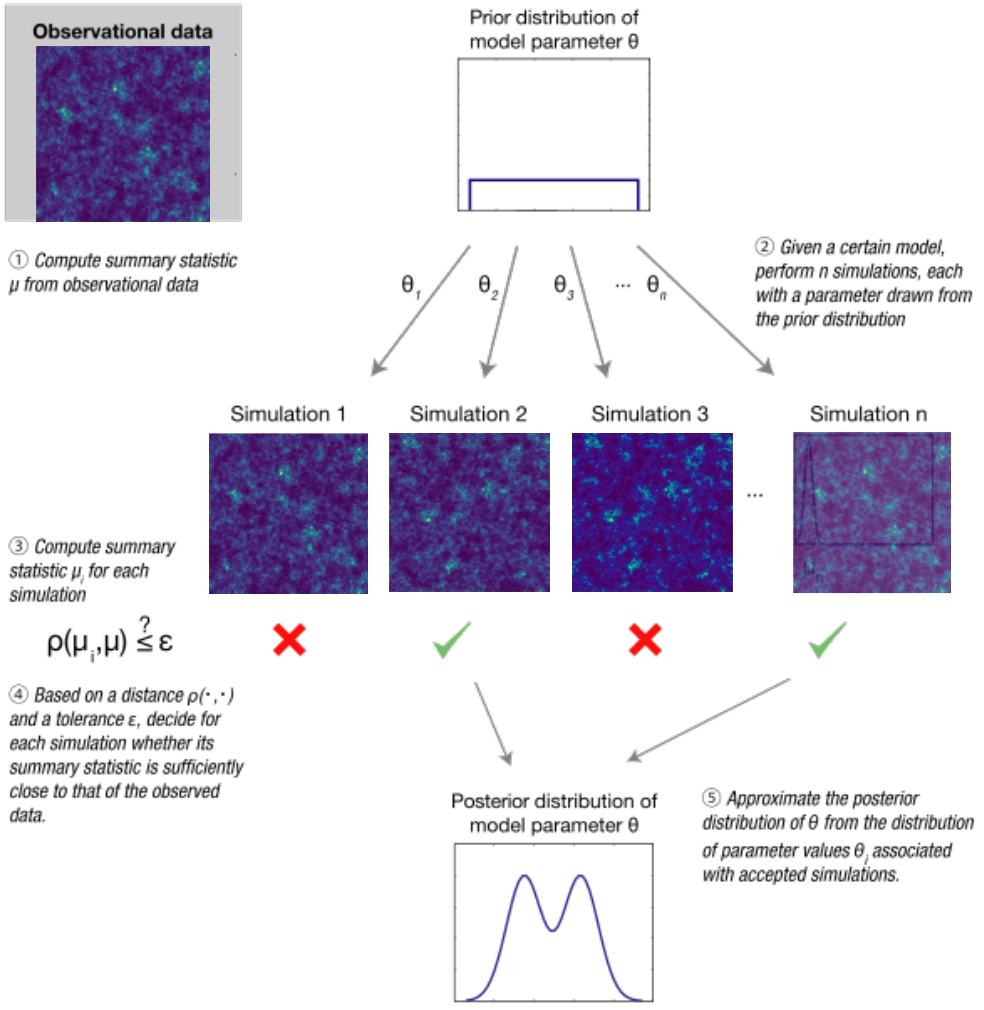
$$\rho(\mu, \hat{\mu})$$

if $\rho < \epsilon$, keep simulation.

PRO: can sample arbitrary distributions

CON: very expensive for large simulations and wide prior ranges

Adapted from https://en.wikipedia.org/wiki/Approximate_Bayesian_computation



Approximate Bayesian Computation

1) Compress observed $\mathbf{d}$ to μ

2) For *i* simulation, compute distance

$$\rho(\mu, \hat{\mu})$$

if $\rho < \epsilon$, keep simulation.

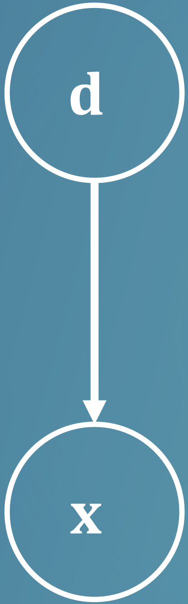
PRO: can sample arbitrary distributions

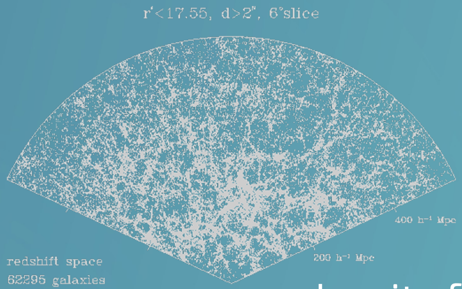
CON: very expensive for large simulations and wide prior ranges

HOW DO WE DEFINE OUR SUMMARY STATISTIC FOR LARGE SCALE STRUCTURE ?

Adapted from https://en.wikipedia.org/wiki/Approximate_Bayesian_computation

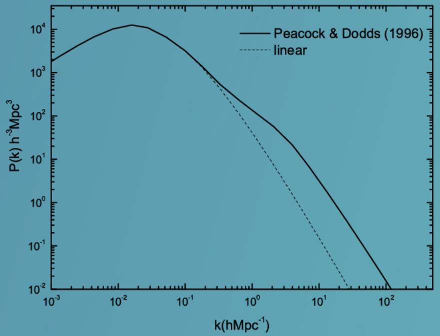
Large Scale Structure Compression ?





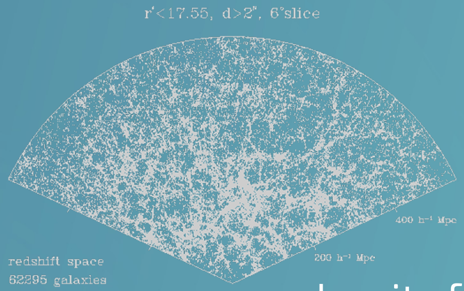
density field

+ noise
+ survey effects ?



power spectrum

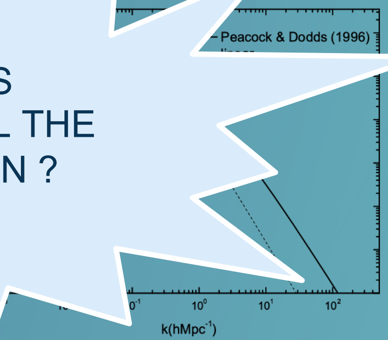
Large Scale Structure Compression ?



density field

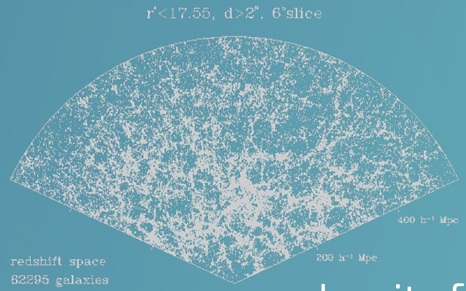
+ noise
+ survey effects ?

DOES THIS
CAPUTURE ALL THE
INFORMATION ?
(NO)



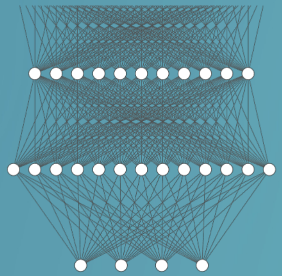
power spectrum

Large Scale Structure Compression ?



density field

+ noise
+ survey effects ?



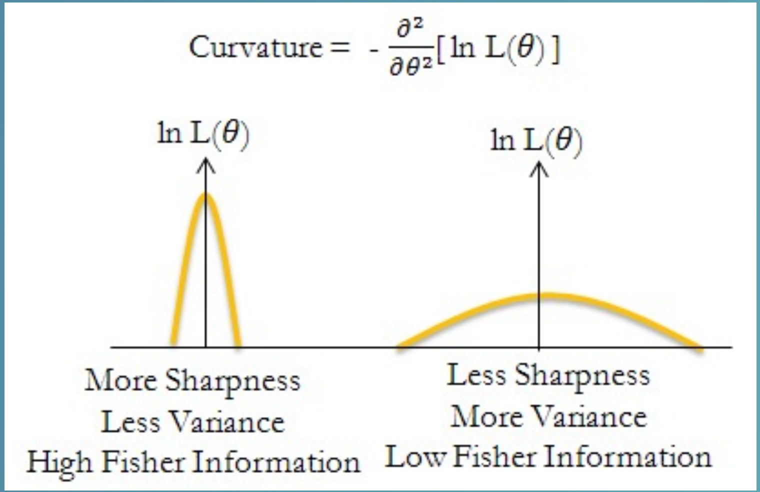
neural compression to summaries

Fisher information: tells us (on average) how informative some data $\mathbf{d}$ is about a parameter θ of a distribution, $\mathcal{L}(\mathbf{d}|\theta)$ that models $\mathbf{d}$

$$\mathbf{F}_{\alpha\beta} = -\left\langle \frac{\partial^2 \ln \mathcal{L}}{\partial\theta_\alpha \partial\theta_\beta} \right\rangle_{\theta=\theta_{\rm fid}}$$

Think of this as the *curvature* of the log-likelihood, $\ln \mathcal{L}$ at $\theta_{\rm fid}$

Fisher information: tells us (on average) how informative some data $\mathbf{d}$ is about a parameter θ of a distribution, $\mathcal{L}(\mathbf{d}|\theta)$ that models $\mathbf{d}$



Fisher information: tells us (on average) how informative some data $\mathbf{d}$ is about a parameter θ of a distribution, $\mathcal{L}(\mathbf{d}|\theta)$ that models $\mathbf{d}$

Example: draw $\mathrm{n}_{\mathbf{d}}$ independent datapoints from a normal distribution, $\mathcal{N}(\mu, \sigma)$. Then the likelihood is:

$$\mathcal{L}(\mathbf{d}|\mu,\sigma) = \prod_{i=1}^{n_d} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\frac{(d_i-\mu)^2}{\sigma^2}\right)$$

And the Fisher matrix is:

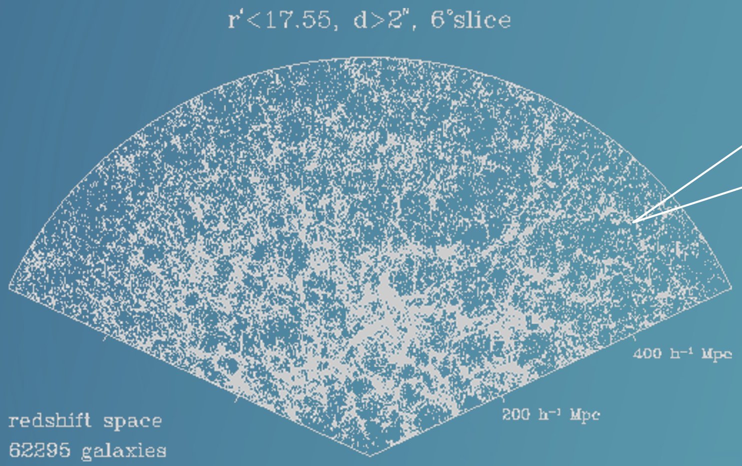
$$F = -\left\langle \frac{\partial^2 \ln \mathcal{L}}{\partial\theta_\alpha \partial\theta_\beta} \right\rangle_{\theta_{\mathrm{fid}}} = \begin{pmatrix} \frac{-n_d}{\sigma} & 0 \\ 0 & \frac{-n_d}{2\sigma^2} \end{pmatrix}_{\sigma_{\mathrm{fid}}}$$

Fisher information: tells us (on average) how informative some data $\mathbf{d}$ is about a parameter θ of a distribution, $\mathcal{L}(\mathbf{d}|\theta)$ that models $\mathbf{d}$

Cramer-Rao bound:

$$\langle(\theta_\alpha - \langle\theta_\alpha\rangle)(\theta_\beta - \langle\theta_\beta\rangle)\rangle \geq \mathbf{F}^{-1}_{\alpha\beta}$$

Gives us a lower bound for the (average) variance of a parameter estimate



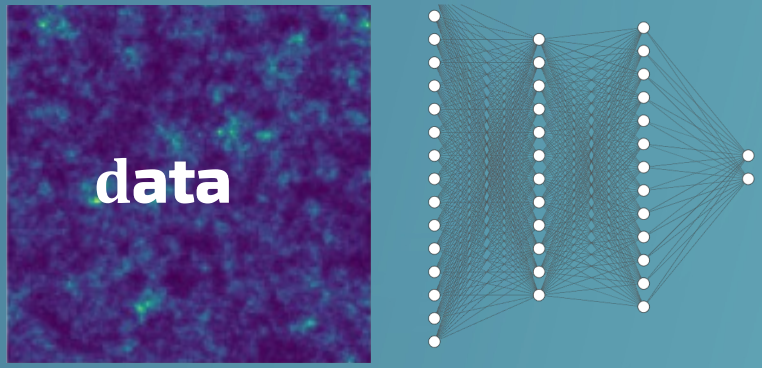
What if we could compress the universe down to a handful of numbers *with the same information content as the full field ?*

Information Maximising Neural Networks

Can we train a neural network to compress a universe simulation down to a couple of numbers ?

$$f : \mathbf{d} \mapsto \mathbf{x}$$

$\boldsymbol{\theta} \in \mathbb{R}^2$ $\longrightarrow$



summaries

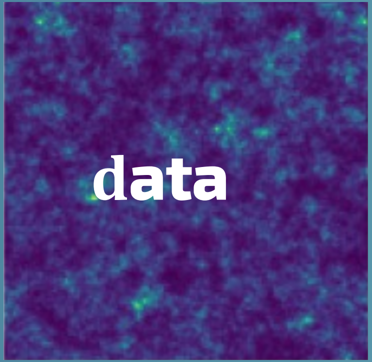
$\mathbf{d} \in \mathbb{R}^{N_{\mathbf{pix}}}$ $\mapsto$ $\mathbf{x} \in \mathbb{R}^2$

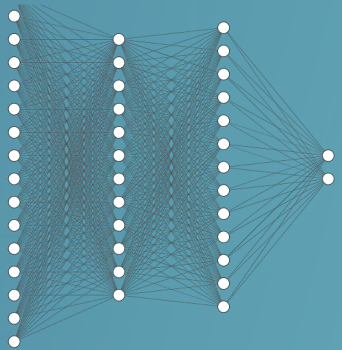
Information Maximising Neural Networks

Can we train a neural network to compress a universe simulation down to a couple of numbers ?

$$f : \mathbf{d} \mapsto \mathbf{x}$$

$\boldsymbol{\theta} \in \mathbb{R}^2 \longrightarrow$





summaries

$\mathbf{d} \in \mathbb{R}^{N_{\text{pix}}} \quad \mapsto \quad \mathbf{x} \in \mathbb{R}^2$

See Alsing & Wandelt (2018) arXiv:1712.00012 for why summary space is taken to be the same dimension as parameter space

Information Maximising Neural Networks

1) adopt a Gaussian likelihood form to compute our Fisher information:

$$-2\ln\mathcal{L}(\mathbf{x}|\mathbf{d}) = \left(\mathbf{x} - \boldsymbol{\mu}_f(\boldsymbol{\theta})\right)^T \boldsymbol{C}_f^{-1}(\mathbf{x} - \boldsymbol{\mu}_f(\boldsymbol{\theta}))$$

Mean and covariance of network outputs

Charnock et al (2018) arXiv:1802.03537

Information Maximising Neural Networks

1) adopt a Gaussian likelihood form to compute our Fisher information:

$$-2\ln\mathcal{L}(\mathbf{x}|\mathbf{d}) = \left(\mathbf{x} - \boldsymbol{\mu}_f(\boldsymbol{\theta})\right)^T \boldsymbol{C}_f^{-1}(\mathbf{x} - \boldsymbol{\mu}_f(\boldsymbol{\theta}))$$

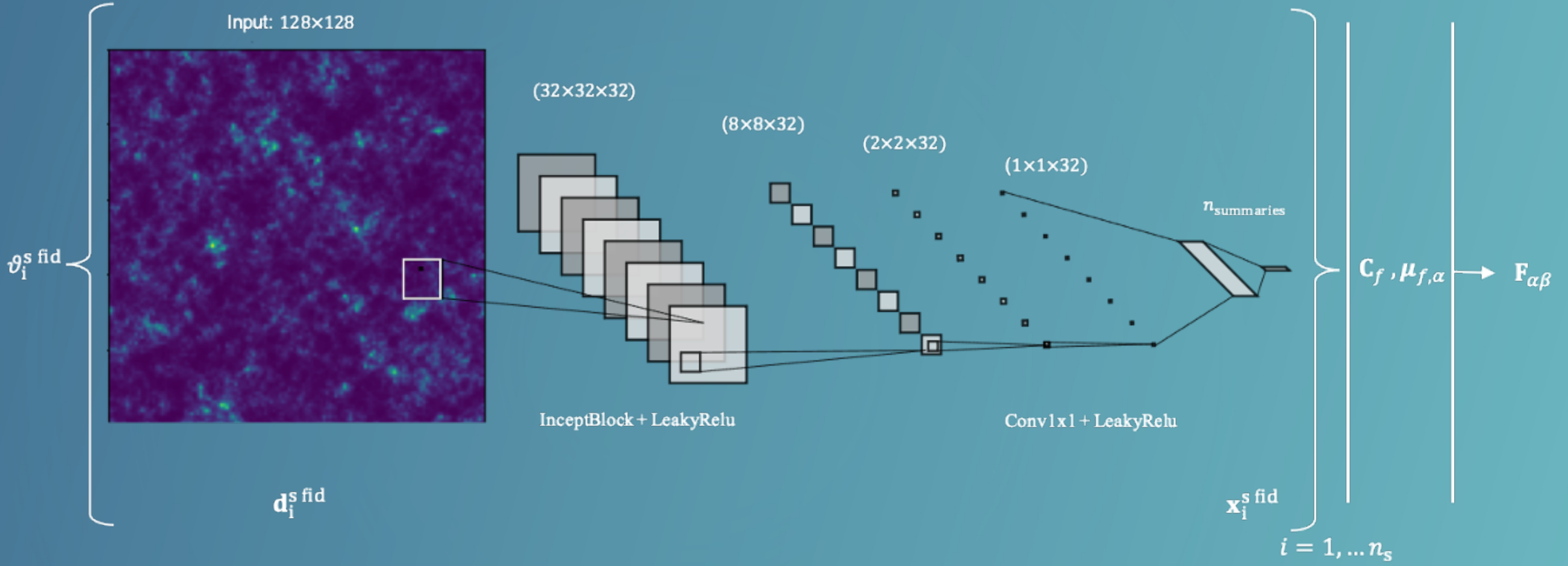
2) Compute IMNN Fisher:

$$\mathrm{F}_{\alpha\beta} = \mathrm{tr}[\boldsymbol{\mu}_{f,\alpha}^T C_f^{-1} \boldsymbol{\mu}_{f,\beta}]$$

3) train until Fisher information is maximised *at a fiducial model*

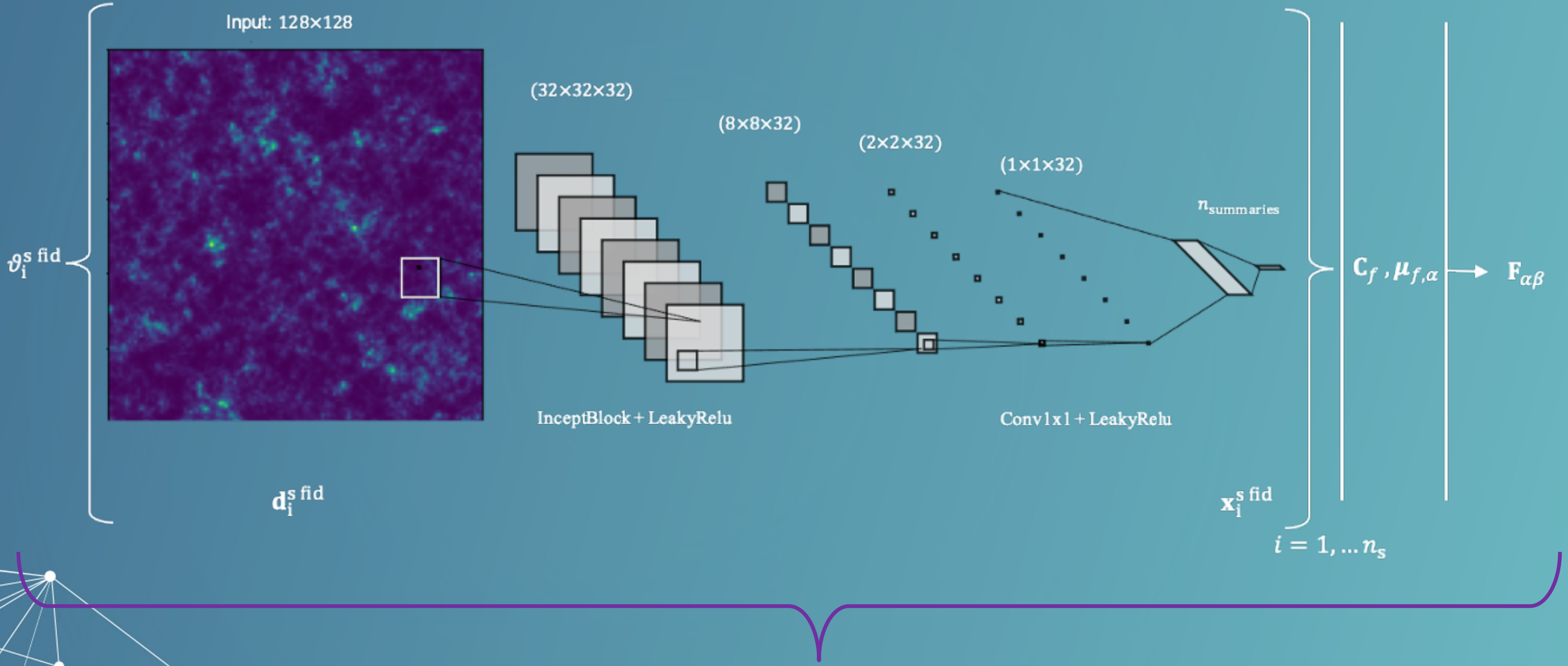
Charnock et al (2018) arXiv:1802.03537

Main IMNN Scheme



Makinen et al (2021) arXiv:2107.07405

Main IMNN Scheme



Completely differentiable in Jax !

Makinen et al (2021) arXiv:2107.07405

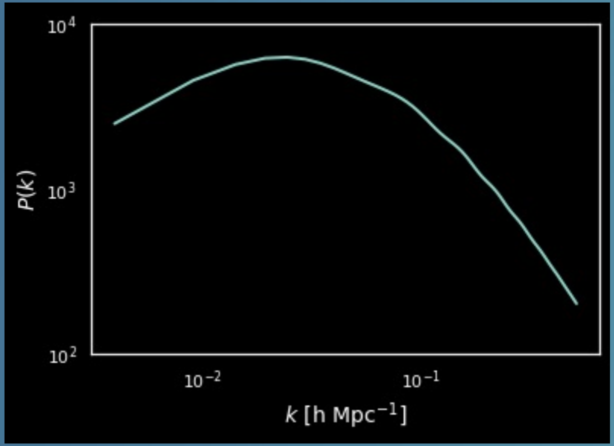
Inference for Mock Dark Matter fields

1. Train IMNN compression on simulations of a fiducial universe with parameters θ_{fid}

2. Observe + compress observed universe to get estimates for θ_{target}

3. Using compression, simulate universes over prior distribution $p(\boldsymbol{\theta})$ to obtain posterior $p(\boldsymbol{\theta} \mid \mathbf{d})$

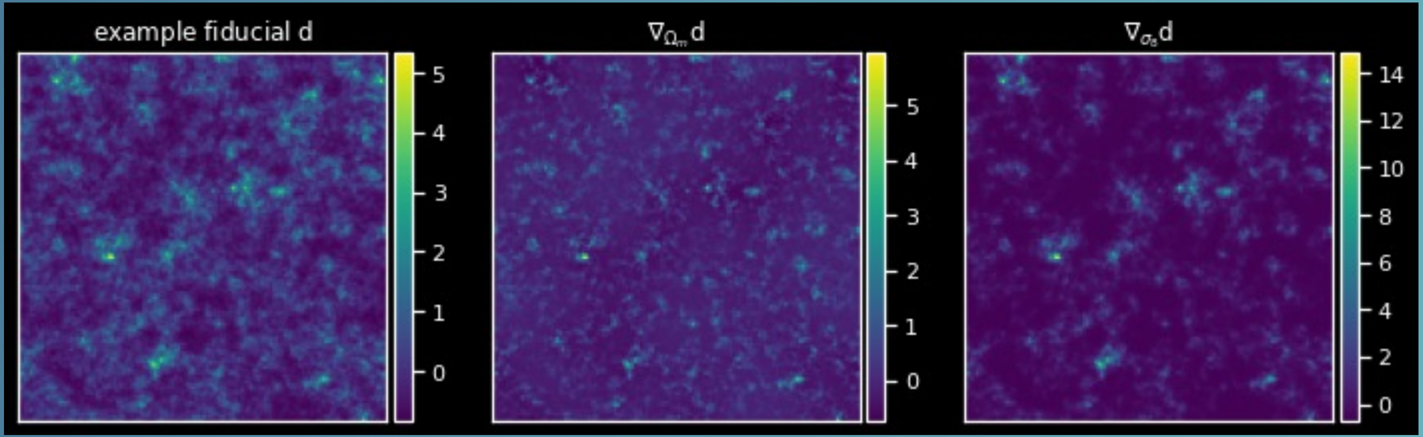
Inference for Mock Dark Matter fields



Train compression on 128x128 fiducial lognormal field simulations generated from Eisenstein-Hu P(k) , with fiducial parameters:

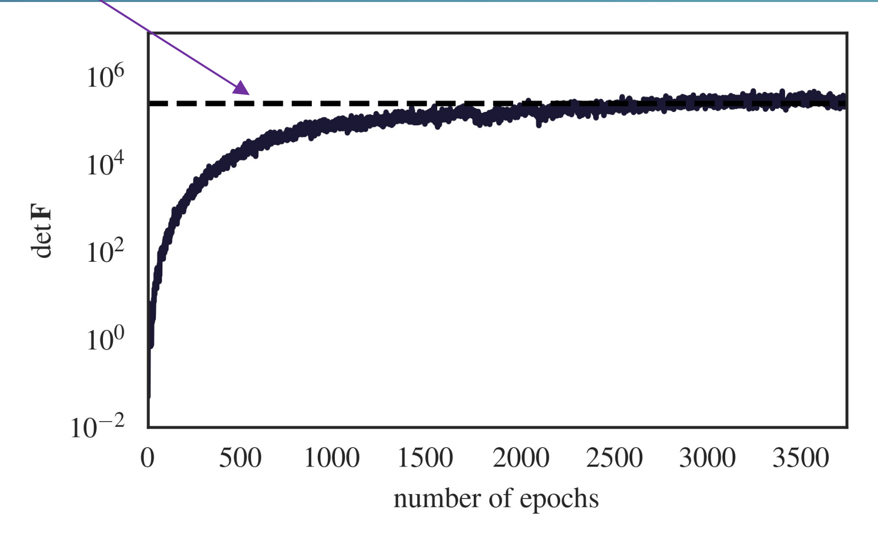
$$\theta_{\rm fid} = (\Omega_c, \sigma_8) = (0.6, 0.6)$$

Simulate a differentiable universe !

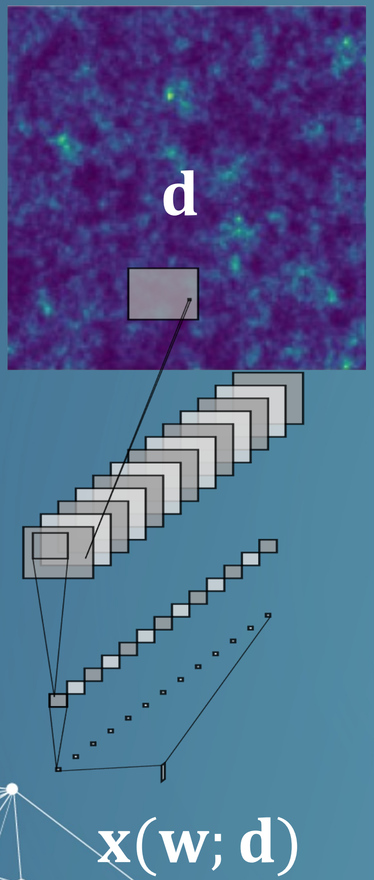


IMNN training: saturate known information content

(known) theoretical field information content (all pixels) !



Makinen et al (2021) arXiv:2107.07405



Compress observed universe

Next: observe universe + compress

Make *score estimates* of the parameters:

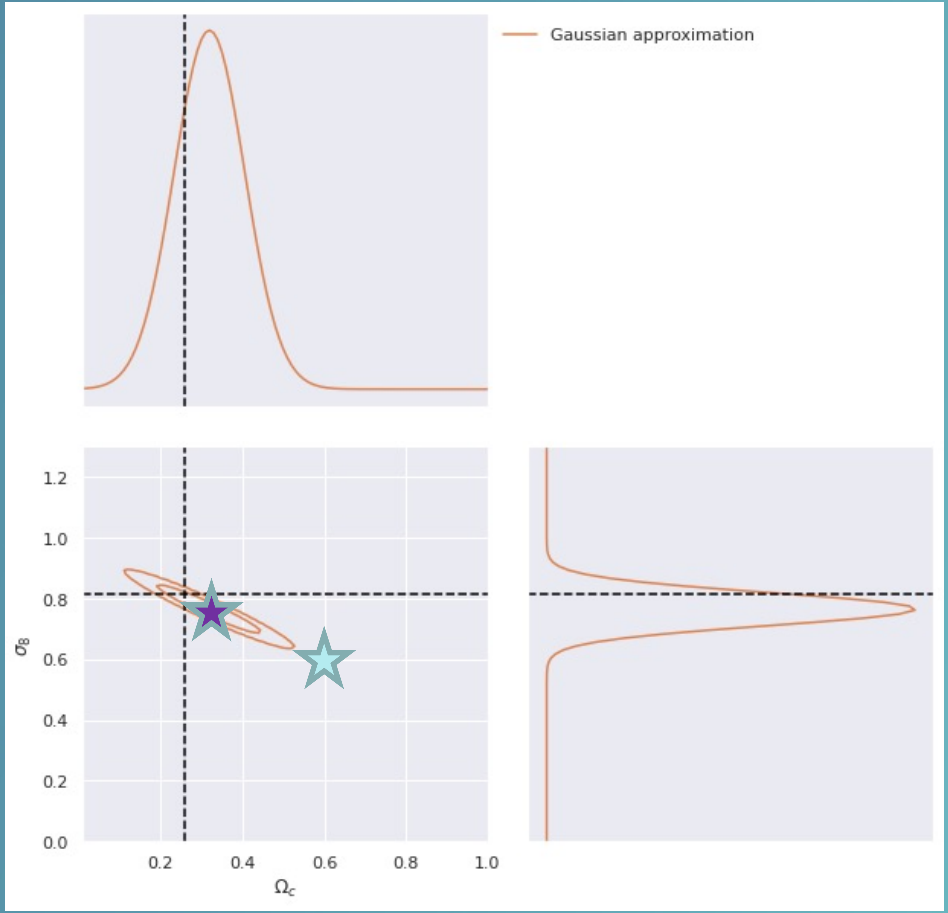
$$\hat{\theta}_\alpha = \theta_\alpha^{\mathrm{fid}} + \mathrm{F}^{-1}_{\alpha\beta}\,\frac{\partial \mu_i}{\partial \theta_\beta}\,\mathbf{C}^{-1}_{ij}\left(\mathrm{x}(\mathbf{w};\mathbf{d}) - \mu\right)_j$$

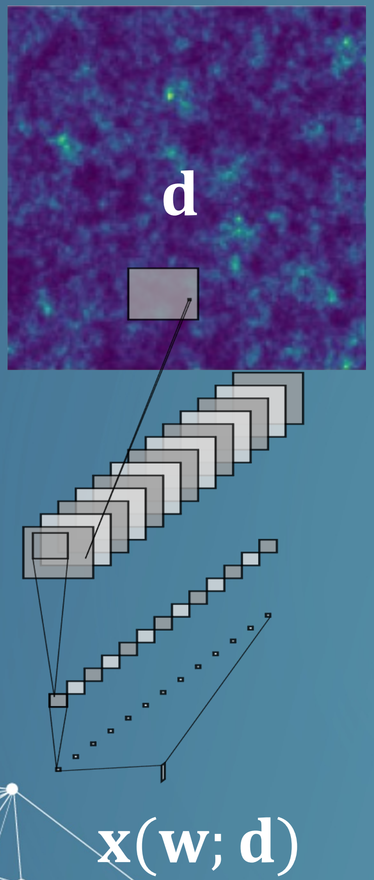
Score Estimates + Fisher Contours

Fiducial (poor)

Score estimates

Fisher Gaussian Approximtion





Step 2: Make estimates + Re-train

retrain compression on IMNN score estimates: new fiducial model parameters

$$\theta_{\rm fid,2} = (\Omega_c, \sigma_8) = (0.28, 0.73)$$

Step 3: Neural Density Estimation

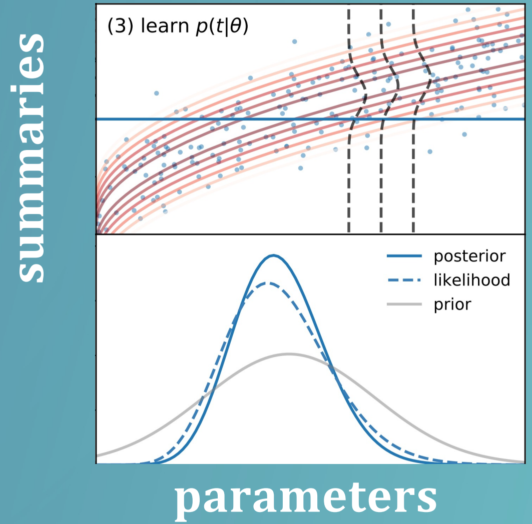
Goal: parameterize the posterior $p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ for Ω_c, σ_8 with compressed simulations

Q: How do we parameterize $p(\mathbf{x}|\boldsymbol{\theta})$ whilst *minimizing* the number of simulations needed ?

Neural Density Estimation

Goal: parameterize the posterior $p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$

A: Using Conditional Masked Autoregressive Flows



Alsing et al (2018): https://arxiv.org/abs/1903.00007

Conditional Masked Autoregressive Flows

Parameterize the *summary data likelihood* with a neural network with weights $\mathbf{w}$:

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{i=1}^{\dim(\mathbf{x})} p\left(\mathrm{x}_i \mid \mathrm{x}_{1:i-1}, \boldsymbol{\theta}; \mathbf{w}\right)$$

Conditional Masked Autoregressive Flows

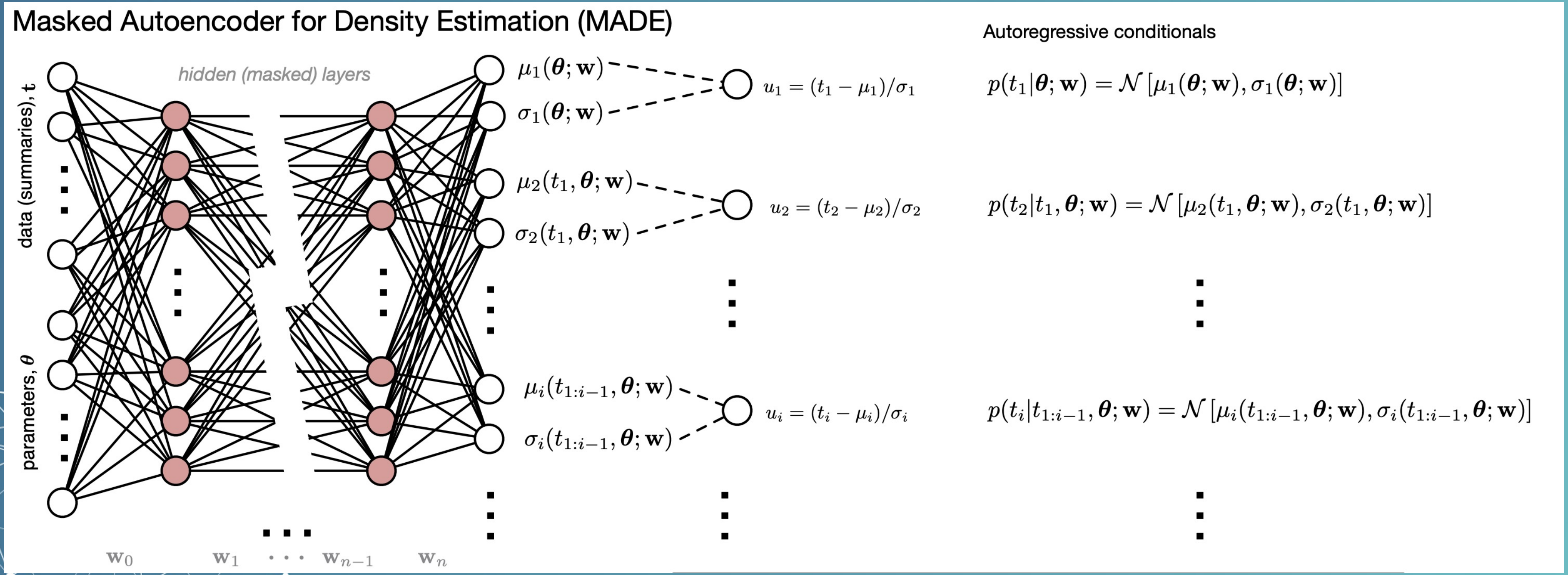
Parameterize the *summary data likelihood* with a neural network with weights $\mathbf{w}$:

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{i=1}^{\dim(\mathbf{x})} p\left(\mathrm{x}_i \mid \mathrm{x}_{1:i-1}, \boldsymbol{\theta}; \mathbf{w}\right)$$

Minimize log-loss: -ln U = $-\sum_i \ln p\left(\mathrm{x}_i \mid \mathrm{x}_{1:i-1}, \boldsymbol{\theta}; \mathbf{w}\right)$

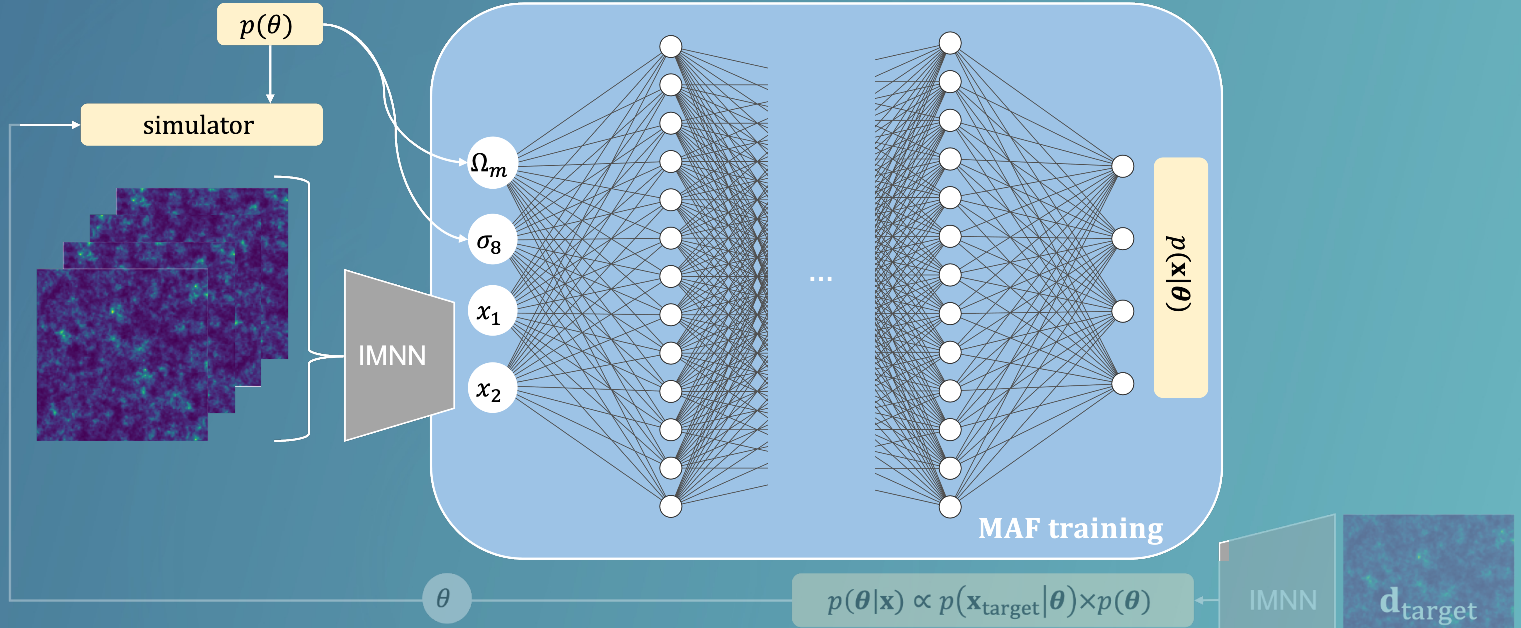
Conditional Masked Autoregressive Flows

Minimize log-loss: $-\ln \mathrm{U} = \sum_i \ln p\,(\,\mathrm{x\_}i \mid \mathrm{x}_{1:i-1}, \boldsymbol{\theta}; \mathbf{w})$

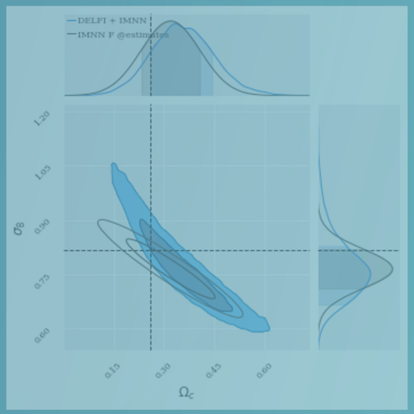


Alsing et al (2018): https://arxiv.org/abs/1903.00007

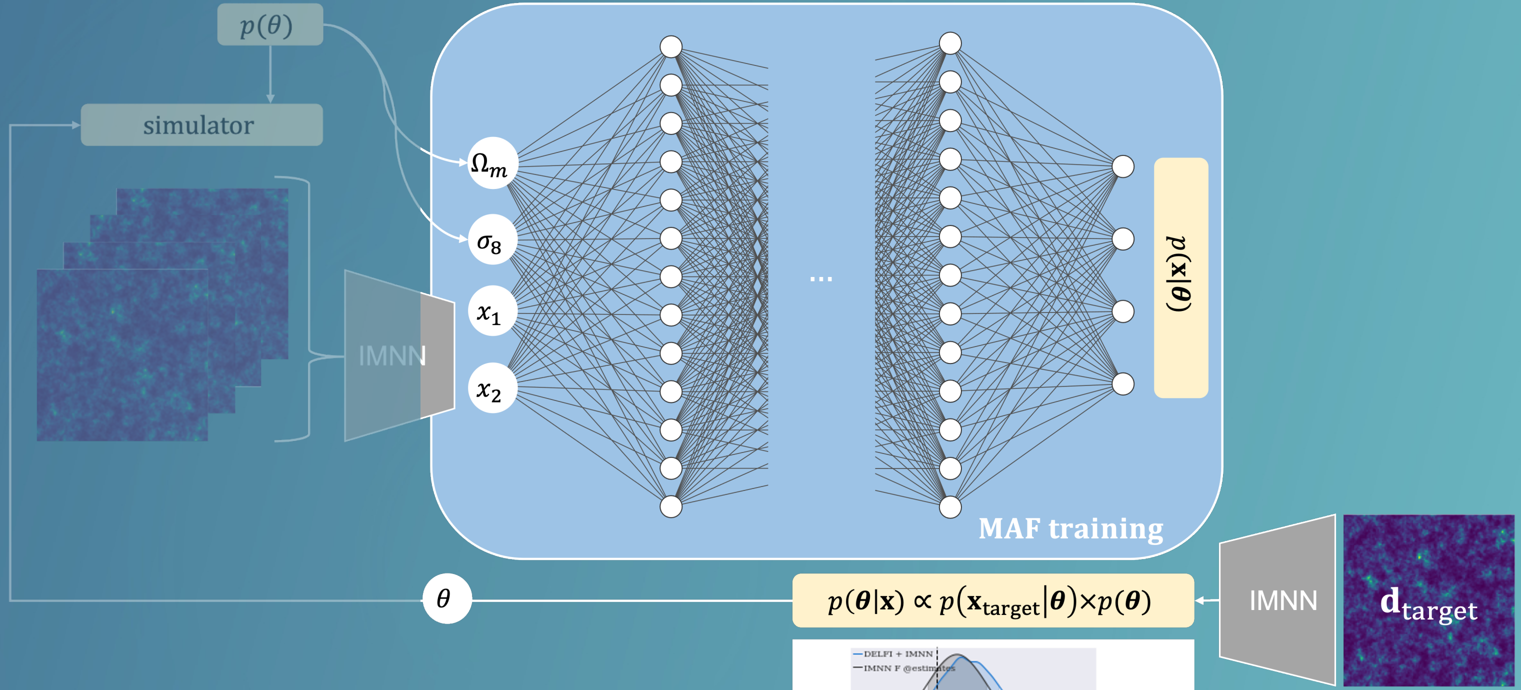
Neural Density Estimation



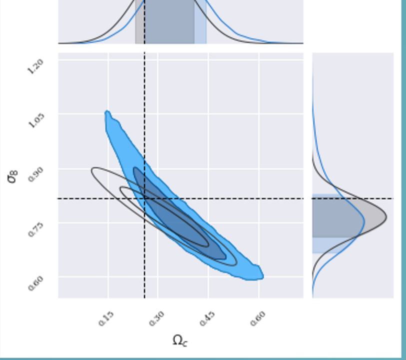
1. Draw from prior
2. Compress simulations
3. Train CMAF



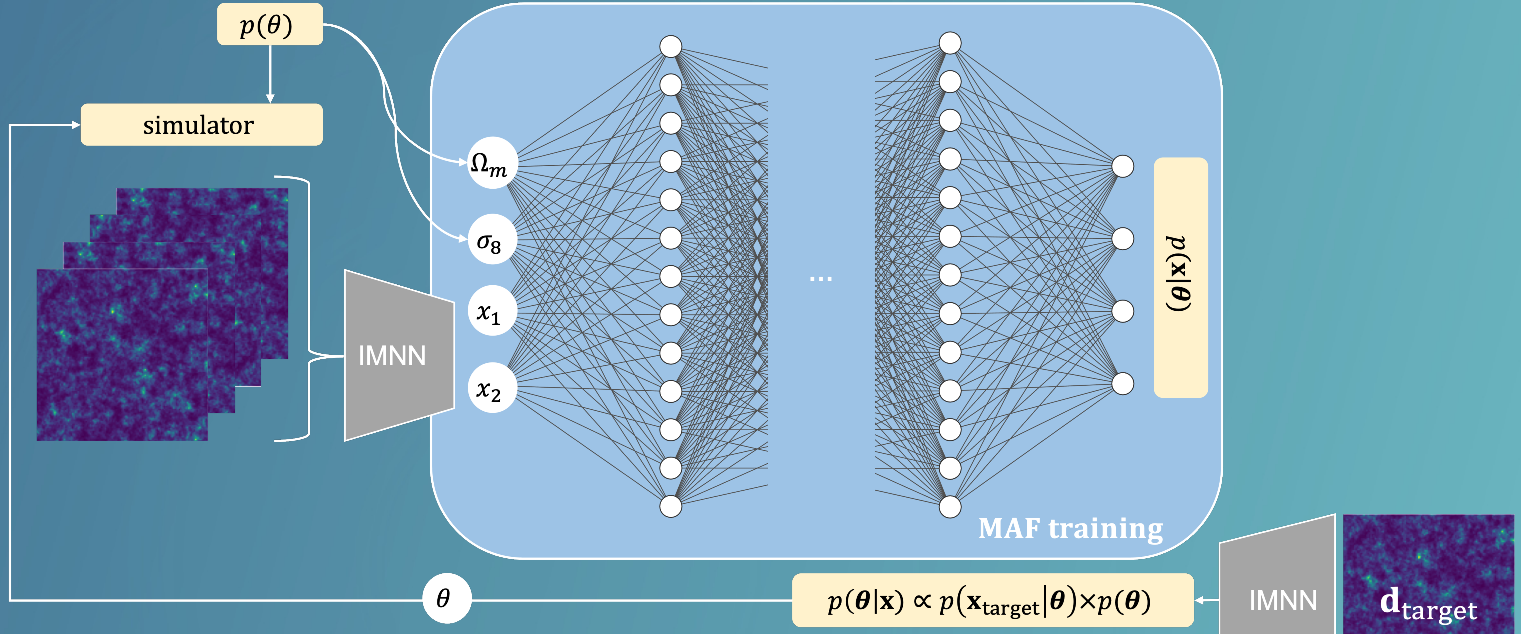
Neural Density Estimation

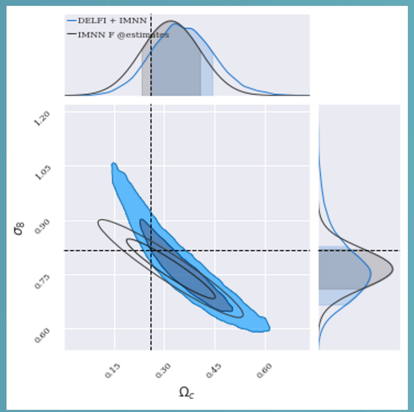


5. Compress target data
6. Evaluate posterior



Neural Density Estimation





7. Draw parameters from posterior, simulate, compress
8. Append new (θ, x) to training data & continue training

Final inference

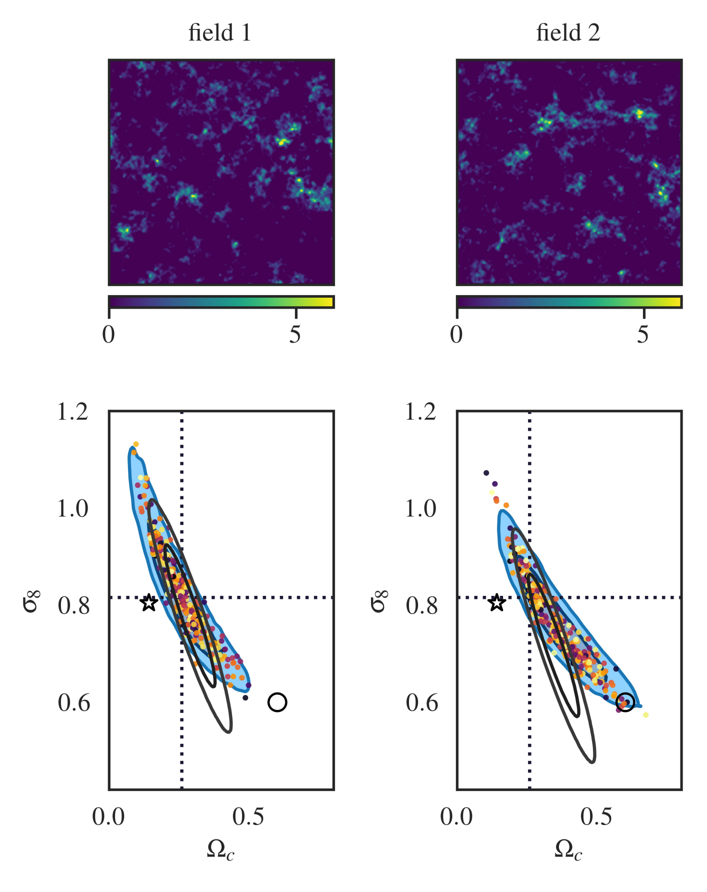
Fiducial (poor)

Score estimates

Fisher Gaussian Approximation

Approximate Bayesian Computation

DELFI + IMNN

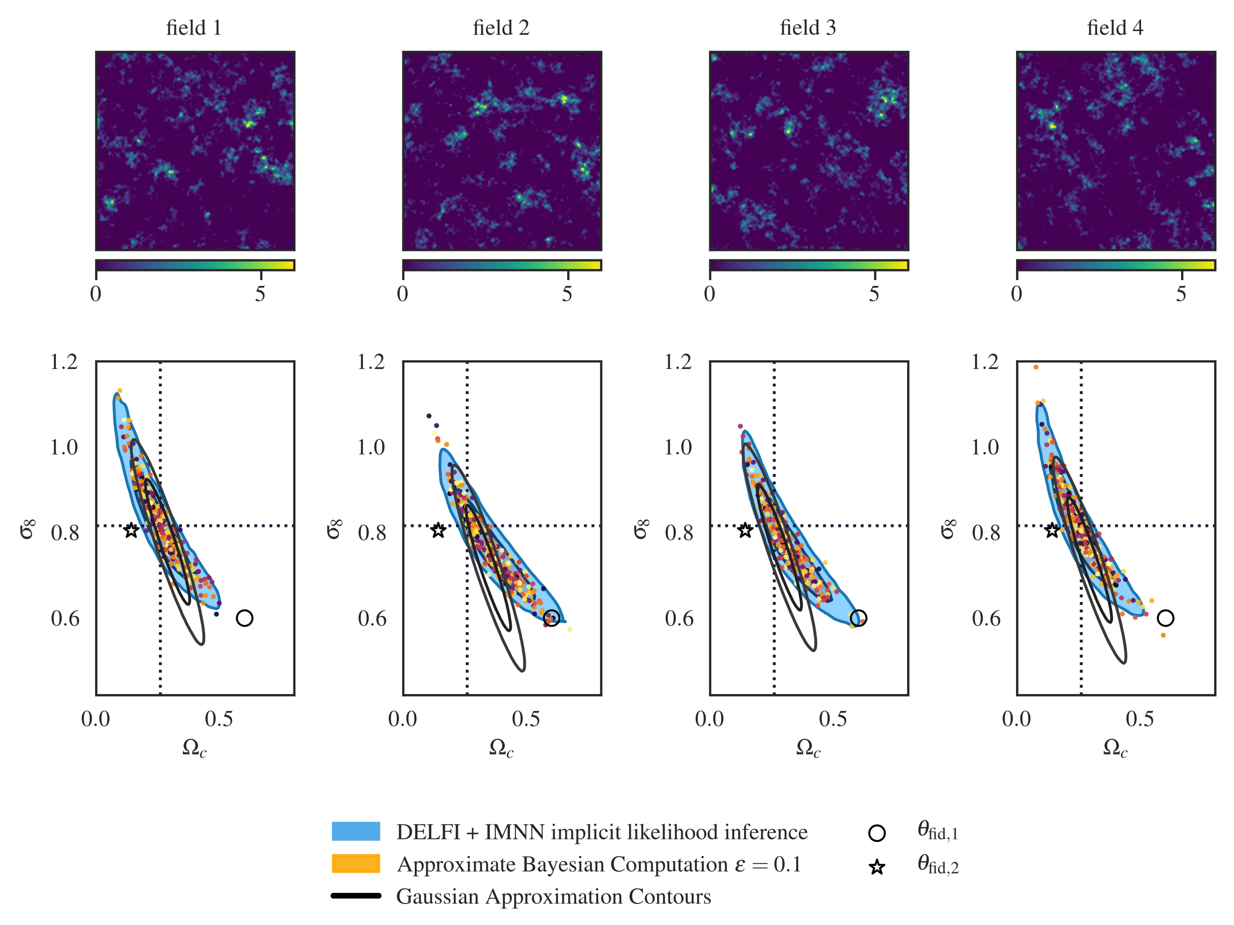


Makinen et al (2021) arXiv:2107.07405

Final inference

-ABC requires 12,000 simulations over prior to obtain 350 accepted points

-DELFI requires 4000 simulations sampled in batches of 1000 from posterior



Makinen et al (2021) arXiv:2107.07405

Takeaways

- Optimal nonlinear compression means we can represent (losslessly) massive simulations with a handful of numbers
- Density estimation massively reduces the number of simulations needed for inference (only $\mathcal{O}(1000)$ versus $\mathcal{O}(10{,}000)$ for Approximate Bayesian Computation

get the code !

Browser-based tutorial: https://bit.ly/imnn-cosmo

Github: https://github.com/tlmakinen/FieldIMNNs

THANK YOU !

(Stay tuned for questions)

lucas.makinen@gmail.com
timothy.makinen@cfa.harvard.edu

https://github.com/tlmakinen

@LucasMakinen

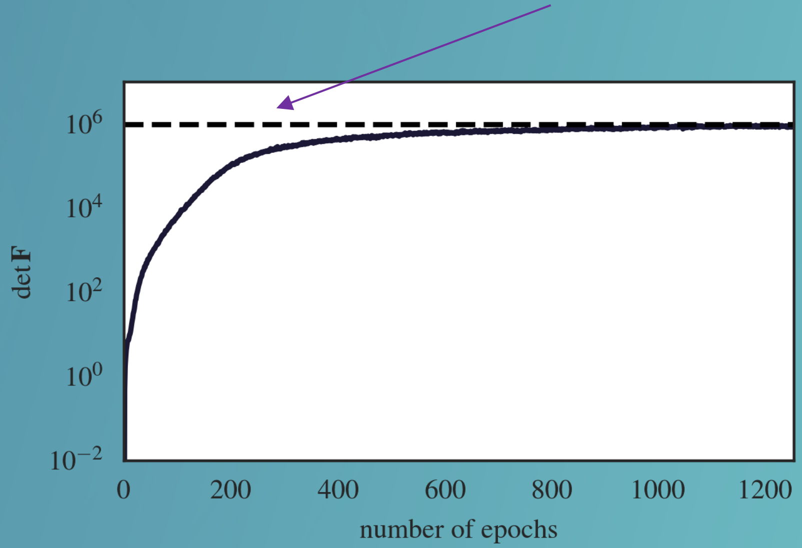
Verifying the pipeline

Want to learn "cosmological" parameters (A, B) from Gaussian fields generated by power spectrum

$$P(k) = Ak^{-B}$$

Train until Fisher information is maximised *at a fiducial model,* $\theta_{\text{fid}} = (1.0, 0.5)$

Theoretical field information content !

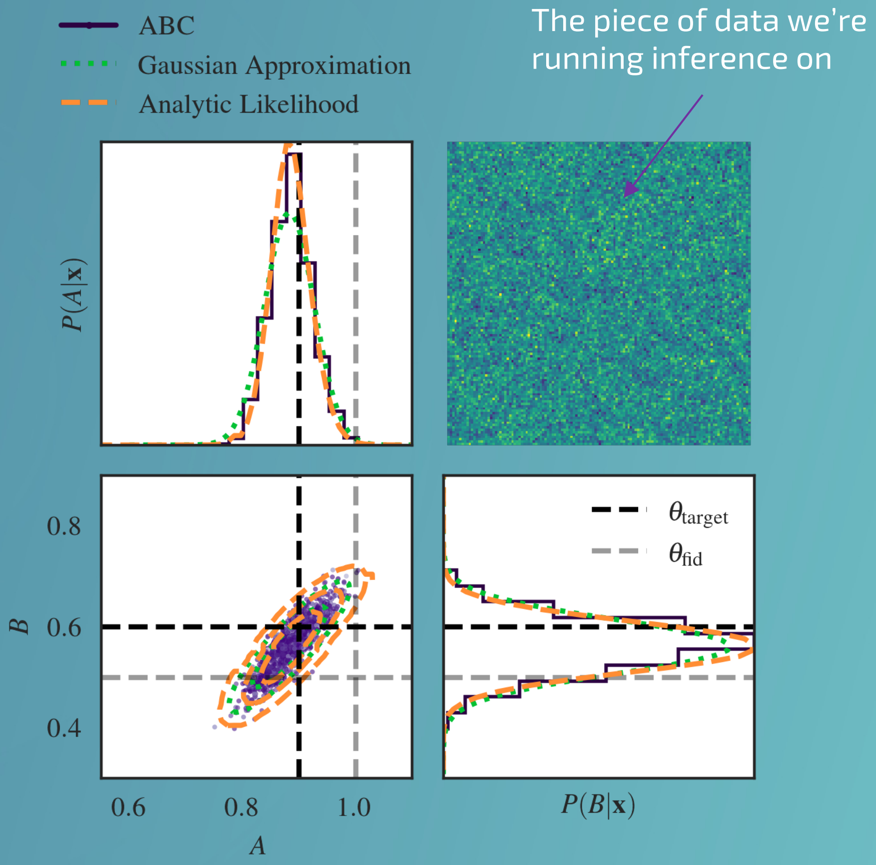


Verifying the pipeline

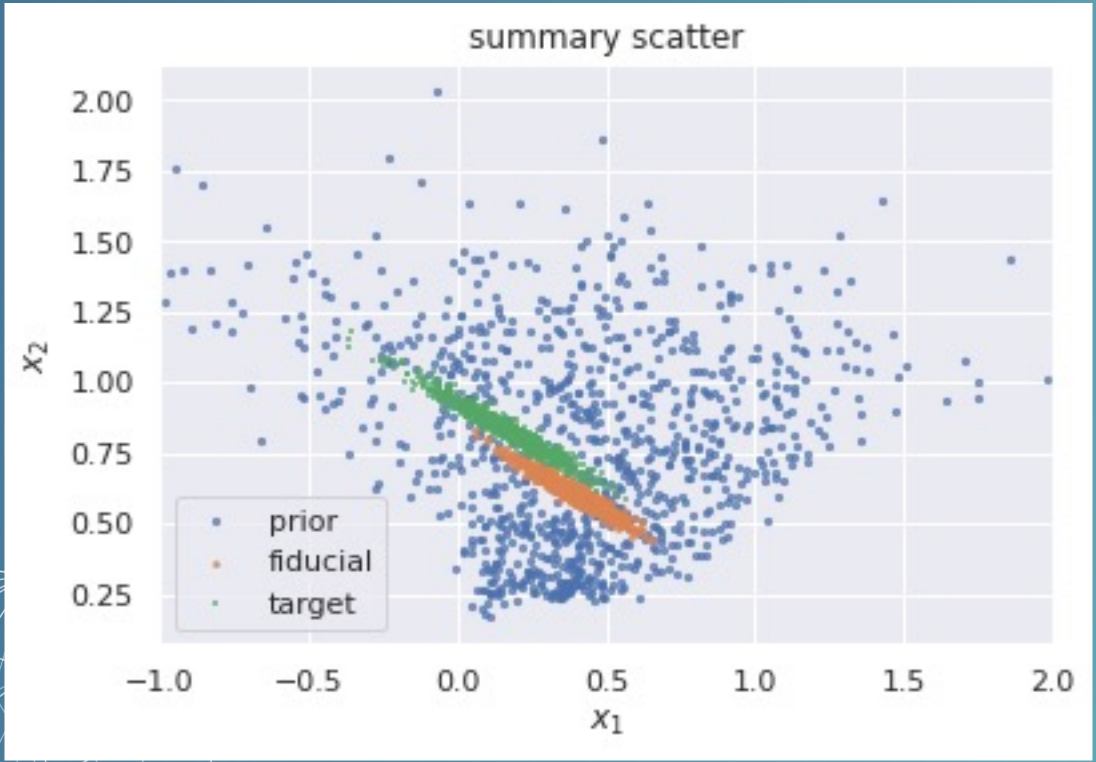
Want to learn "cosmological" parameters (A, B) from fields generated by power spectrum

$$P(k) = Ak^{-B}$$

Run Approximate Bayesian Computation (ABC) on target data with $\theta_{\text{target}} = (0.9, 0.6)$



What do IMNN outputs look like ?



Here we're actually plotting the *score estimates* of parameters computed from the network outputs. Score estimates for a simulation's parameters should be easier for the CMAF to learn than raw neural network outputs

Approximate Bayesian Computation for IMNN summaries

for every *i* simulation, compute:

$$\rho = \sqrt{(\mathbf{x}_i - \mathbf{x}_{\text{target}})^T \mathrm{F}_{\text{IMNN}} (\mathbf{x}_i - \mathbf{x}_{\text{target}})}$$

if $\rho < \epsilon$, keep simulation.