Implicit Likelihood Inference

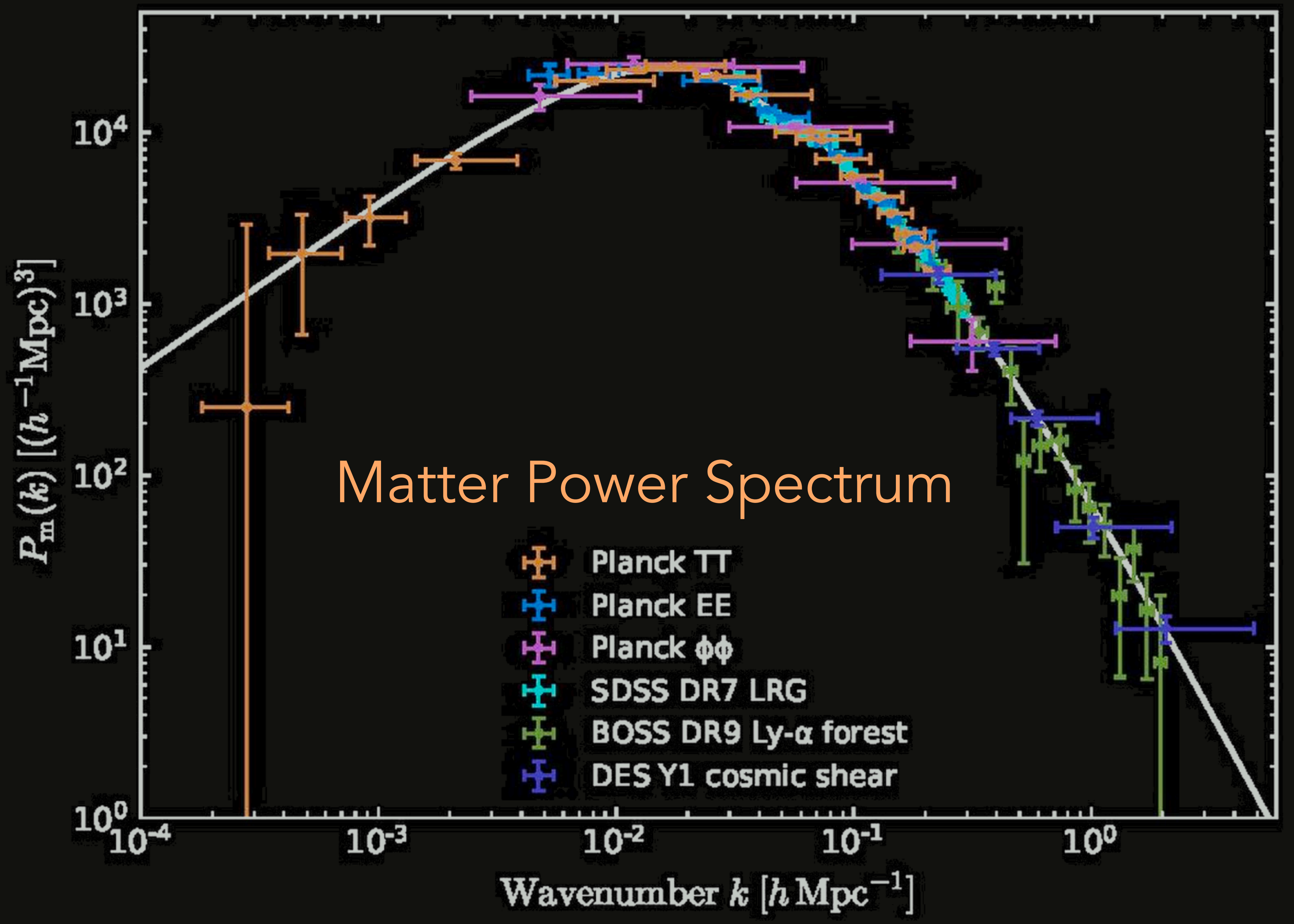
Leander Thiele (Kavli IPMU)

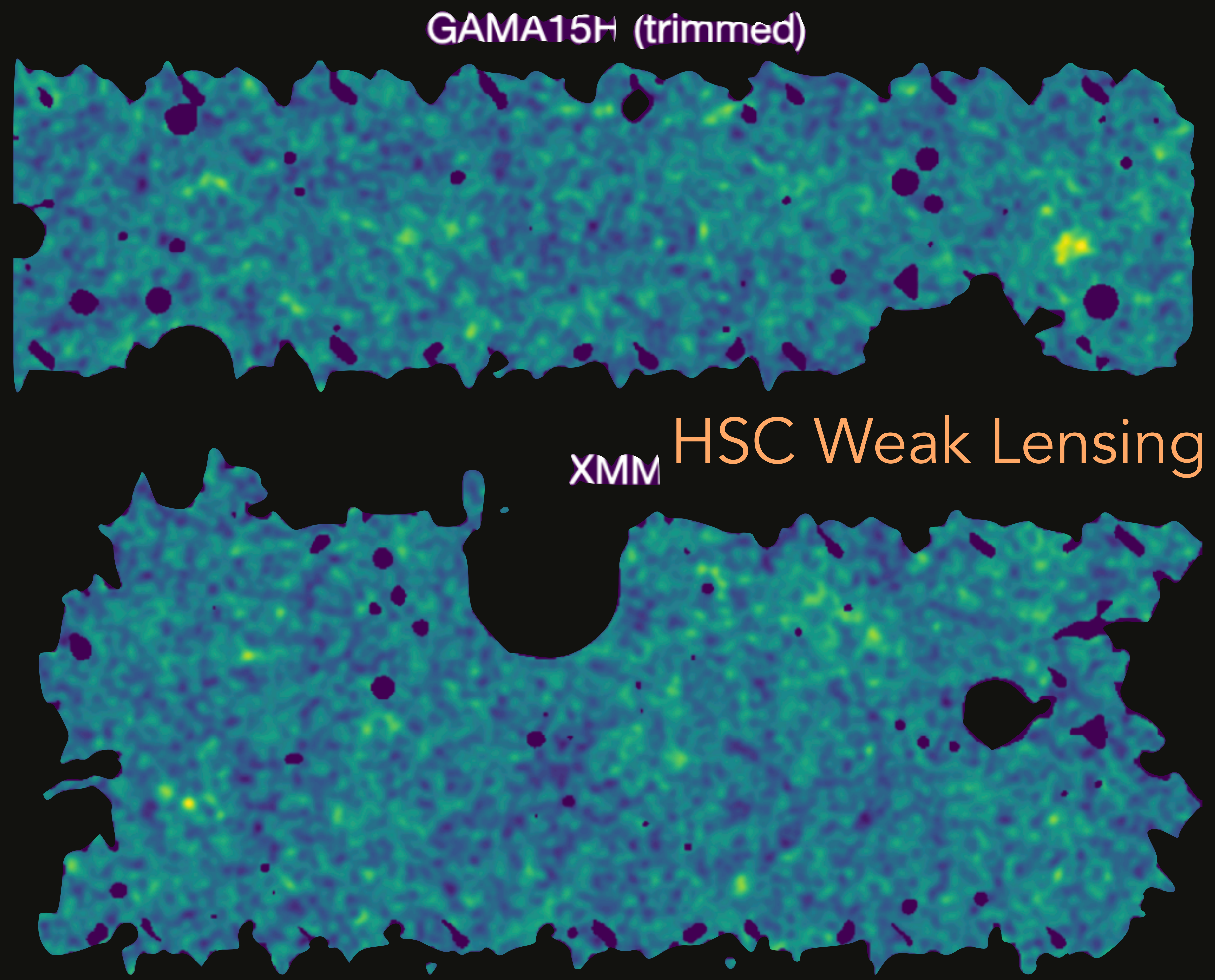
A3Net Seoul, 8/22/2025

Outline

Why should nature have conspired to put all information into the (quasi-)linear modes we happen to be able to describe with perturbation theory?

Accessing the information in non-linear regime typically requires simulation-based methods.





HSC Weak Lensing Mass Maps

Outline

Implicit Likelihood Inference: A tool to solve inverse problems which are implicitly defined through simulations, typically using deep neural networks

1. Motivation & Overview of Methods

2. Example application

3. Challenges & next steps

Bayesian inference

$$\underbrace{\boxed{P(\text{parameters} \mid \text{data})}}_{\text{posterior}} = \frac{\overbrace{\boxed{P(\text{data} \mid \text{parameters})}}^{\text{likelihood}} \; \overbrace{\boxed{P(\text{parameters})}}^{\text{prior}}}{\underbrace{\boxed{P(\text{data})}}_{\text{evidence}}}$$

Bayesian inference

More concretely:

θ=interesting parameters, η=nuisance parameters, ζ=initial conditions,

x=data, m=model:

$$P(x \mid \theta) = \int D\eta\, D\zeta\, \delta[\, x - m(\theta, \eta, \zeta)\,]$$

$$\underbrace{\boxed{P(\text{parameters} \mid \text{data})}}_{\text{posterior}} = \frac{\overbrace{\boxed{P(\text{data} \mid \text{parameters})}}^{\text{likelihood}}\ \overbrace{\boxed{P(\text{parameters})}}^{\text{prior}}}{\underbrace{\boxed{P(\text{data})}}_{\text{evidence}}}$$

Bayesian inference

More concretely:

θ=interesting parameters, η=nuisance parameters, ζ=initial conditions,

x=data, m=model:

$$P(x \mid \theta) = \int D\eta\, D\zeta\, \delta[\, x - m(\theta, \eta, \zeta)\,]$$

initial conditions $\zeta \sim P(\zeta)$,
e.g. quantum fluctuations

nuisance parameters $\eta \sim P(\eta)$,
e.g. star formation model

parameters $\theta \sim P(\theta)$,
e.g. Ω_m

observable data x,
e.g. catalog of galaxies

physical process $m(\theta, \eta, \zeta)$

Bayesian inference

More concretely:

θ=interesting parameters, η=nuisance parameters, ζ=initial conditions,

x=data, m=model:

$$P(x \mid \theta) = \int D\eta \, D\zeta \; \delta[\, x - m(\theta, \eta, \zeta) \,]$$

Traditional case:

$$P(x \mid \theta) = \int D\eta \; \mathrm{Gaussian}[\, x - \mu(\theta, \eta), \Sigma \,]$$

[do remaining low-dimensional η-integral with Monte Carlo]

initial conditions $\zeta \sim P(\zeta)$,
e.g. quantum fluctuations

nuisance parameters $\eta \sim P(\eta)$,
e.g. star formation model

parameters $\theta \sim P(\theta)$,
e.g. Ω_m

observable data x,
e.g. catalog of galaxies

physical process $m(\theta, \eta, \zeta)$

Bayesian inference

More concretely:

θ=interesting parameters, η=nuisance parameters, ζ=initial conditions,

x=data, m=model:

$$P(x \mid \theta) = \int D\eta\, D\zeta\, \delta[\, x - m(\theta, \eta, \zeta)\,]$$

Traditional case:

$$P(x \mid \theta) = \int D\eta\, \mathrm{Gaussian}[\, x - \mu(\theta, \eta), \Sigma\,]$$

But what do we do if the Gaussian approximation doesn't hold?

→ assume we have simulator that evaluates m(θ, η, ζ) accurately

Neural Implicit Likelihood Inference (ILI)

neural likelihood estimation (NLE)

neural posterior estimation (NPE)

$$P(\text{parameters}\,|\,\text{data}) = \frac{P(\text{data}\,|\,\text{parameters}) \quad P(\text{parameters})}{P(\text{data})}$$

neural ratio estimation (NRE)

We are given parameter vectors θ and simulated data vectors x.

Problem is to approximate one of the above probability distributions.

How can we map these tasks to optimization problems solvable with neural networks?

Neural Implicit Likelihood Inference (ILI)

neural likelihood estimation (NLE) neural posterior estimation (NPE)

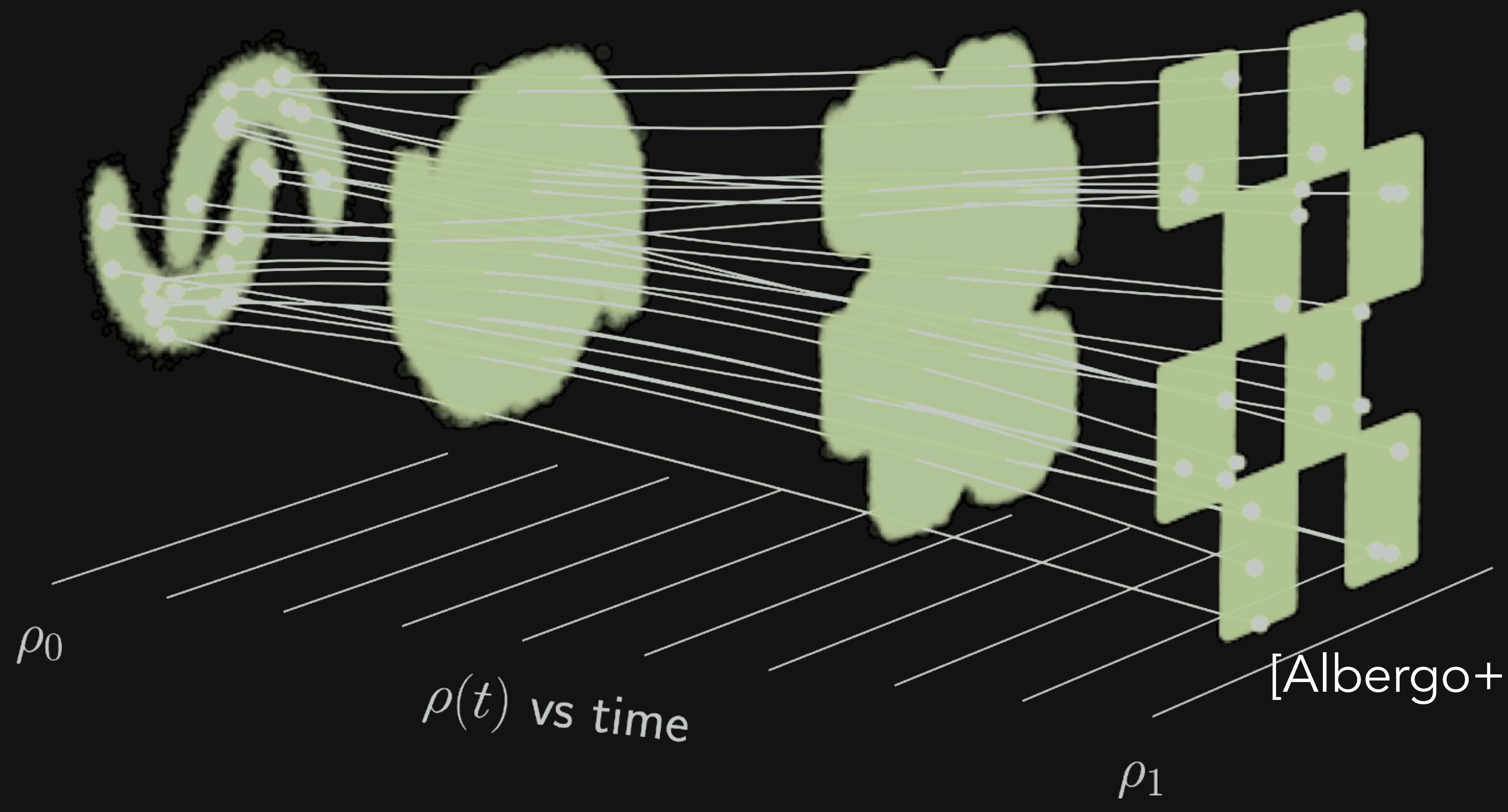
$$P(\text{parameters} \,|\, \text{data}) = \frac{P(\text{data} \,|\, \text{parameters}) \quad P(\text{parameters})}{P(\text{data})}$$

neural ratio estimation (NRE)

neural likelihood estimation (NLE):
conditioned normalizing flow q(x|θ)

neural posterior estimation (NPE):
conditioned normalizing flow q(θ|x)

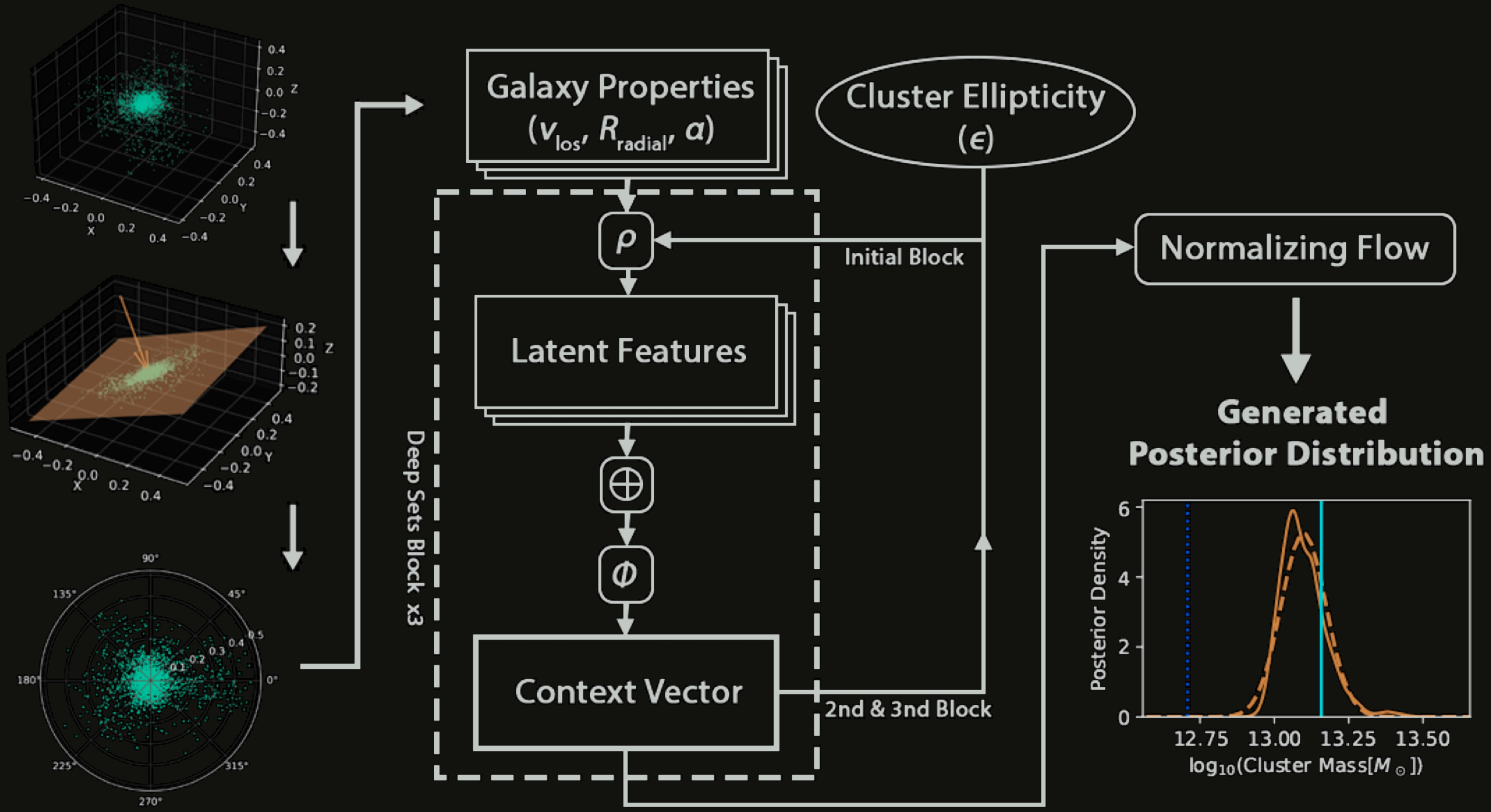
neural ratio estimation (NRE):
classifier between (x, θ) ~p(x, θ) and ~p(x)p(θ)



[Albergo+2023]

Practicality: data size & structure

Usually, data vectors too high dimensional or live in awkward spaces
→ We deal with this by constructing a useful latent representation through *embedding networks*



[B.Y.Wang & LT 2025]

Neural Implicit Likelihood Inference (ILI)

- *in the limit*, any likelihood learnable
- any simulate-able effect can be incorporated
- no formal difference between nuisance parameters and initial conditions
- primary choice at the moment:
  - NPE: empirically good performance, need to deal with flow
  - NRE: classification → super flexible, empirically more tuning required

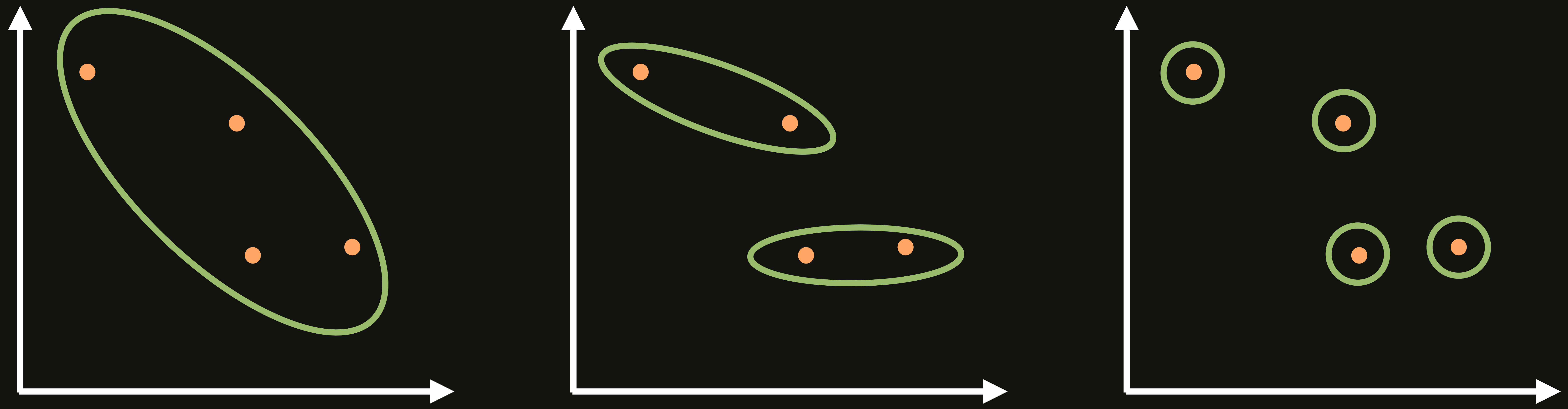
Curse of dimensionality

Let's say: 200,000 simulations, 10 model parameters, 10 data points
→ this is optimistic!

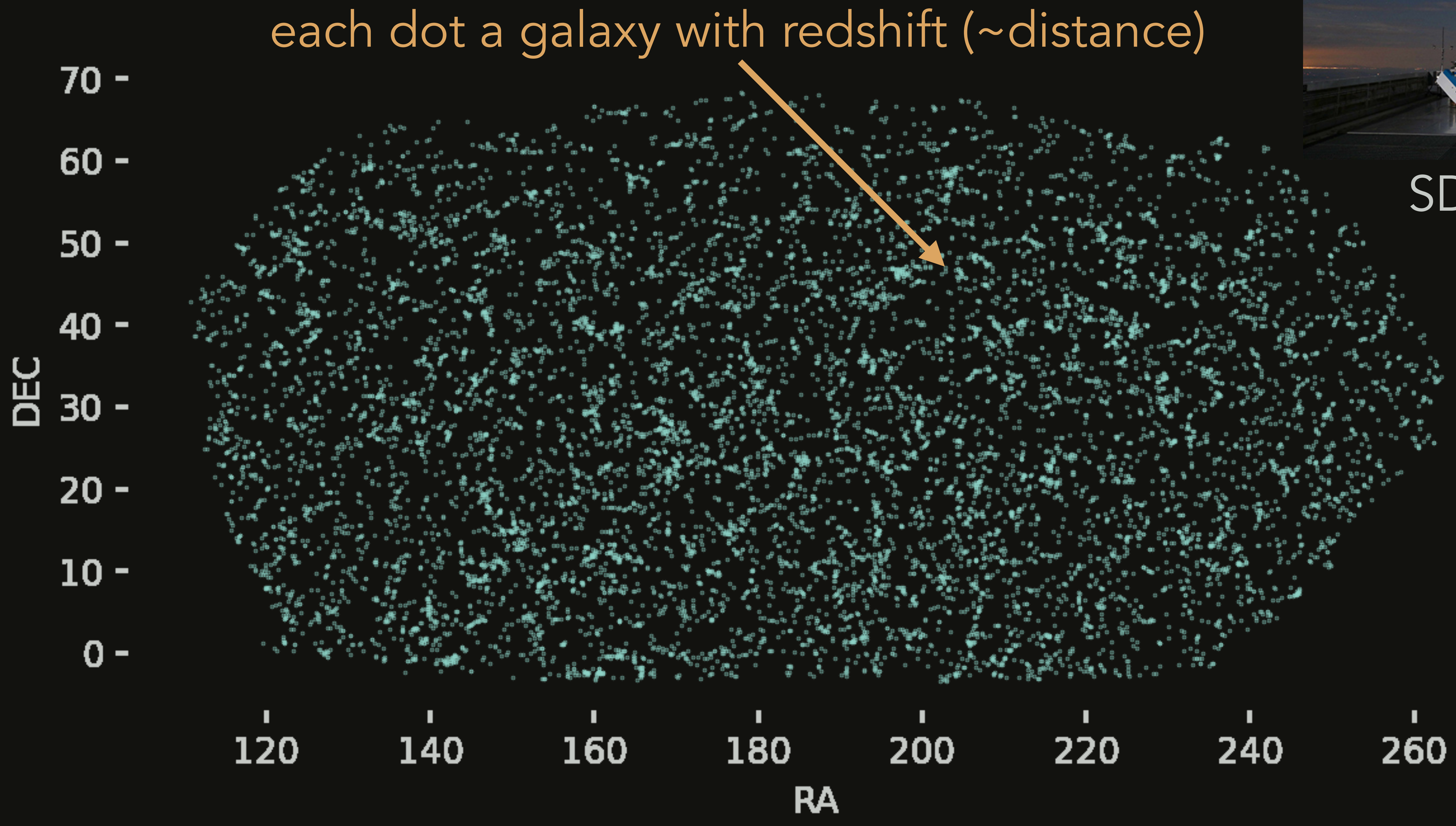
We're trying to learn a density in this 10+10 dimensional space

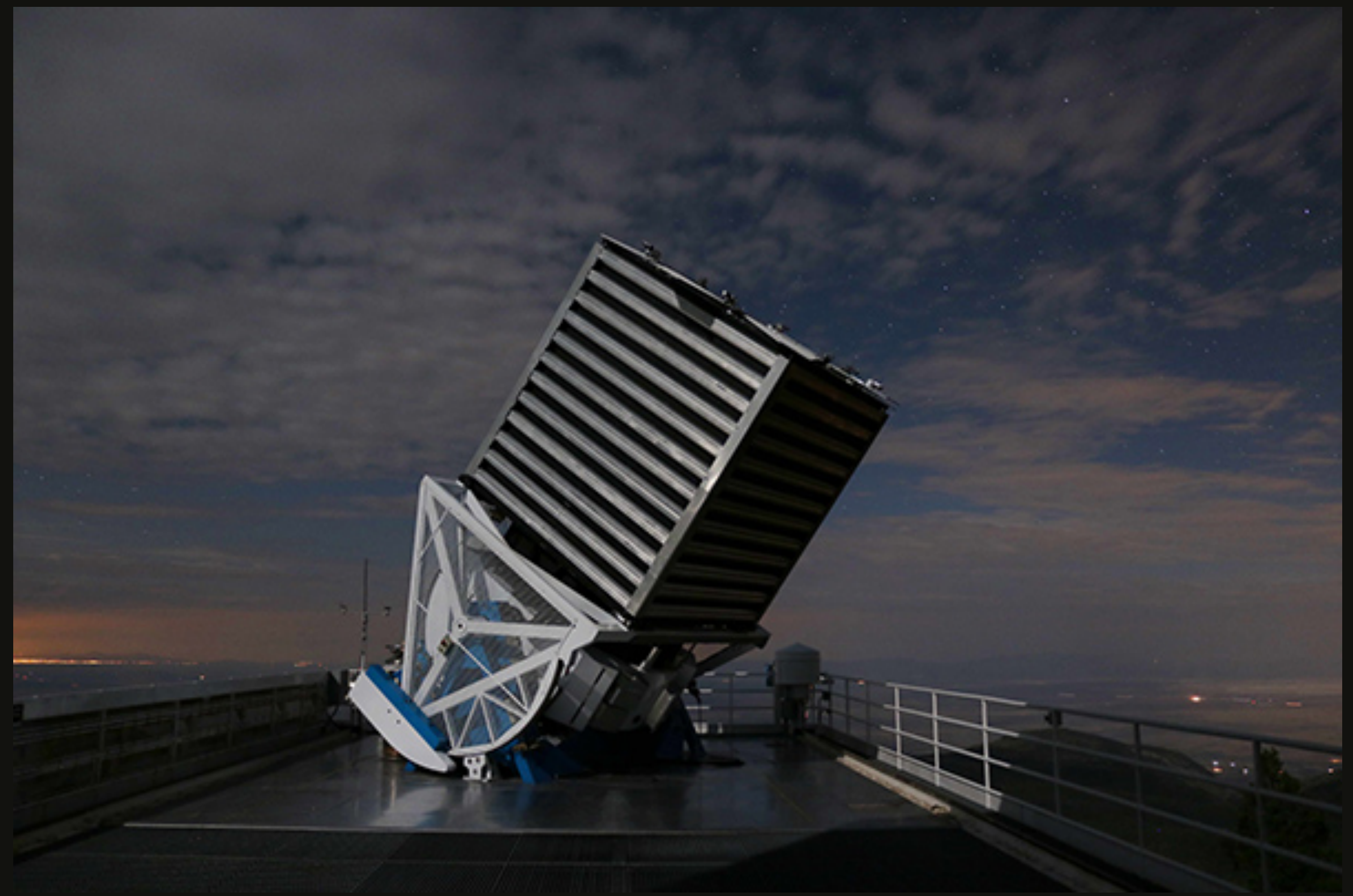
$$(200{,}000)^{1\,/\,(10\,+\,10)} = 1.8, \text{ let's say } 2$$

What is the implicit prior?



A 3-D Map of the Universe





SDSS/BOSS survey

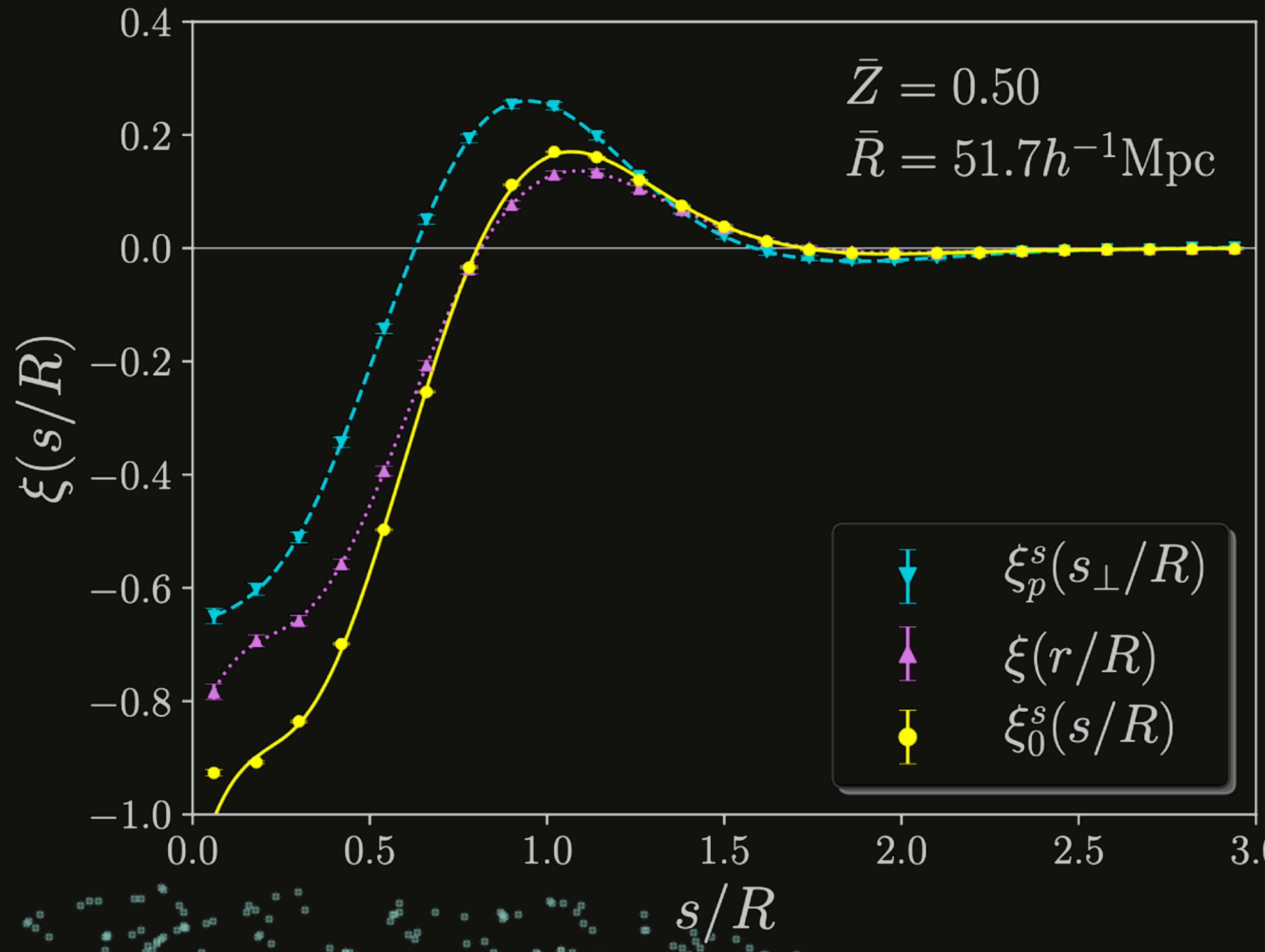
How to summarize this map?

1) pairs of galaxies (power spectrum)

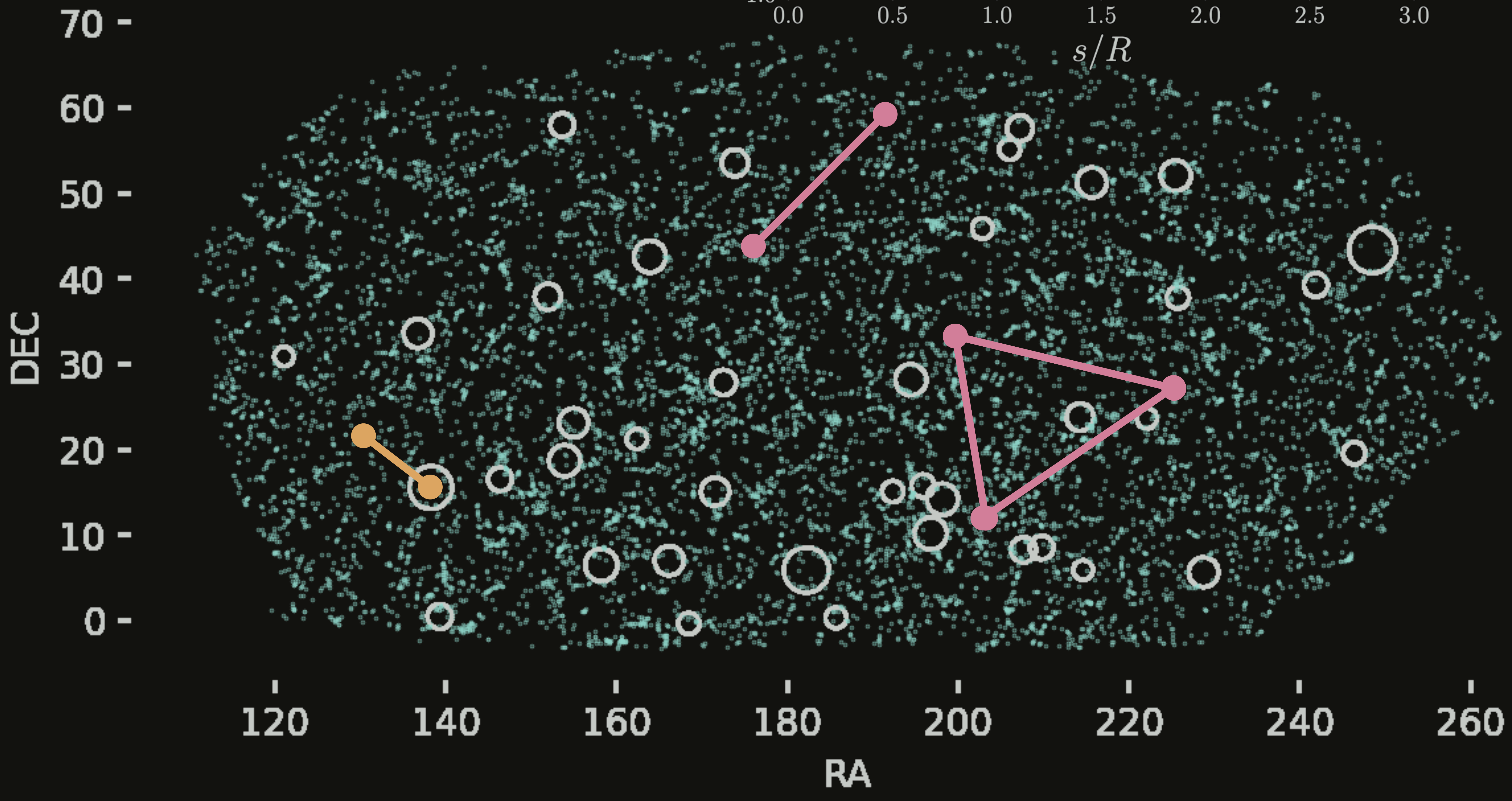
2) triangles of galaxies (bispectrum)

3) …

4) "empty regions": *cosmic voids*
   - size distribution
   - void-galaxy pairs
   - …



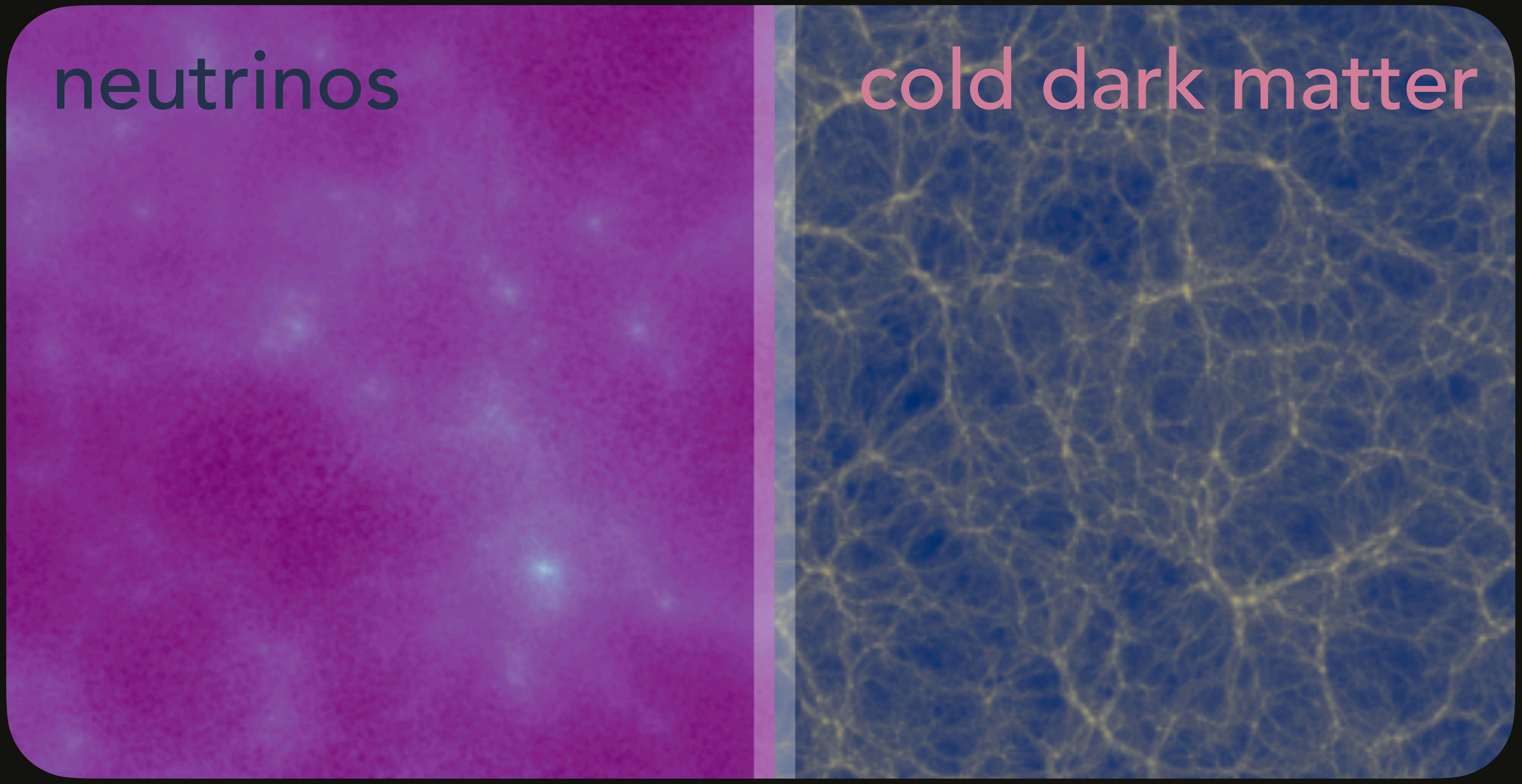
[Hamaus+2020]

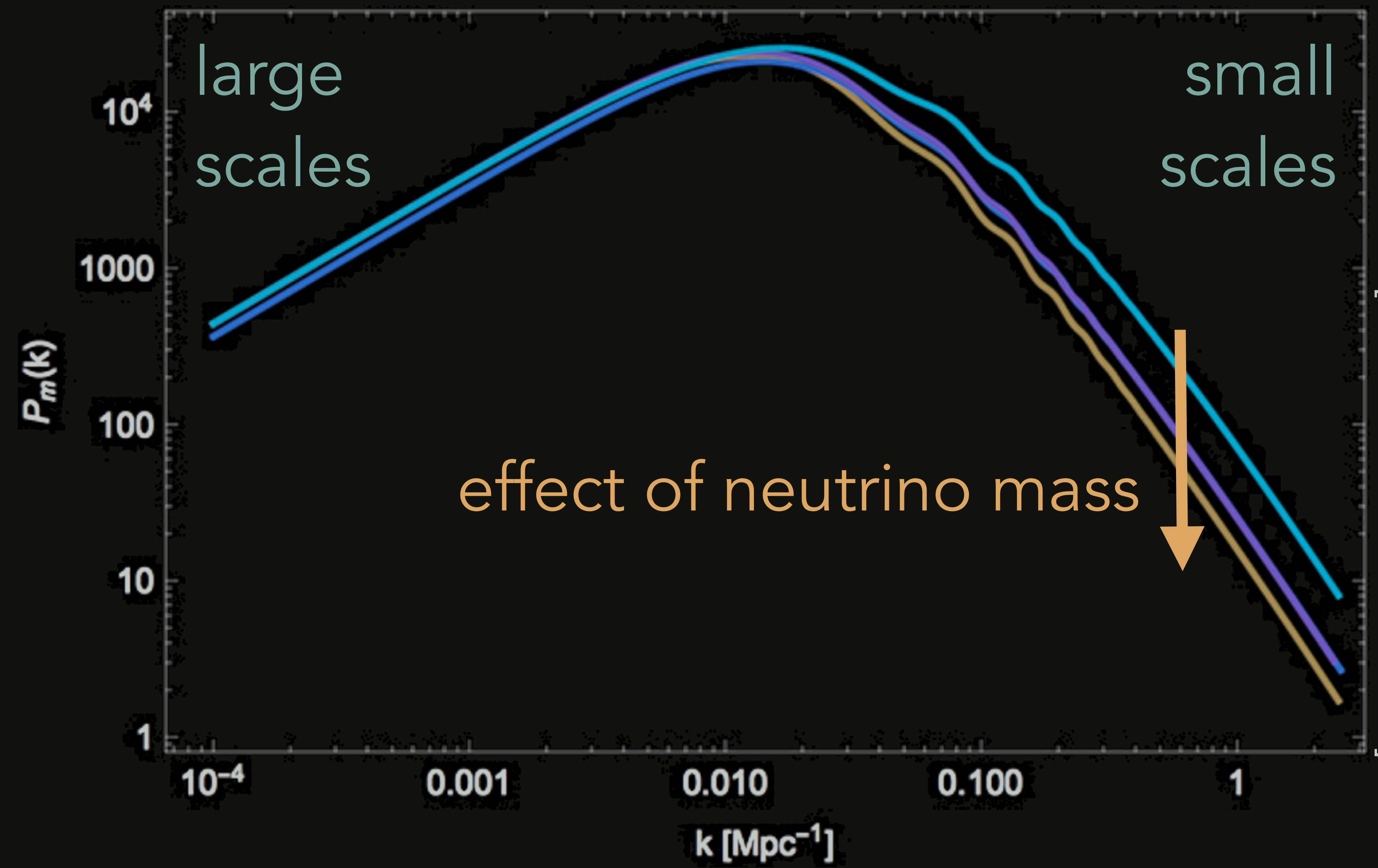


What can voids do for us?

- upweight underdensities → complementary to correlation functions
  - corrections to general relativity
  - dark energy
  - neutrino mass

[FLAMINGO]

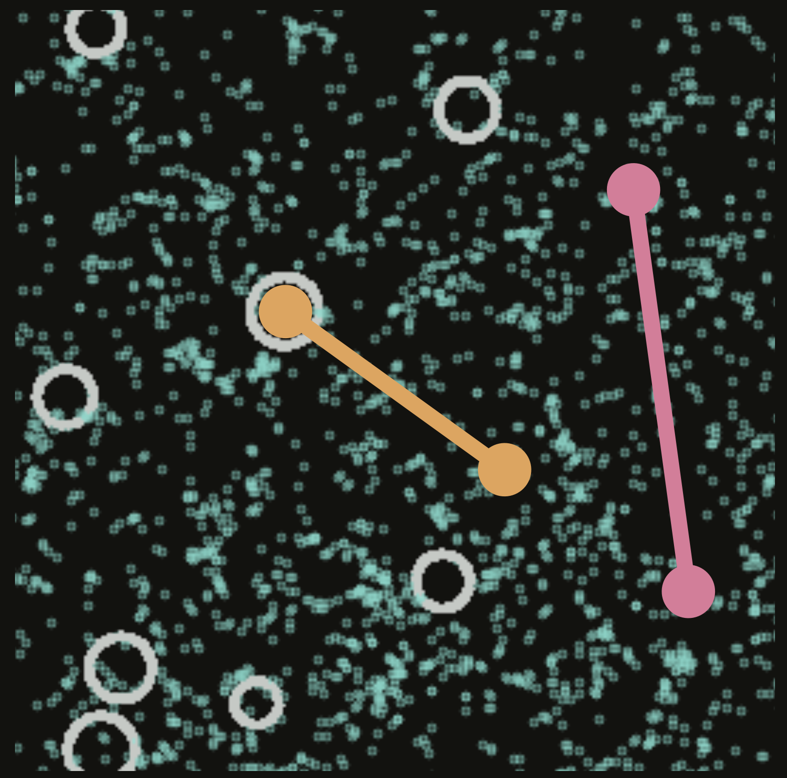




[Gerbino&Lattanzi2018]

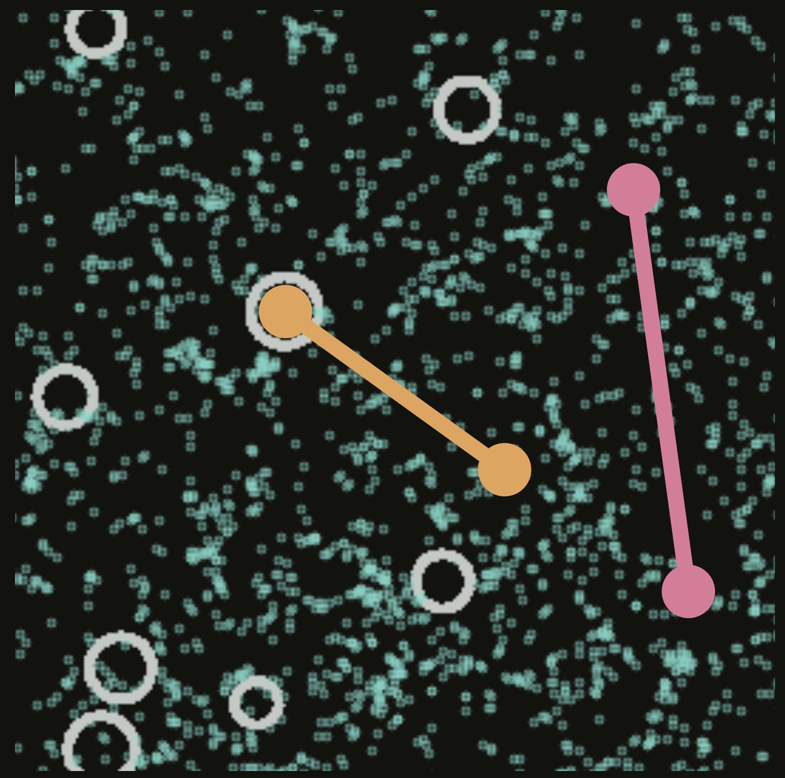
Simulation-based inference

- Want to constrain neutrino mass sum, $\sum m_\nu$, with BOSS data:
  - galaxy auto power spectrum
  - void size function (histogram of void sizes)
  - void galaxy cross power spectrum (void profile)

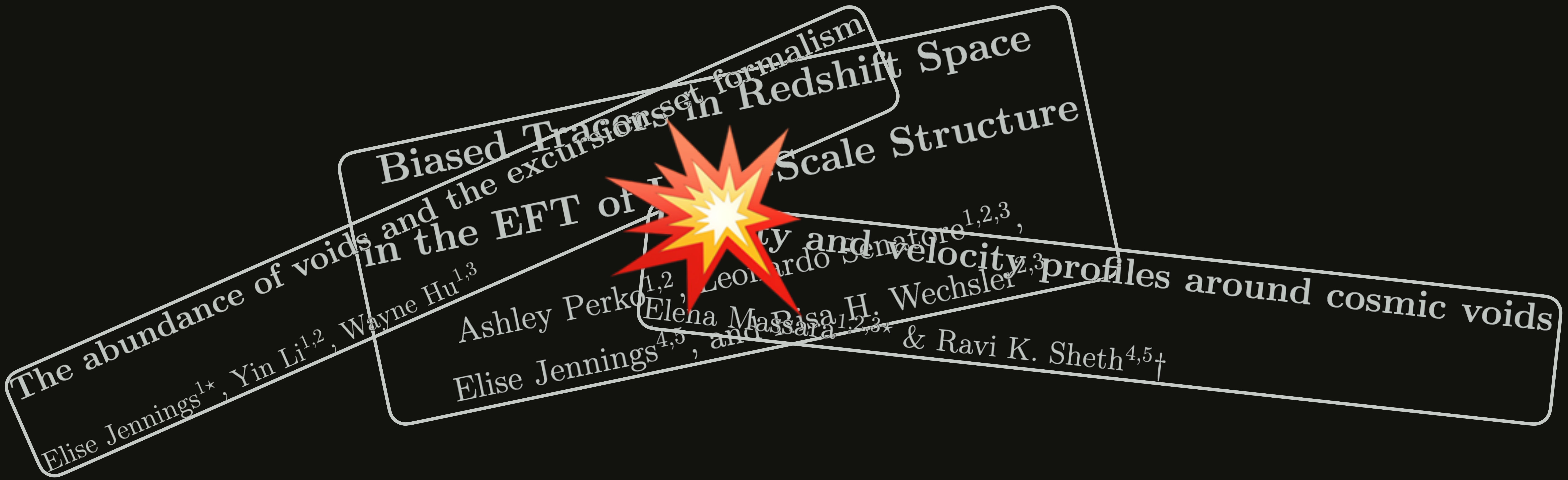


Simulation-based inference

- Want to constrain neutrino mass sum, $\sum m_\nu$, with BOSS data:
  - galaxy auto power spectrum
  - void size function (histogram of void sizes)
  - void galaxy cross power spectrum (void profile)

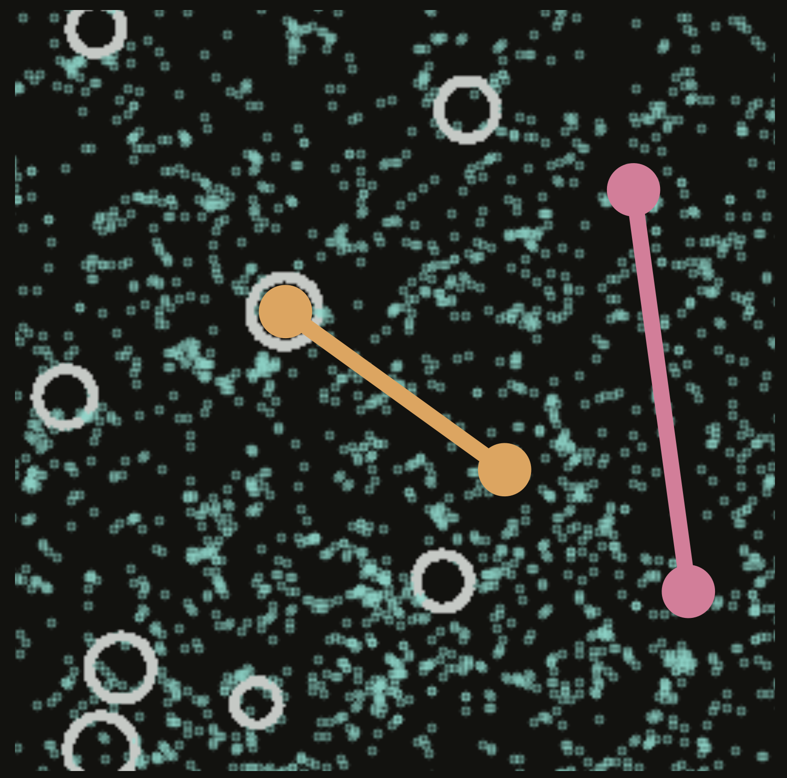


- Joint modeling of these statistics difficult with analytic methods



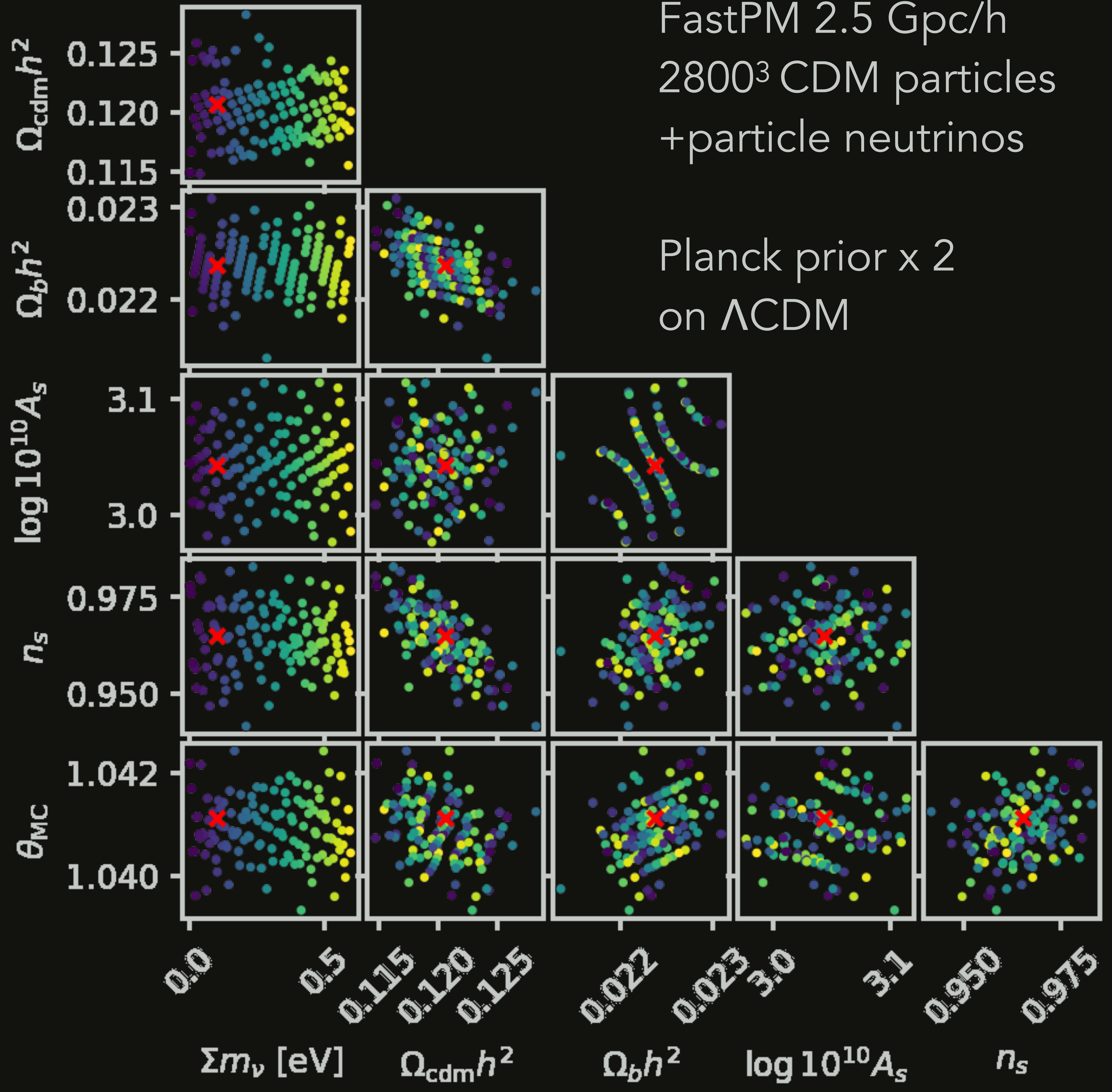
Simulation-based inference

- Want to constrain neutrino mass sum, $\sum m_\nu$, with BOSS data:
  - galaxy auto power spectrum
  - void size function (histogram of void sizes)
  - void galaxy cross power spectrum (void profile)



- Joint modeling of these statistics difficult with analytic methods
- Thus, resort to simulations (PM + HOD)
- Likelihood unknown → Implicit-Likelihood Inference

Simulations



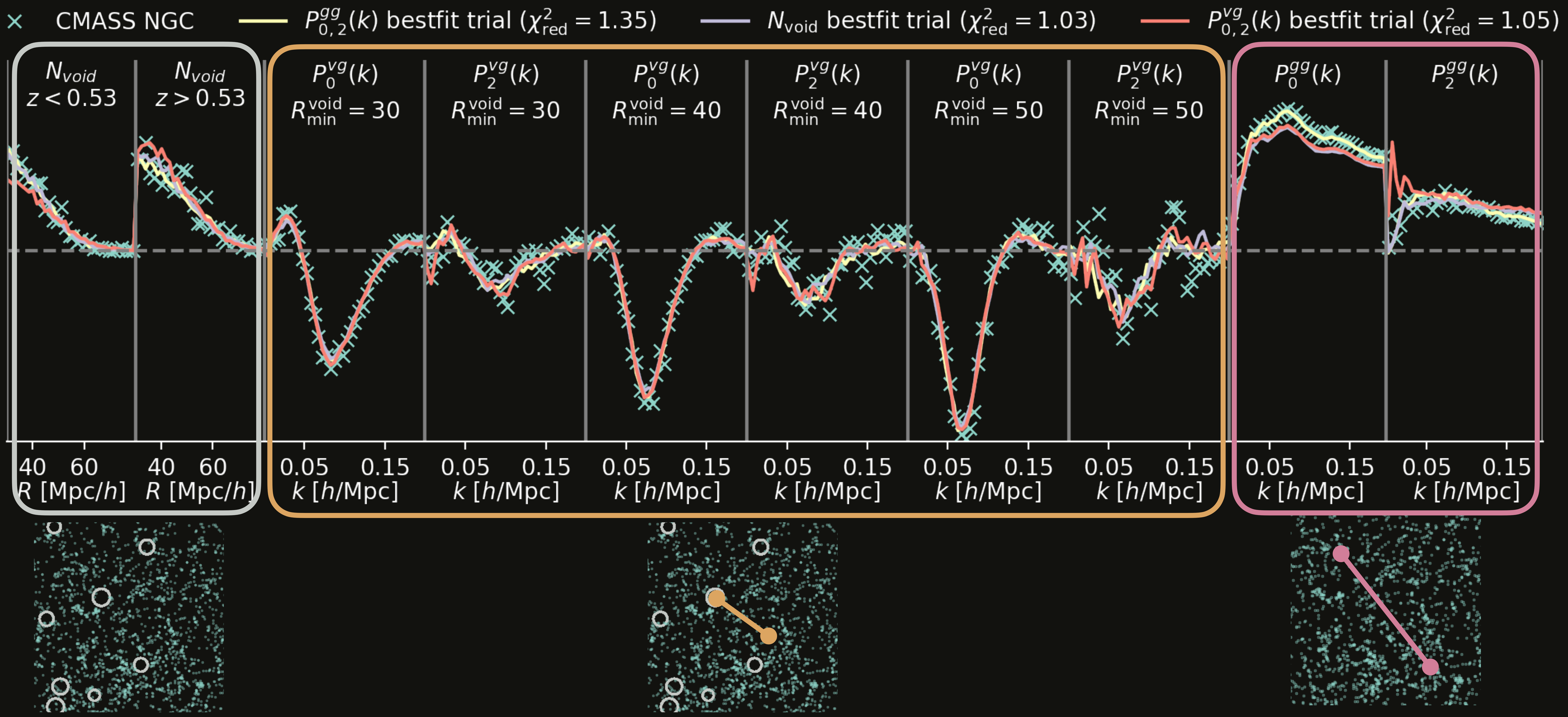
FastPM 2.5 Gpc/h
2800^3 CDM particles
+particle neutrinos

Planck prior x 2
on ΛCDM

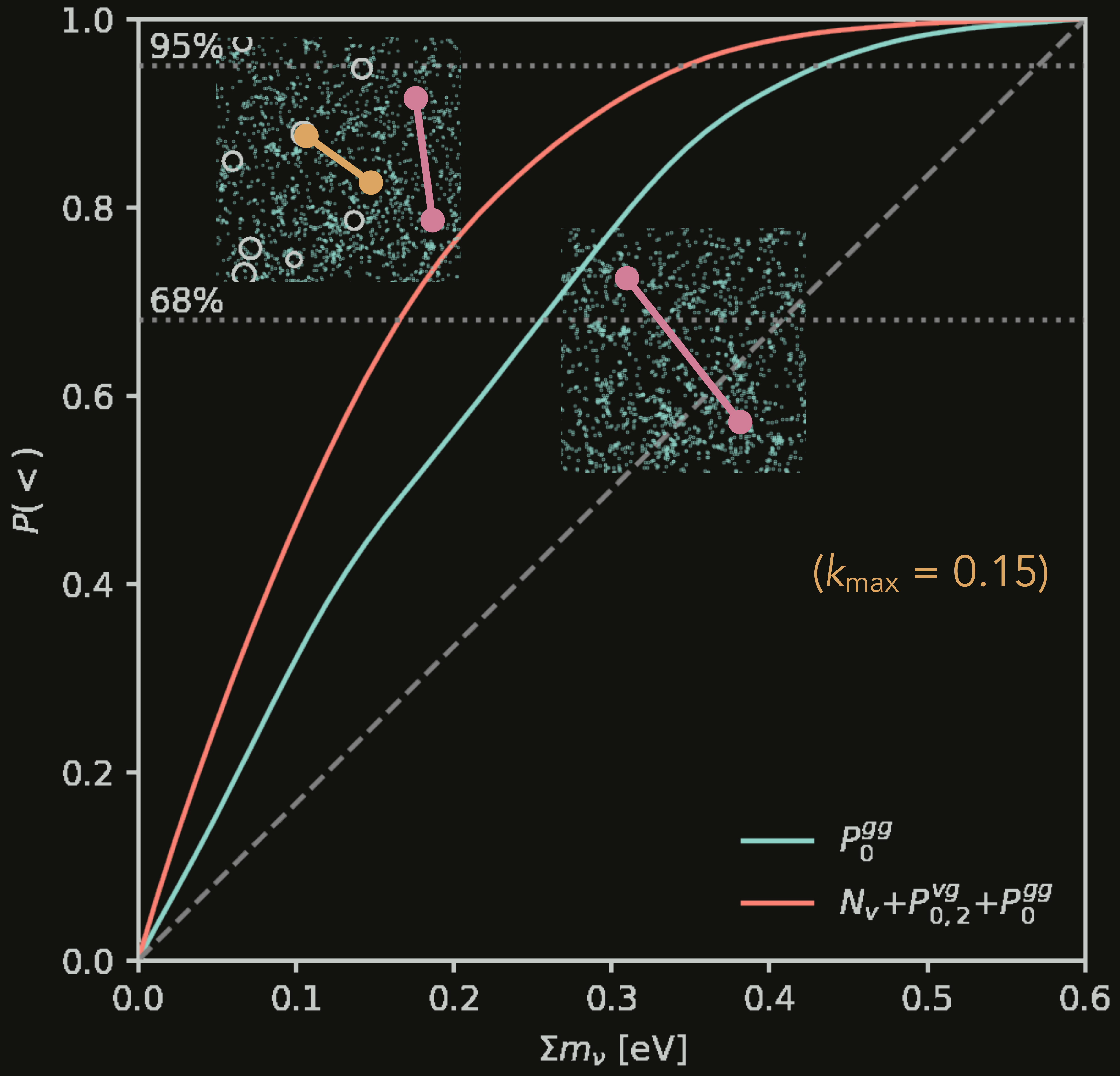
- populate gravity-only simulations with galaxies using HOD
- project on lightcone and add survey realism

Data Vector

Use MOPED compression to reduce dimensionality.



Main posterior



With conservative scale cut of k_{max}=0.15 hMpc^{-1}, voids tighten upper bound on neutrino mass.

Issues away from the limit

- *in the limit*, any likelihood learnable
- what is the limit?
    - infinite model expressivity (usually ok in astro & cosmology)
    - ability to find good global optimum (usually ok)
    - infinite training set size / fast & accurate simulation codes

$$L = \int_{p(x,\theta)} \mathscr{L} \approx \sum_{\text{training set}} \mathscr{L}_{\text{approx}}$$

Implicit Likelihood Inference in Crisis?

A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful

Joeri Hermans*
Unaffiliated

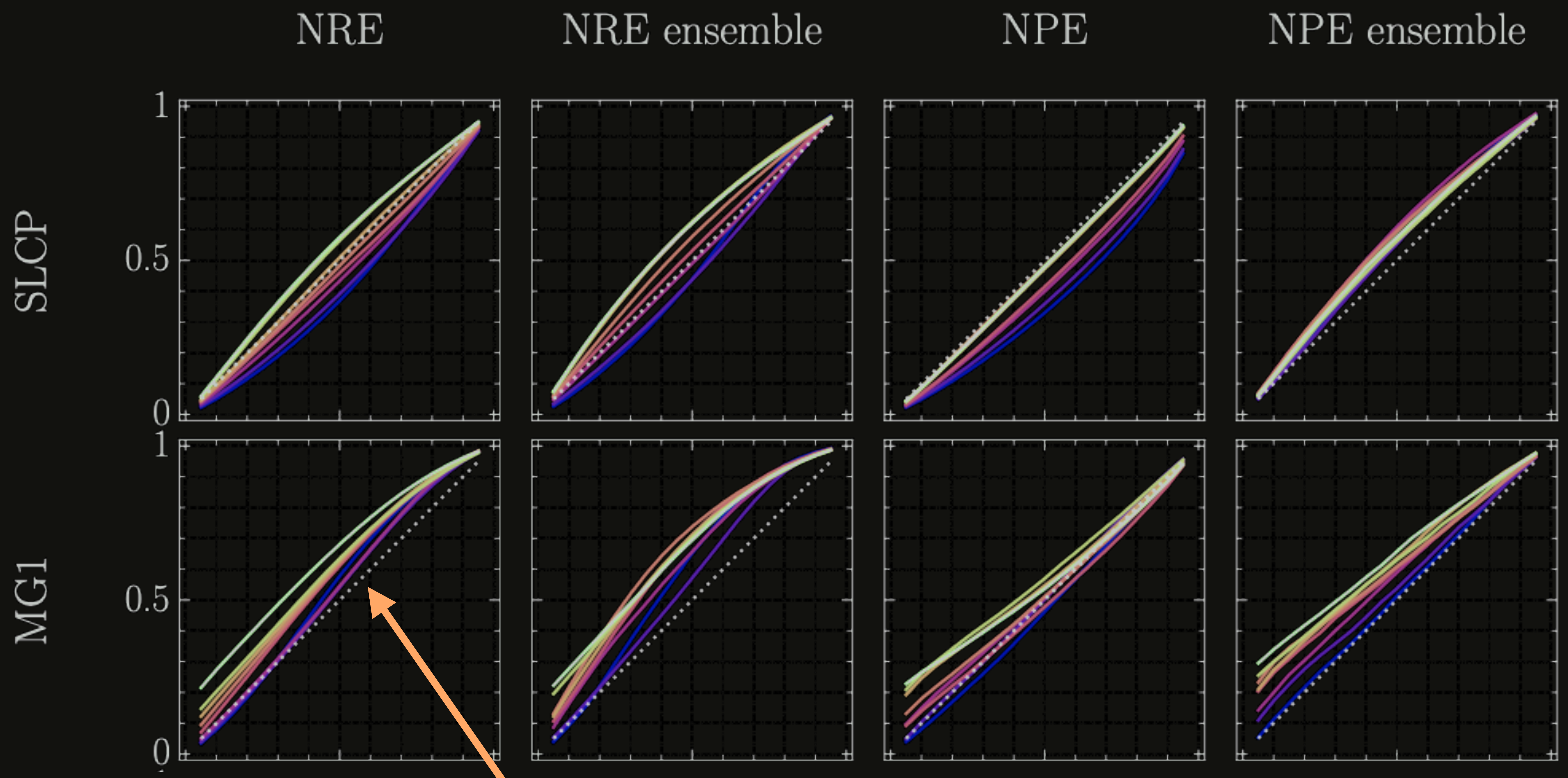
Arnaud Delaunoy*
University of Liège

François Rozet
University of Liège

Antoine Wehenkel
University of Liège

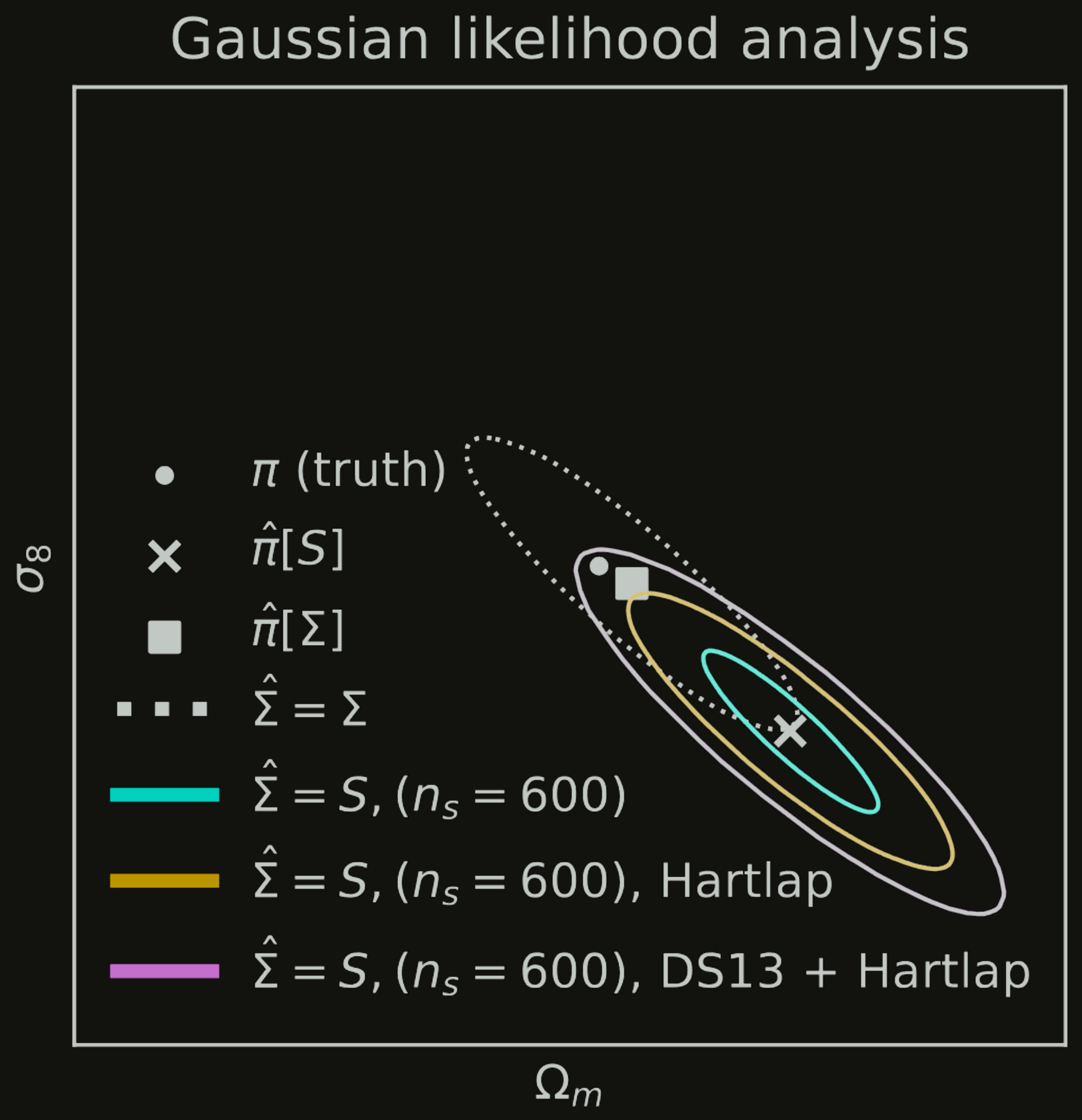
Volodimir Begy
University of Vienna

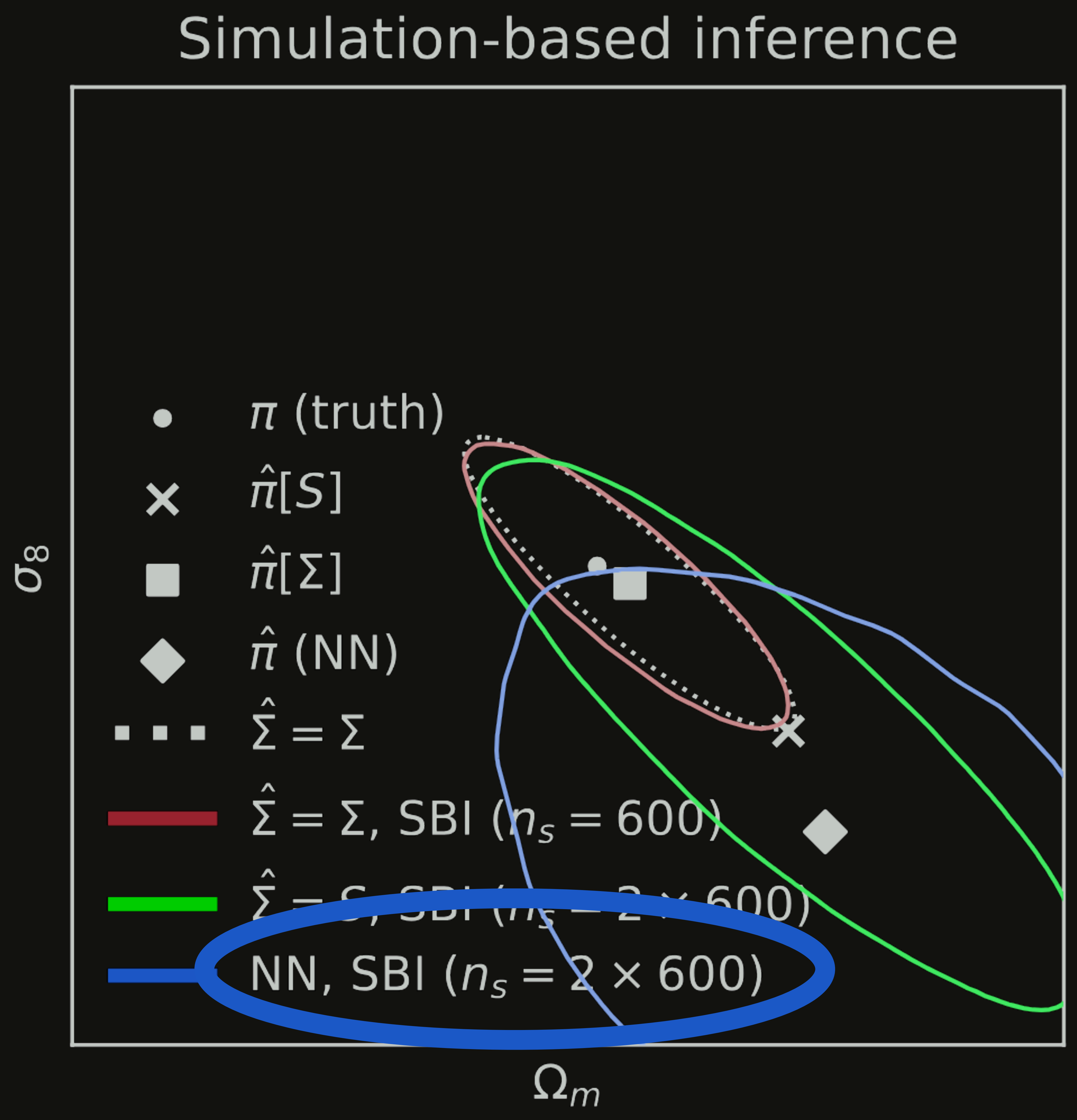
Gilles Louppe
University of Liège



Want to be along diagonal

Implicit Likelihood Inference in Crisis?





[Homer, Friedrich, Gruen 2024]

Implicit Likelihood Inference: Way Forward

The ultimate goal: accurate inferences on physical parameters for analytically intractable inverse problems

Standing in the way: high cost of numerical simulations, limiting training set size and thus accuracy

Progress is being made in several directions:
- bring simulation cost down (hardware, software, ML enhancement)
- improve sample efficiency of ILI
- incorporate information from cheaper simulations (multi-fidelity learning)