

Machine Learning enhanced Bayesian Workflows for Galaxies

Harry Bevins

Work with Will Handley, Pablo Lemos, Peter Sims, Eloy de Lera Acedo,
Anastasia Fialkov, Justin Alsing and Thomas Gessy-Jones

Papers: 2205.12841 and 2207.11457

Code: <https://github.com/htjb/margarine>

Contents

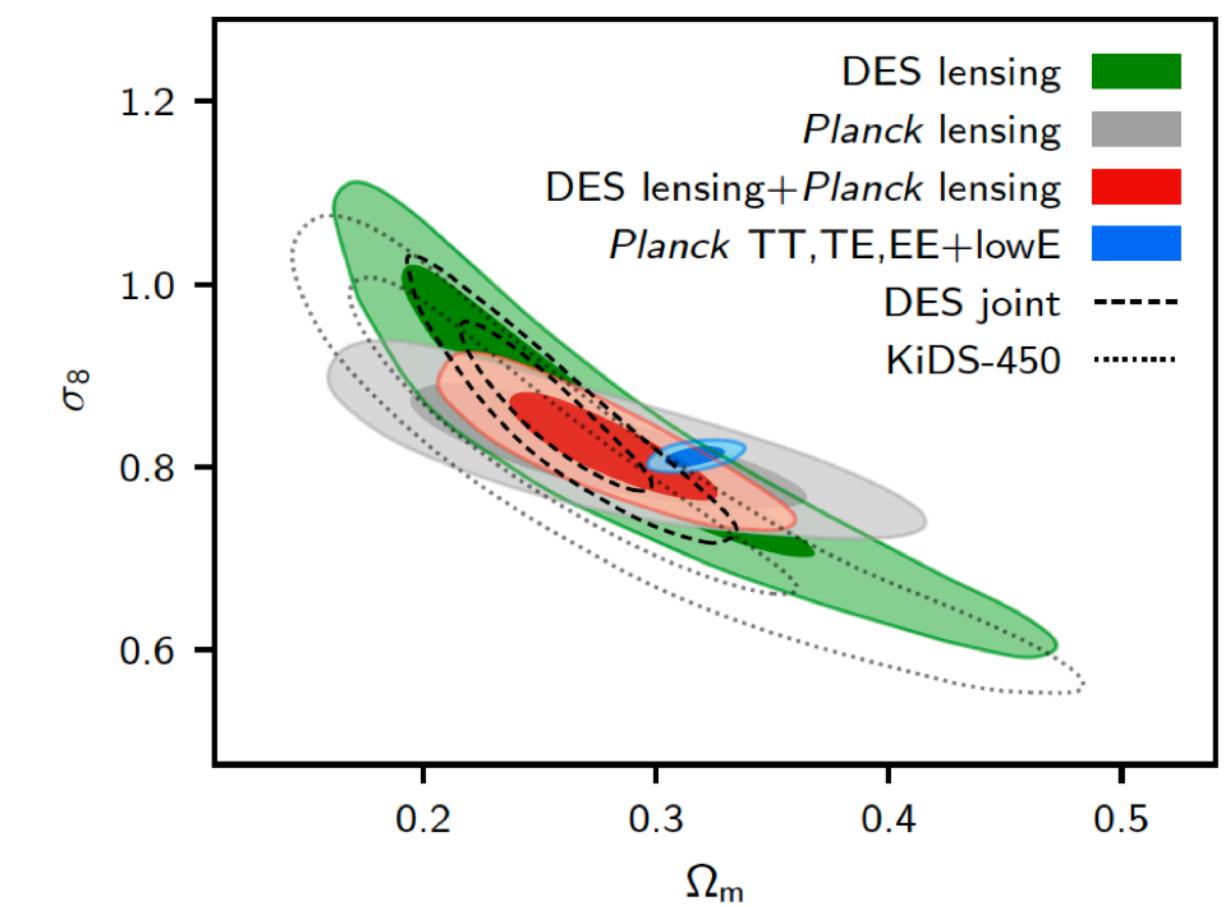
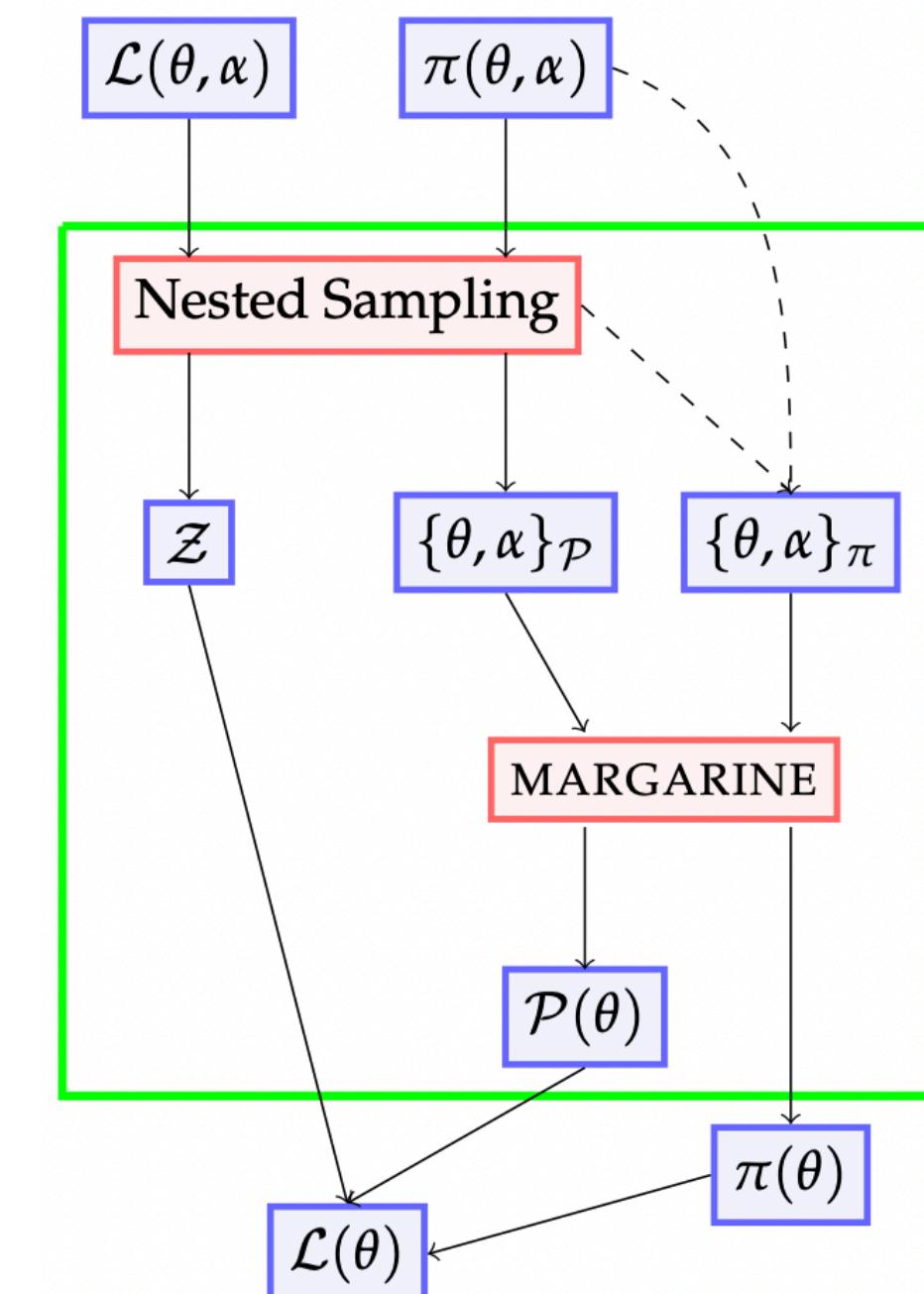
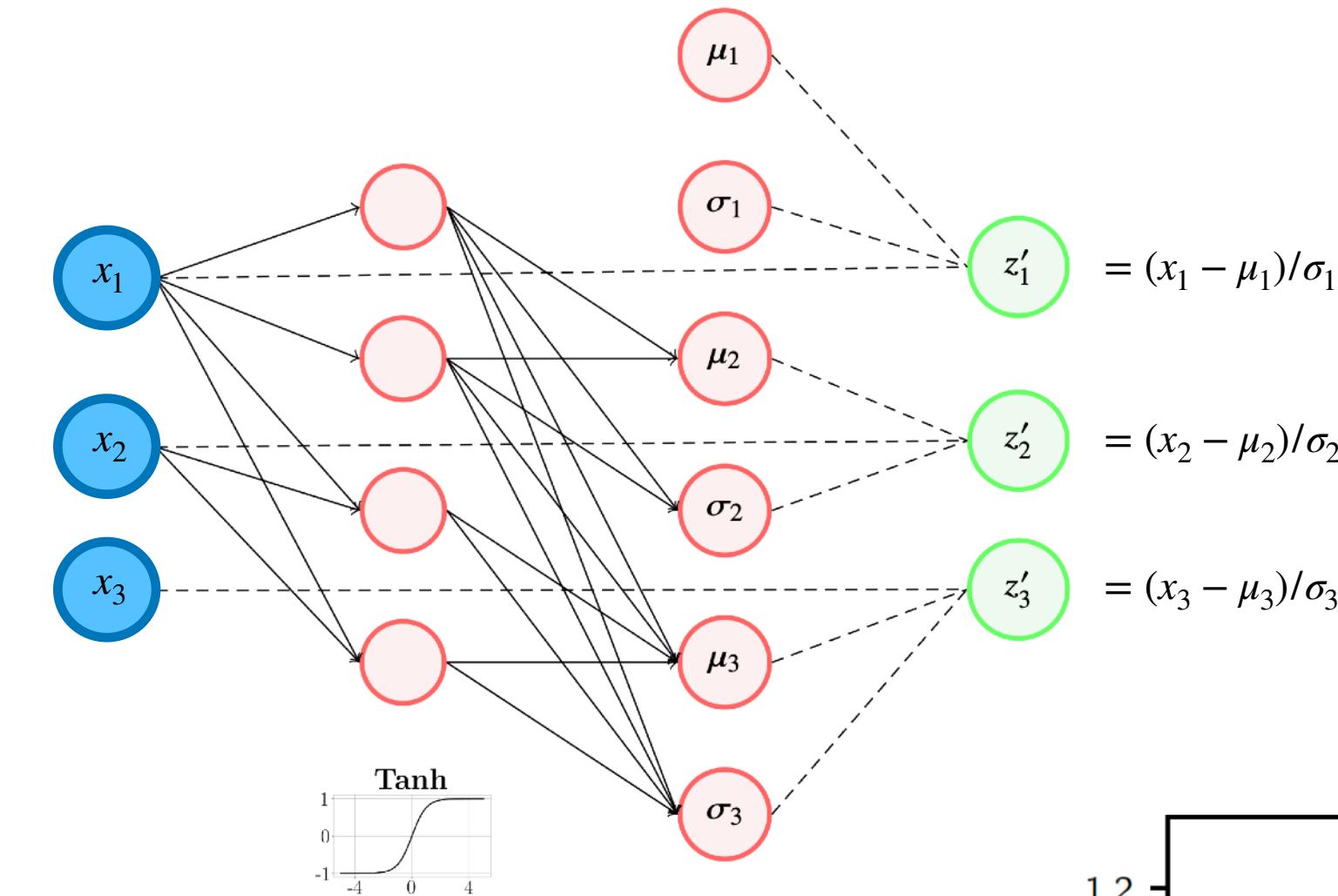
1. The Marginal Parameter Space

2. Normalising Flows

3. Marginal Bayesian Workflows

4. And (possible) Applications

5. Calibrating Tension Statistics



The Marginal Parameter Space

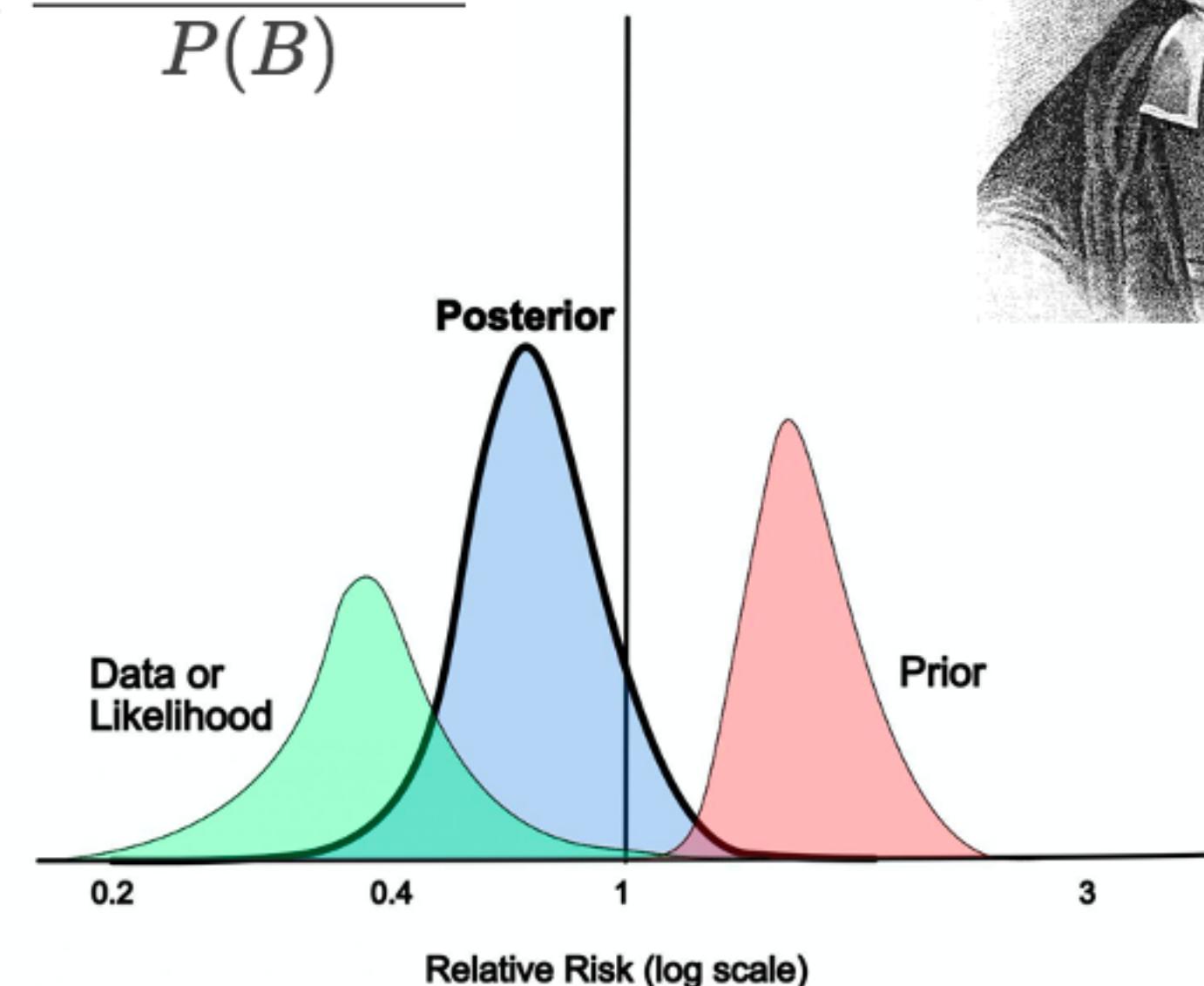
Bayesian Analysis

- In Astrophysics and Cosmology we make a lot of use of Bayesian inference

$$P(\Theta | D, M) = \frac{P(D | \Theta, M)P(\Theta | M)}{P(D | M)}$$

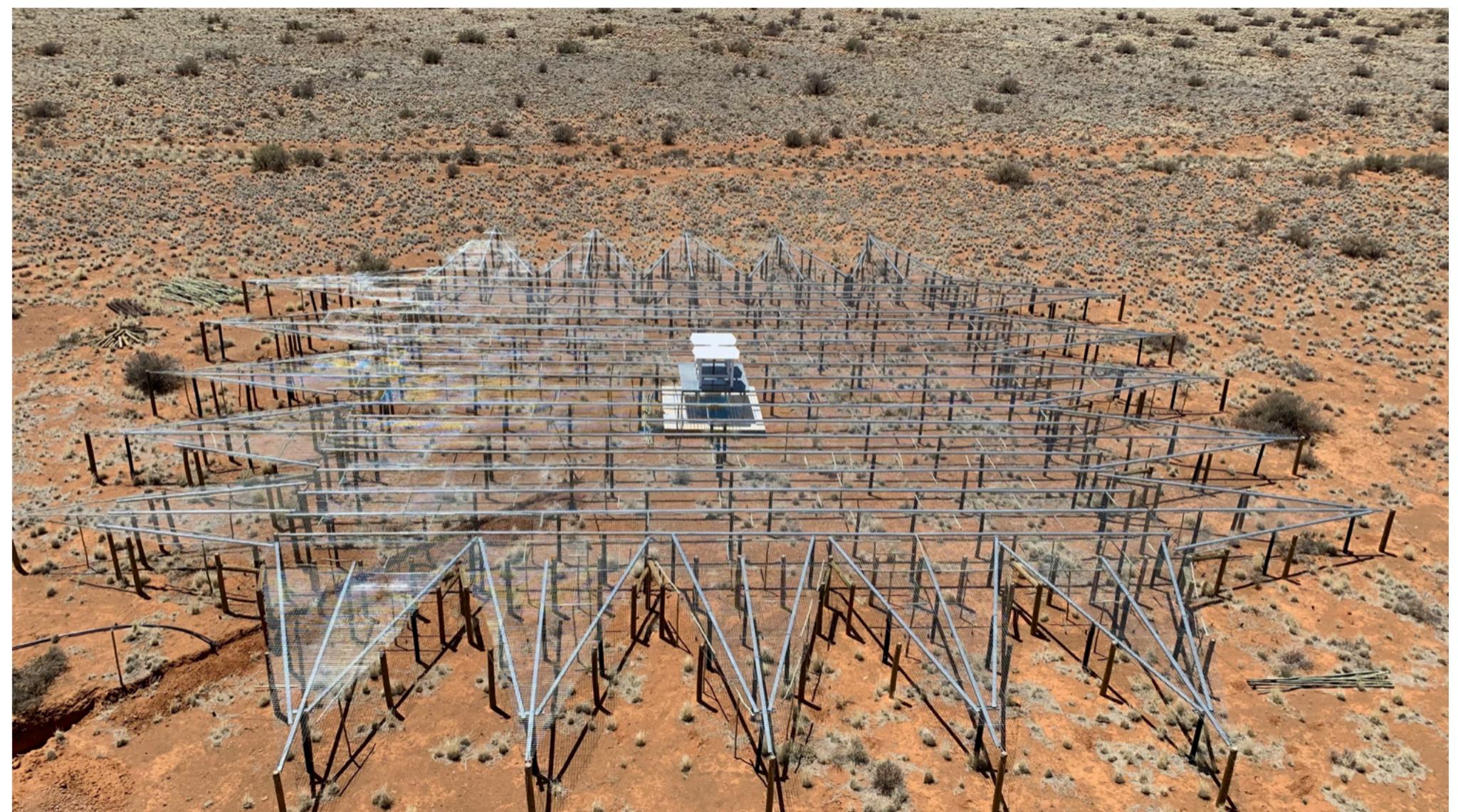
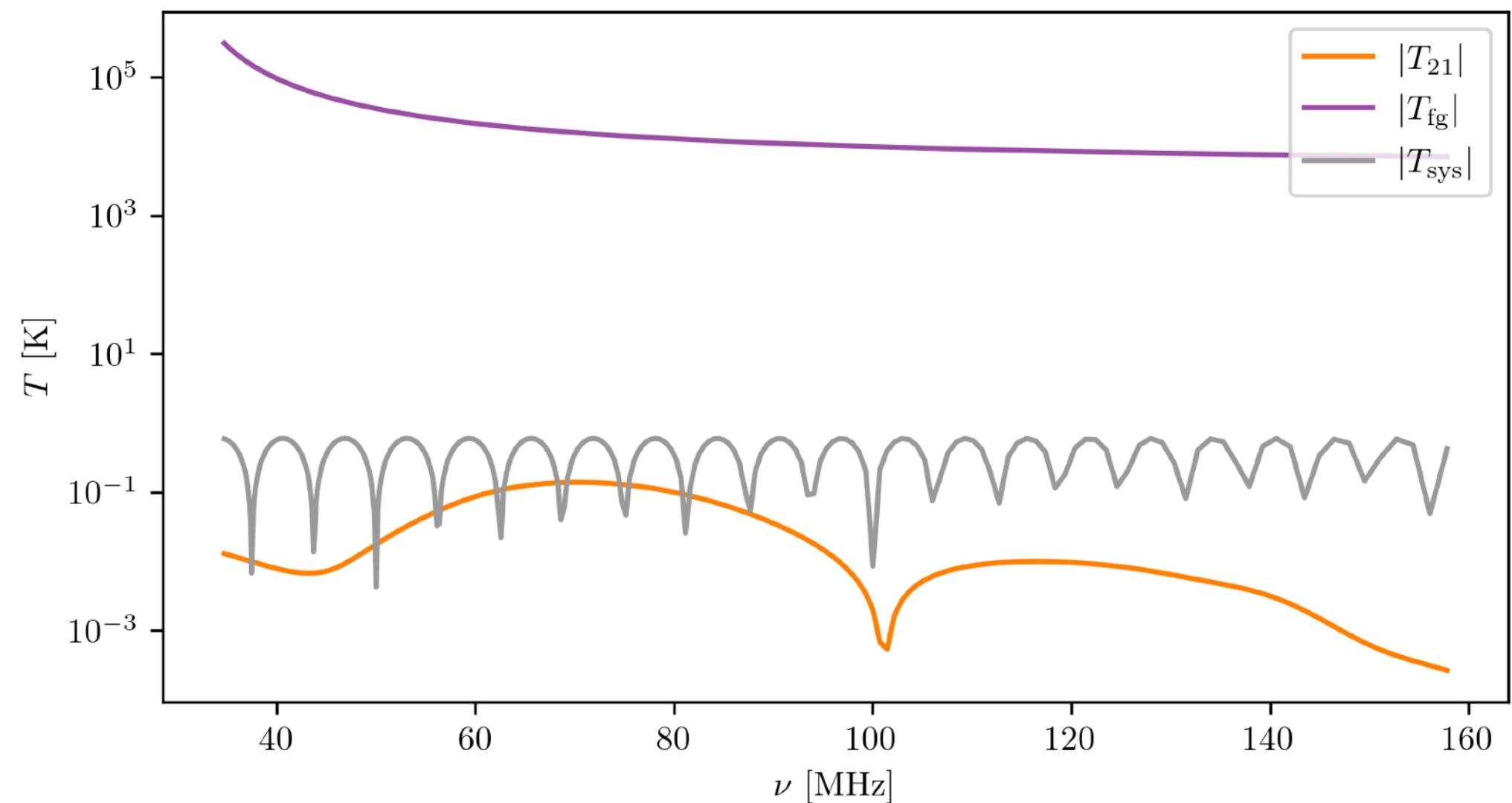
- Parameter estimation via the posterior
 $P(\Theta | D, M)$
- Model comparisons via the evidence
 $Z = P(D | M)$
- Tension quantification also with the evidence

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



Modelling our data

- Building models for our data is key to advancing our understanding
- Often however we also have to model our instrument and contaminating signals
- In 21cm Cosmology this amounts to instrument calibration, foreground modelling and modelling of the environment
- These contaminants are known as ‘nuisance’ components and are modelled with ‘nuisance’ parameters



The issue

- Our model is composed of many components

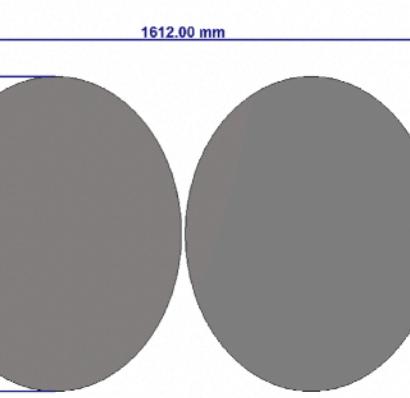
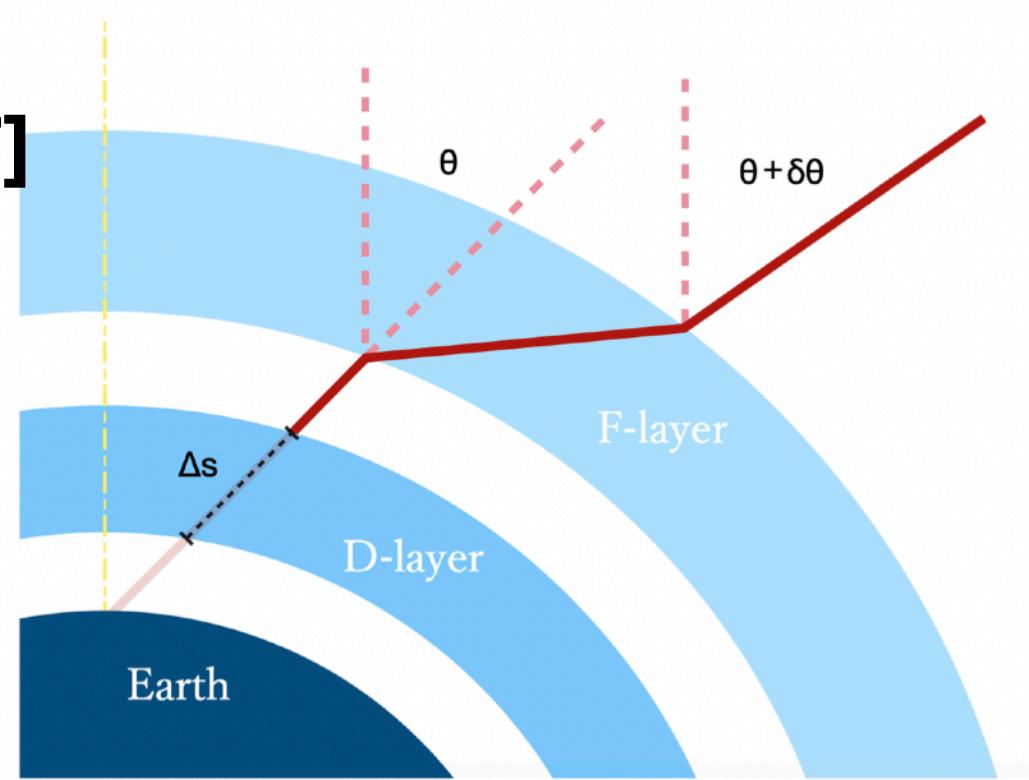
$$M(\Theta) = M_{T21}(\theta, \alpha_{instrument}) + M_{FG}(\alpha_{FG}, \alpha_{instrument}) \\ + M_{Env}(\alpha_{Env}, \alpha_{instrument})$$

- We can see clearly that

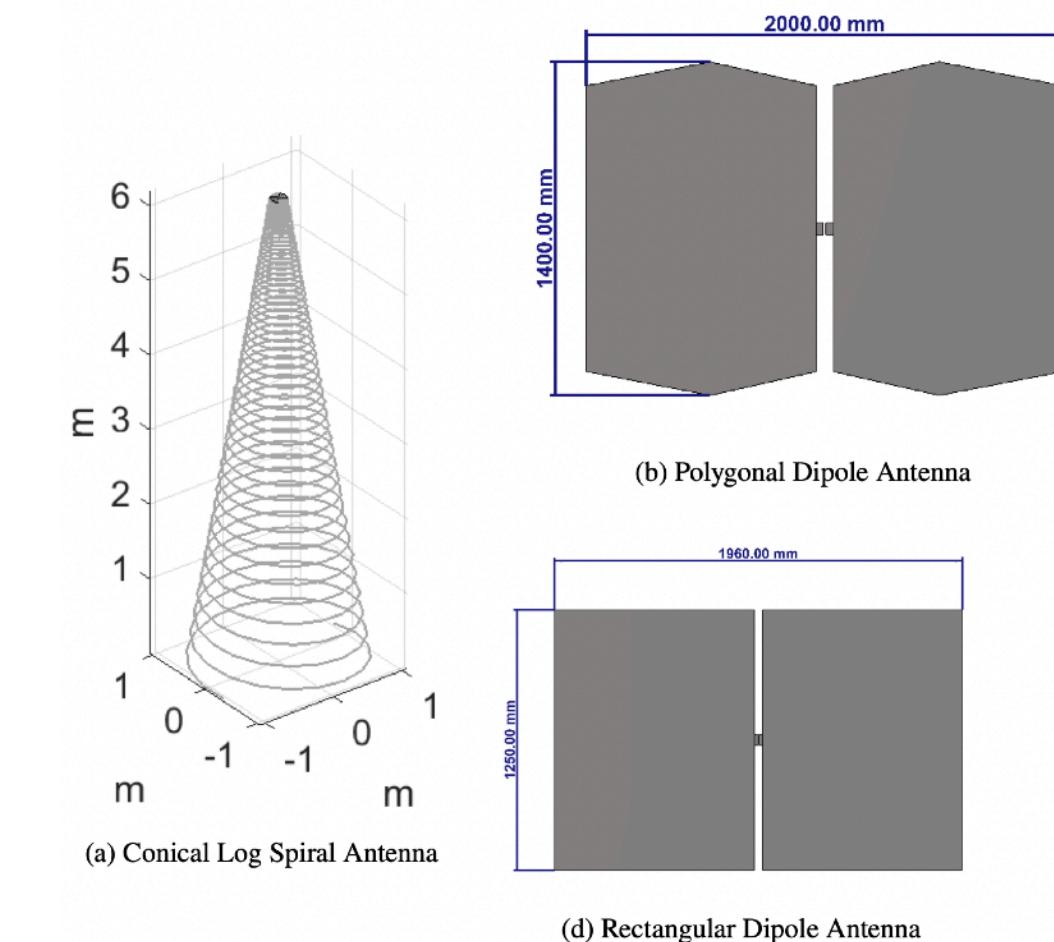
$$\Theta = \{\theta, \alpha_{FG}, \alpha_{instrument}, \alpha_{Env}\} = \{\theta, \alpha\}$$

- Often the number of nuisance parameters far exceeds the number of signal parameters
- Makes doing inference, doing joint inference and understanding our results hard!

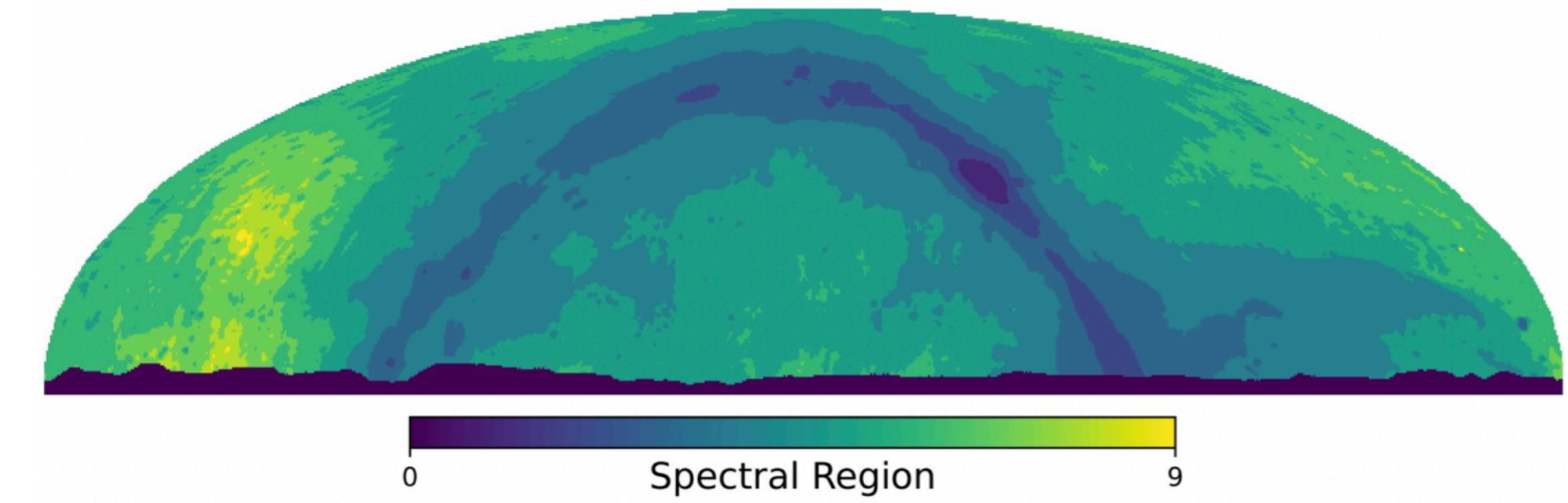
Shen+ [2011.10517]



Cumner+ [2106.10193]



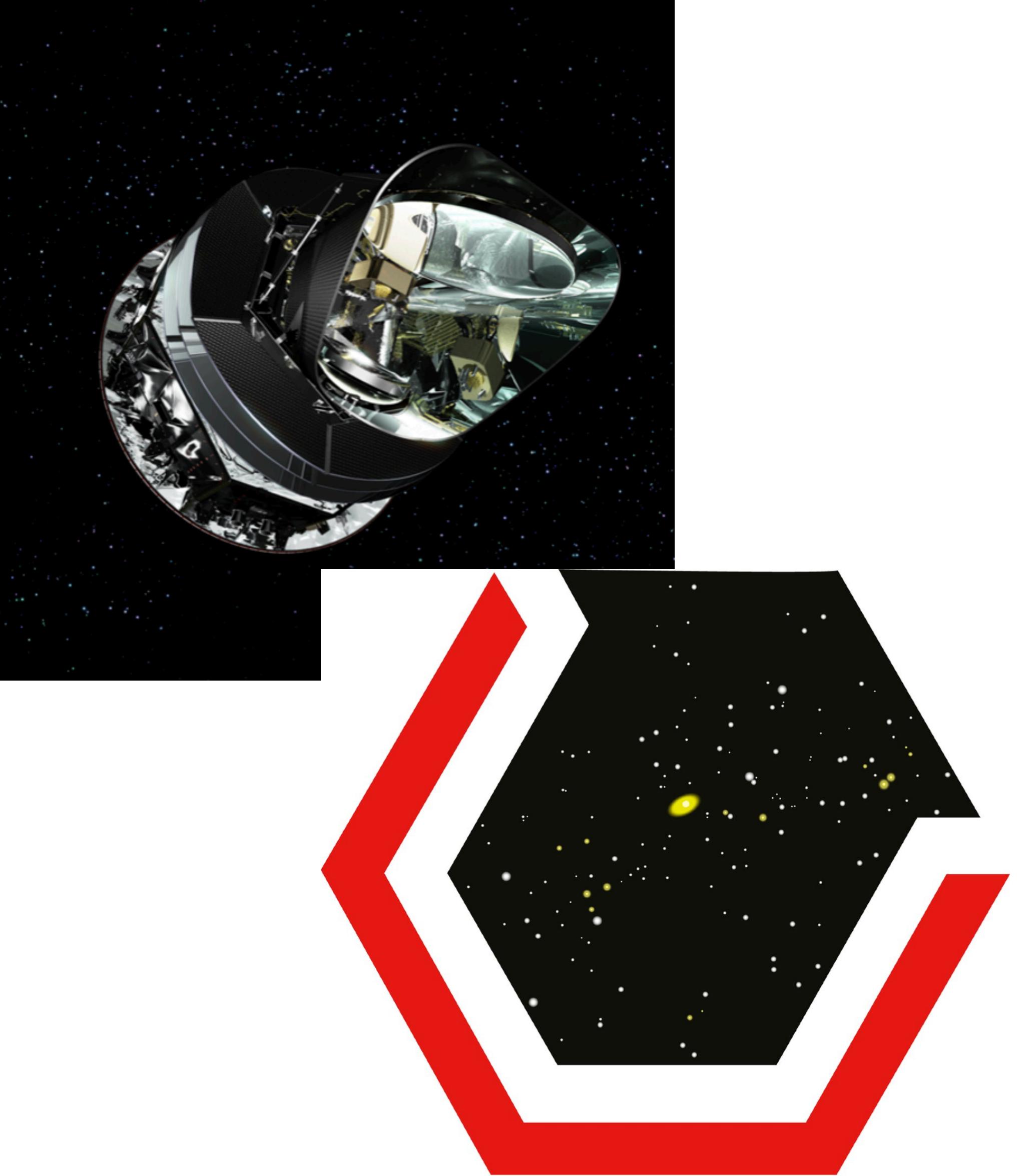
Coarse-Grained Spectral Index Map



Pattison+ [2307.02908]

The issue: Joint analysis

- For example both Planck and DES have 15 and 20 nuisance parameters
- There are only 6 cosmological parameters
- For a combined Planck and DES analysis we have to sample around 41 parameters
- The scaling of a nested sampling algorithm like polychord goes as d^3
- Could we find a way to sample just the 6 cosmological parameters in our joint analysis and recover the same results?



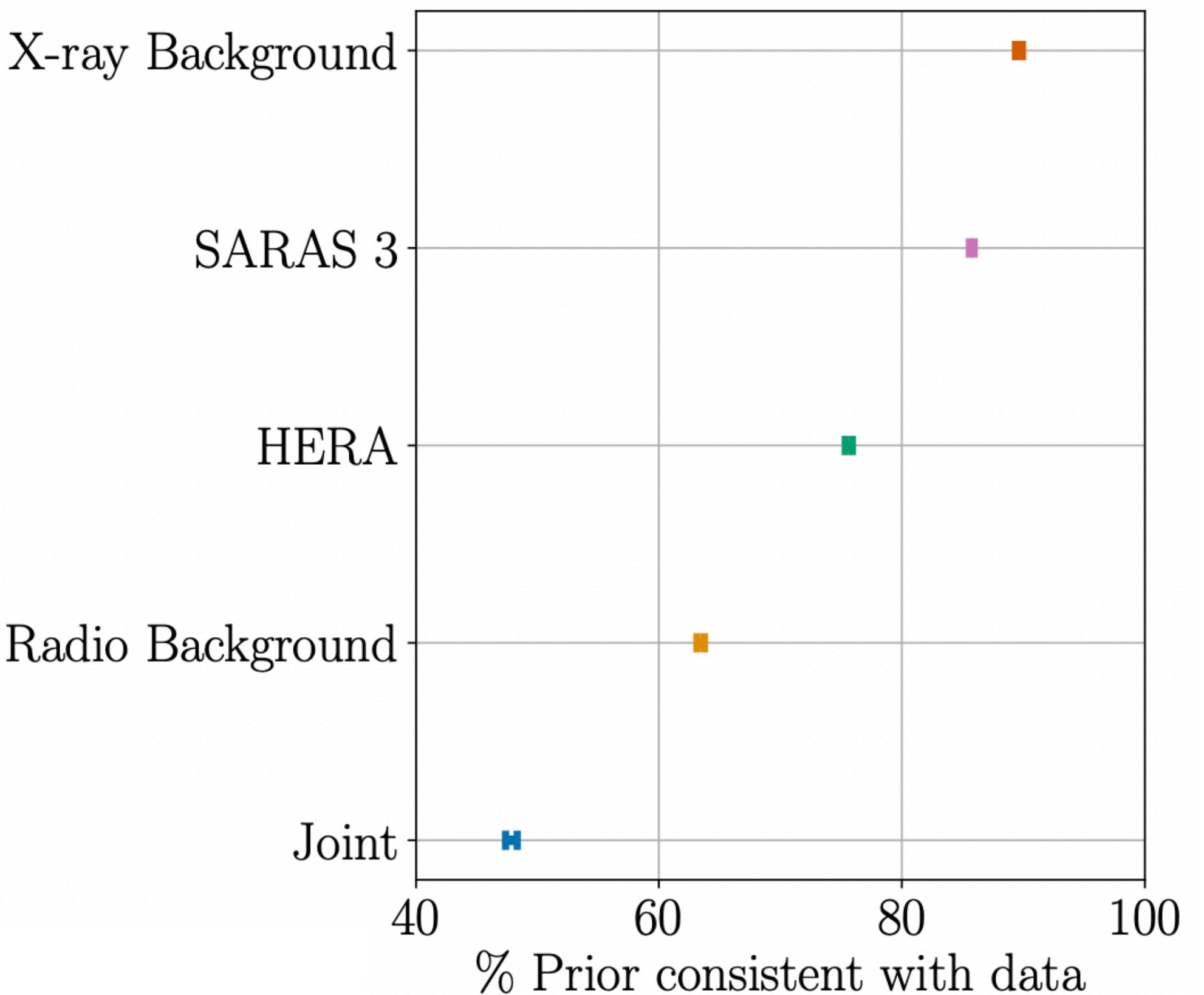
THE DARK
ENERGY SURVEY

The issue: Constraining power?

- We might also want to ask whether Planck gives us more information about our 6 cosmological parameters than DES
- We can do this with the KL divergence

$$D_{KL}(P \parallel \pi) = \int P(\theta, \alpha \mid D, M) \log \frac{P(\theta, \alpha \mid D, M)}{\pi(\theta, \alpha \mid M)} d\Theta$$

- This isn't quite what we want!
- Can we find a way to calculate the KL divergence in just the signal parameter space?

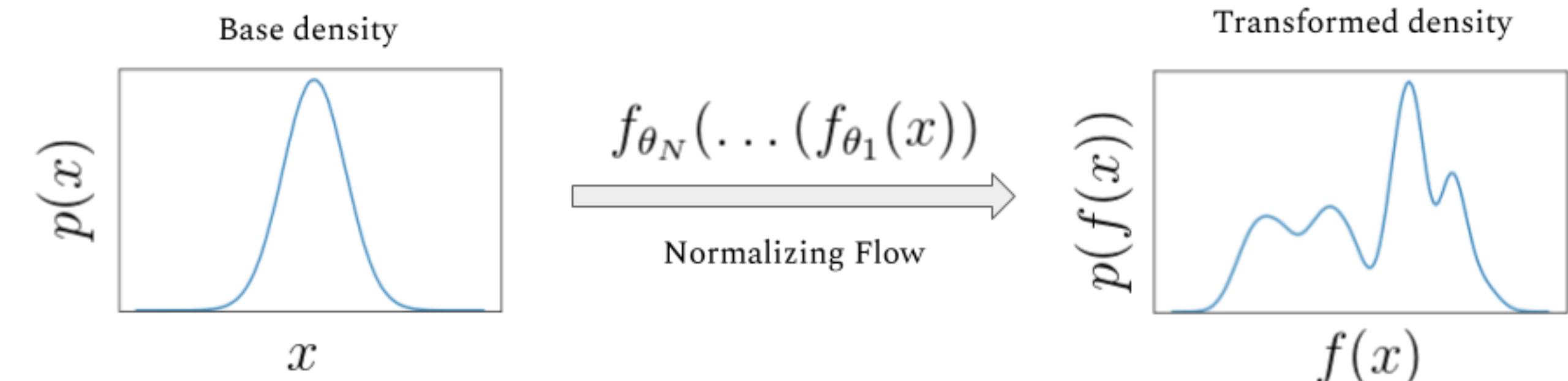
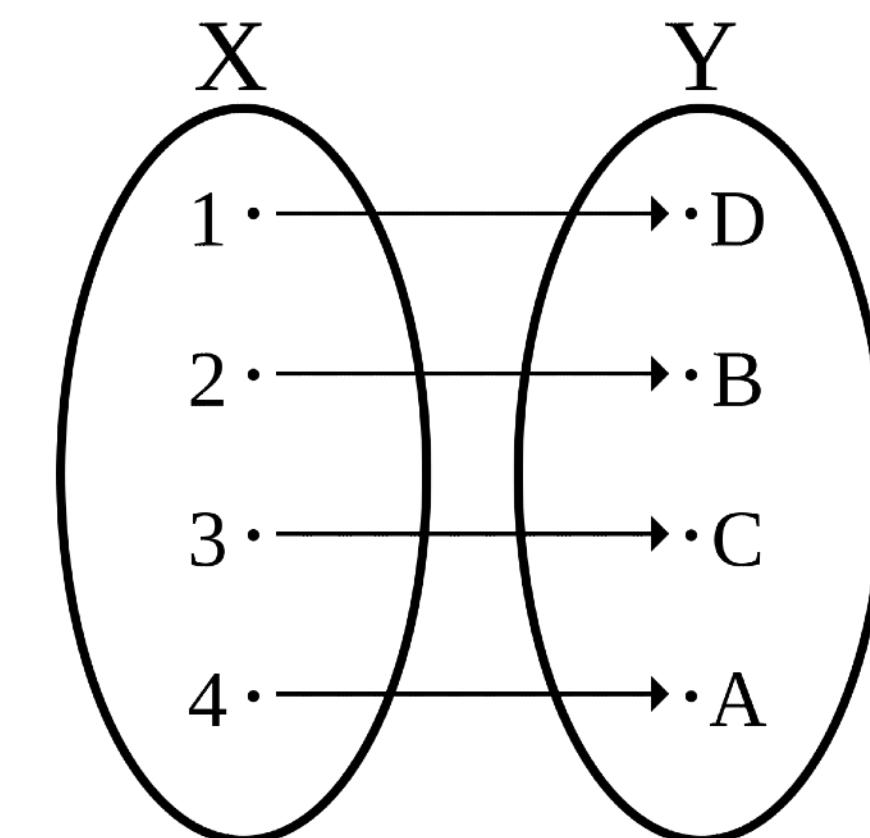


Pochinda+ [2312.08095]

Normalising Flows

Yes with Normalising Flows

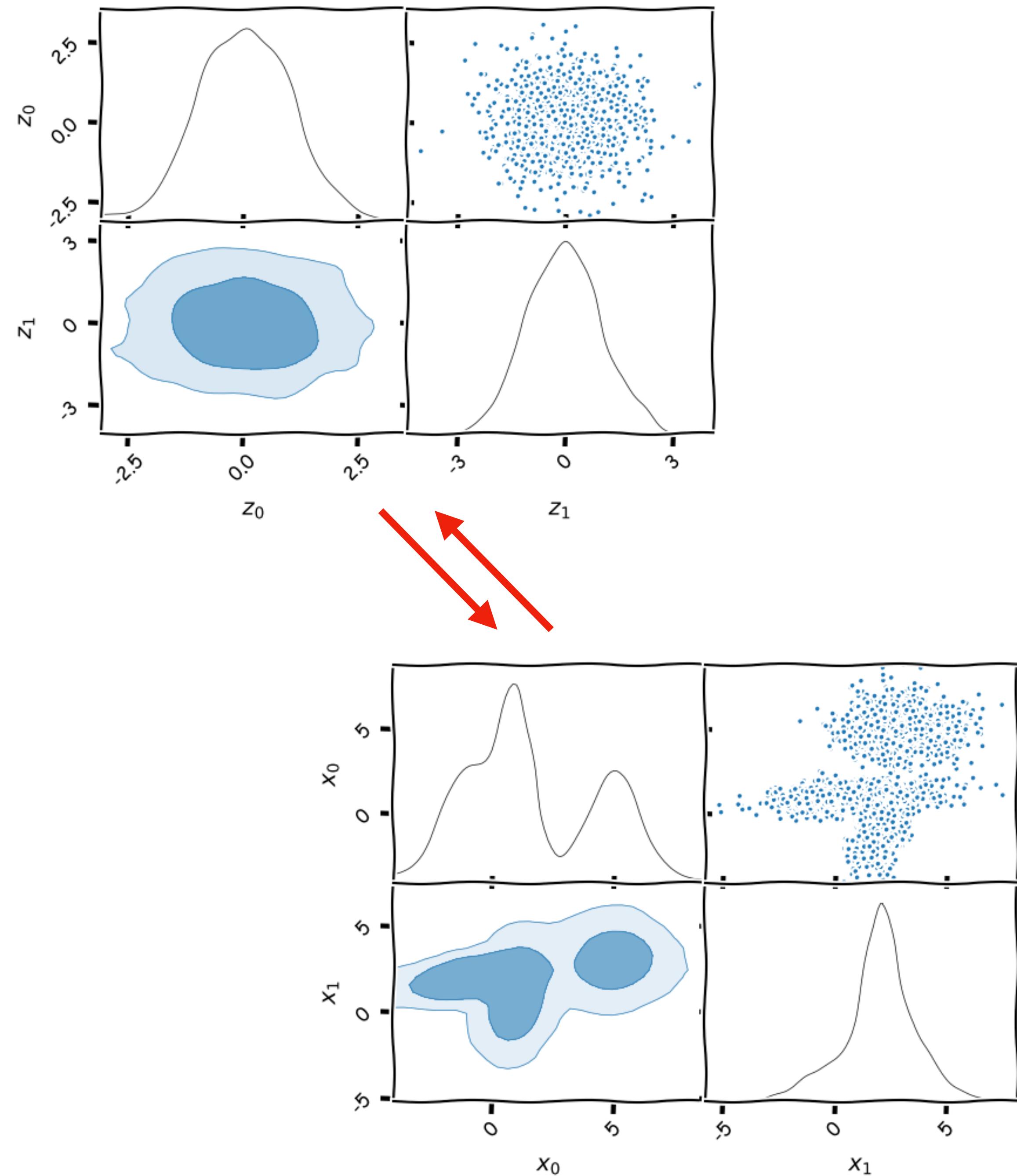
- We can do both these things with normalising flows (NFs) and more
- NFs are a neural density estimation tool
- Given samples on $P(\theta, \alpha | D, M)$ we can use NFs to marginalise out α and assess the analytically intractable $P(\theta | D, M)$
- These are neural networks that parameterise a bijective transformation from one distribution to another



But what does that mean?

- We learn a transformation f_ϕ from samples on a known distribution $z \sim \mathcal{N}(z; \mu, \sigma)$ to samples from a more complex target $x \sim P(x)$
- And approximate the probability on the target as

$$P_\phi(x) = \mathcal{N}(f_\phi^{-1}(x); \mu, \sigma) \left| \frac{df_\phi^{-1}(x)}{dx} \right|$$



Masked Autoregressive Flow

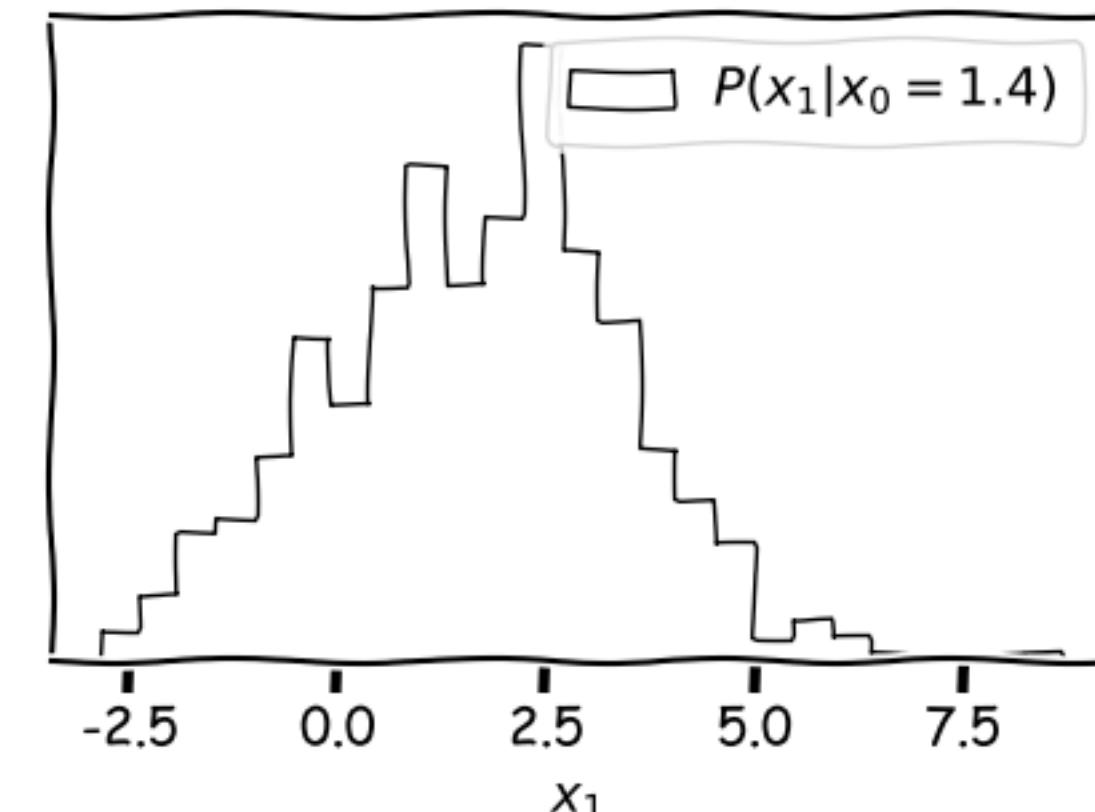
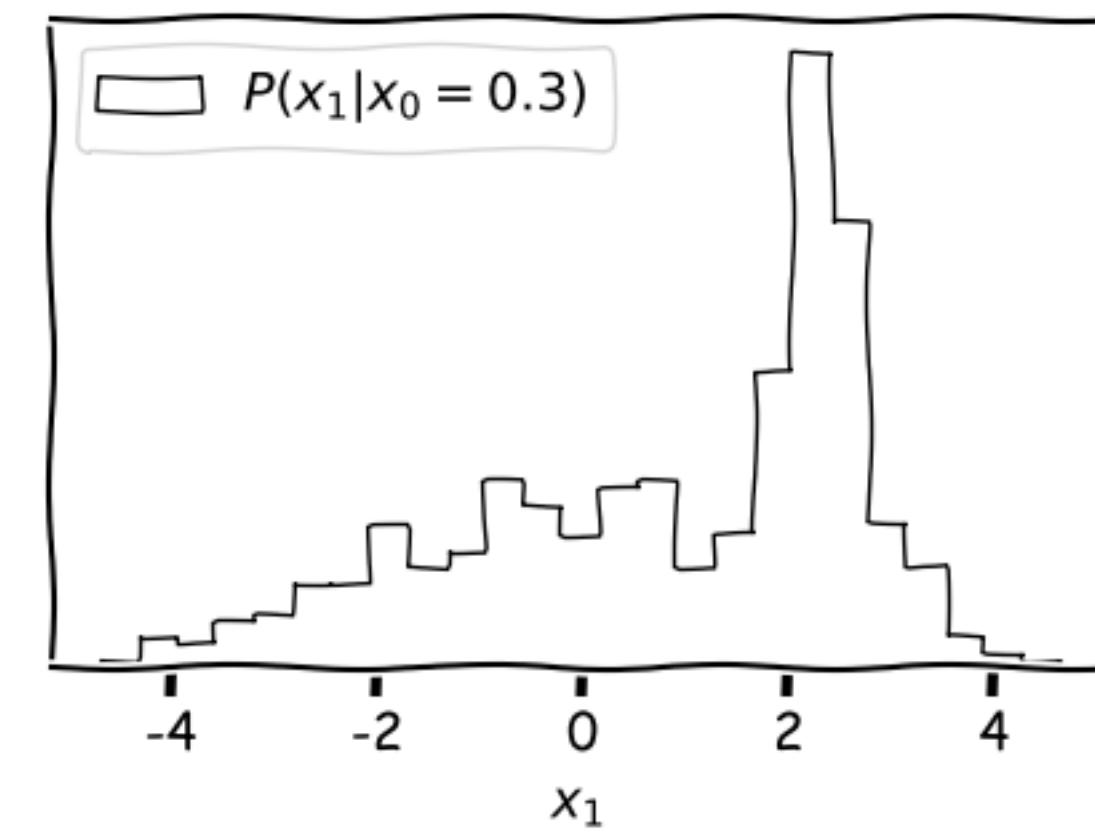
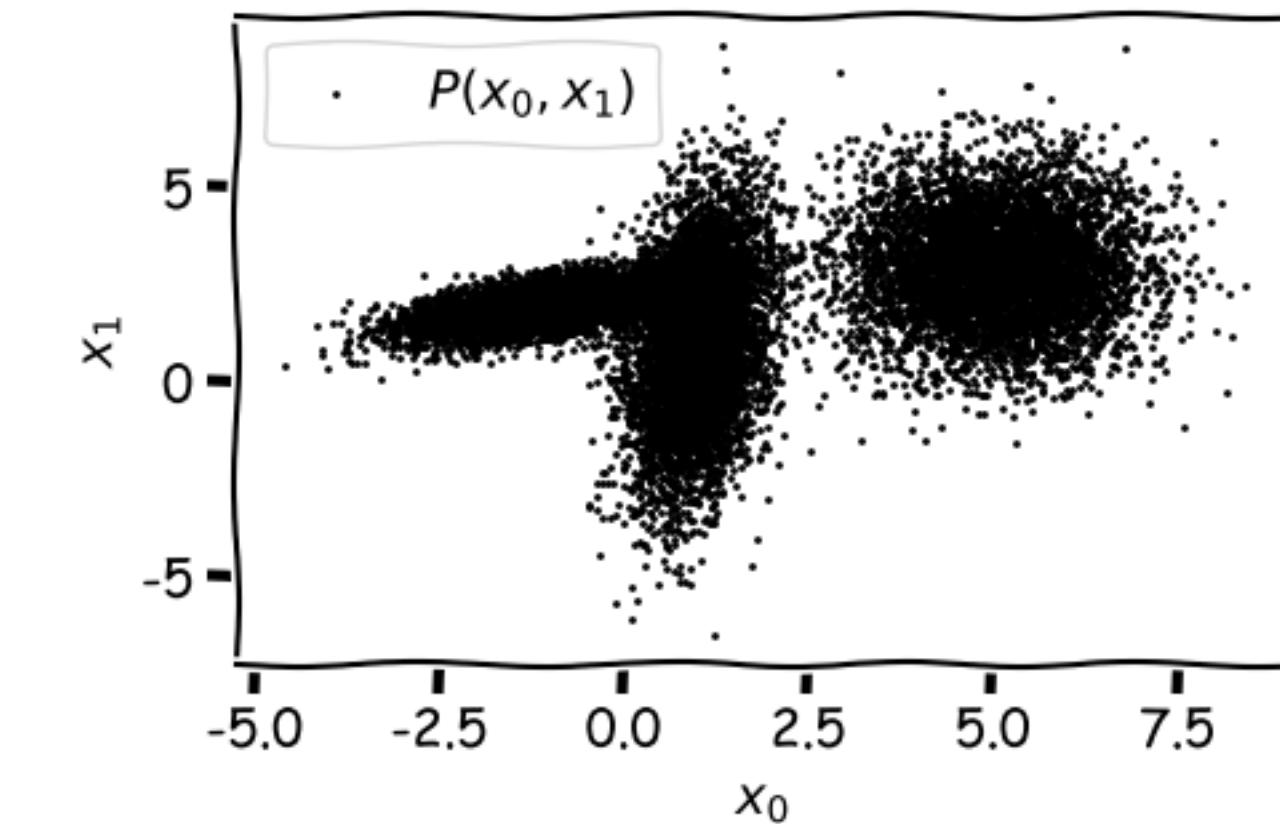
- Parameterise the target distribution as the product of a series of conditional distributions

$$P(x) = \prod_i^{N_{dims}} P(x_i | x_j \in N_{dims} \neq i)$$

- And model each conditional as a Gaussian

$$P(x_i | x_j \in N_{dims} \neq i) = \mathcal{N}(\mu_i, \sigma_i)$$

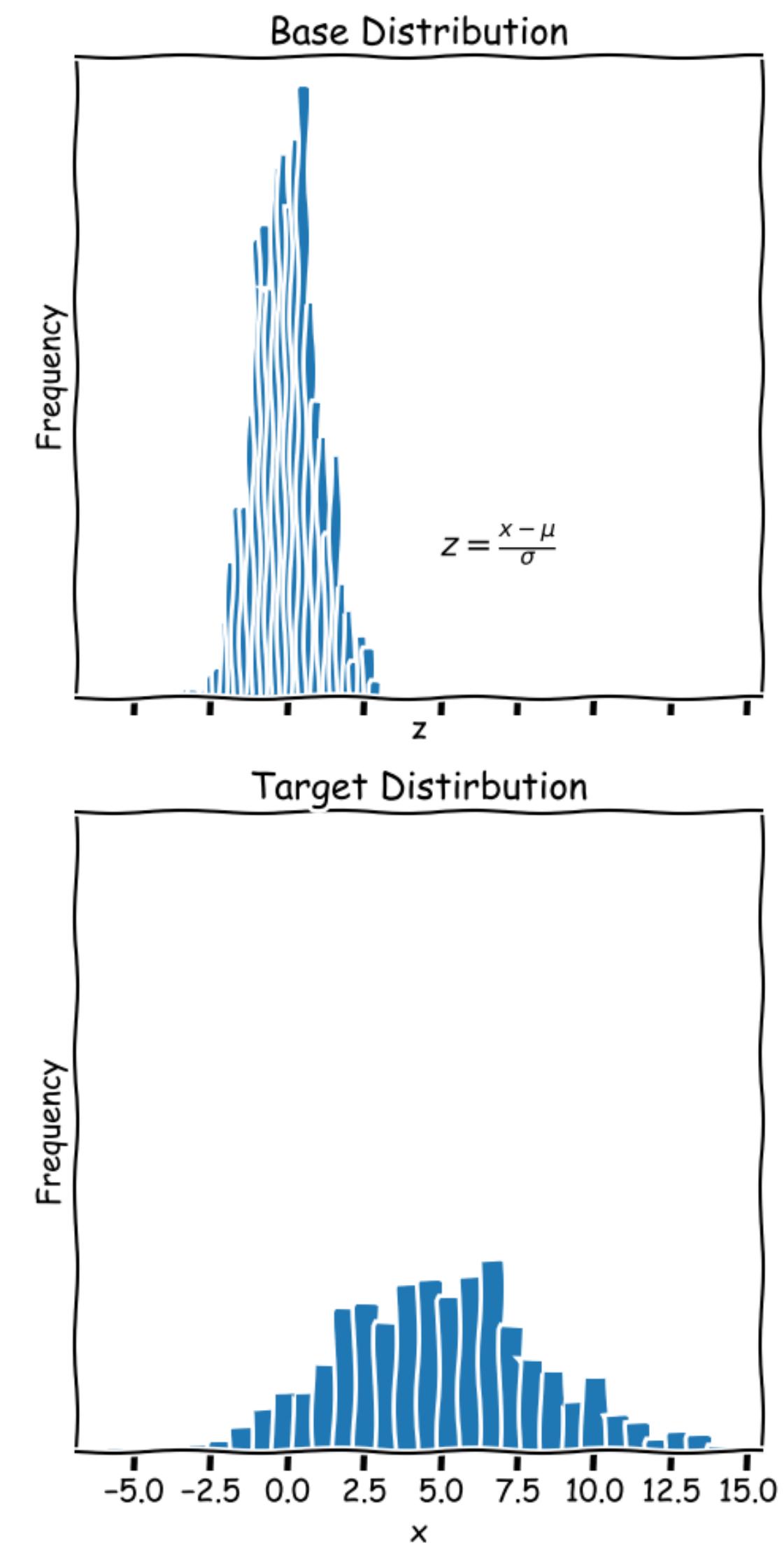
Where $\mu_i(x_j \in N_{dims} \neq i)$ and $\sigma_i(x_j \in N_{dims} \neq i)$



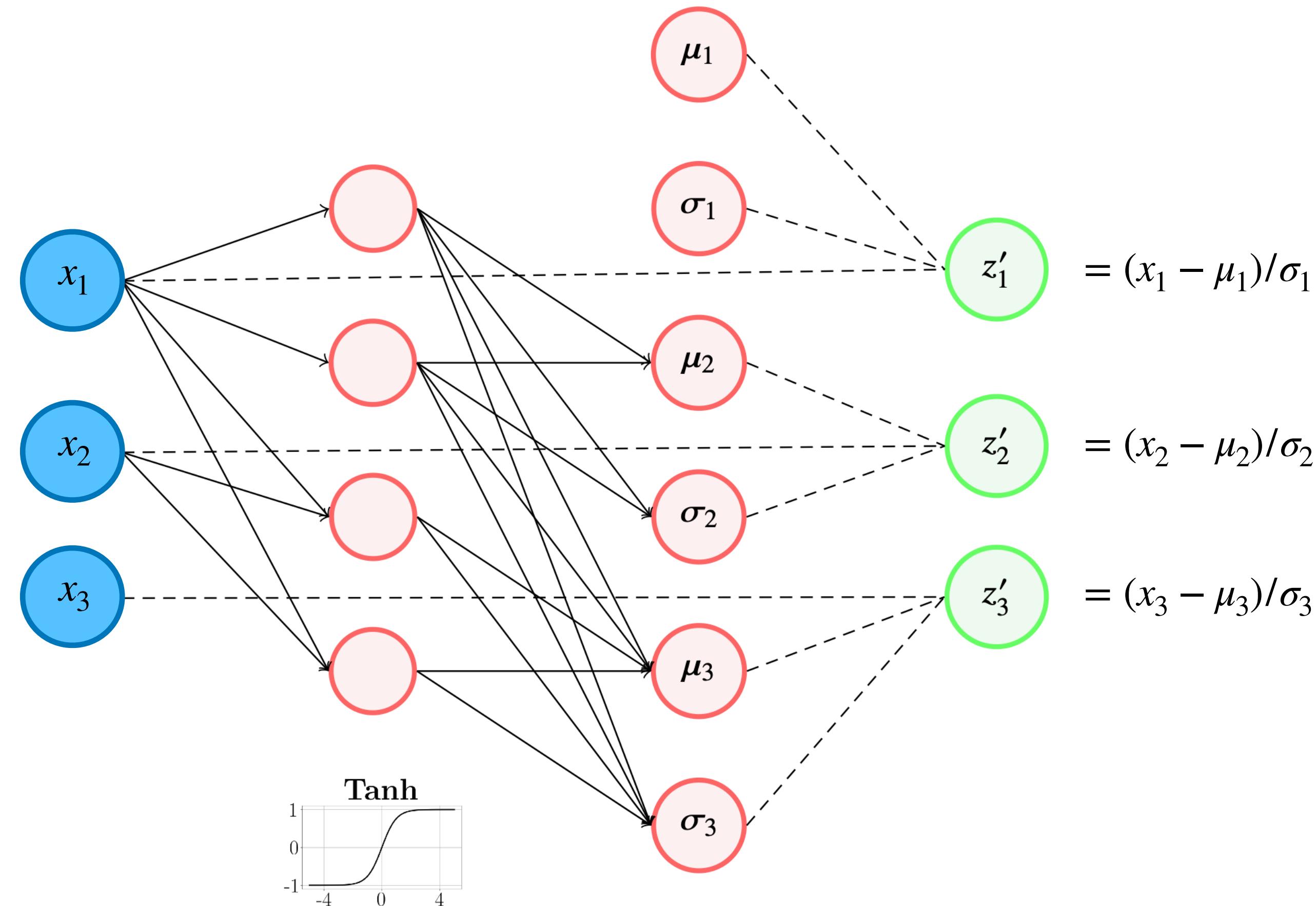
Masked Autoregressive Flow

- Conditionality appears as masking some of the connections in the network
- And for each dimension the network outputs a mean and standard deviation that transform the samples back on to the standard normal base

$$z = \frac{x - \mu}{\sigma}$$

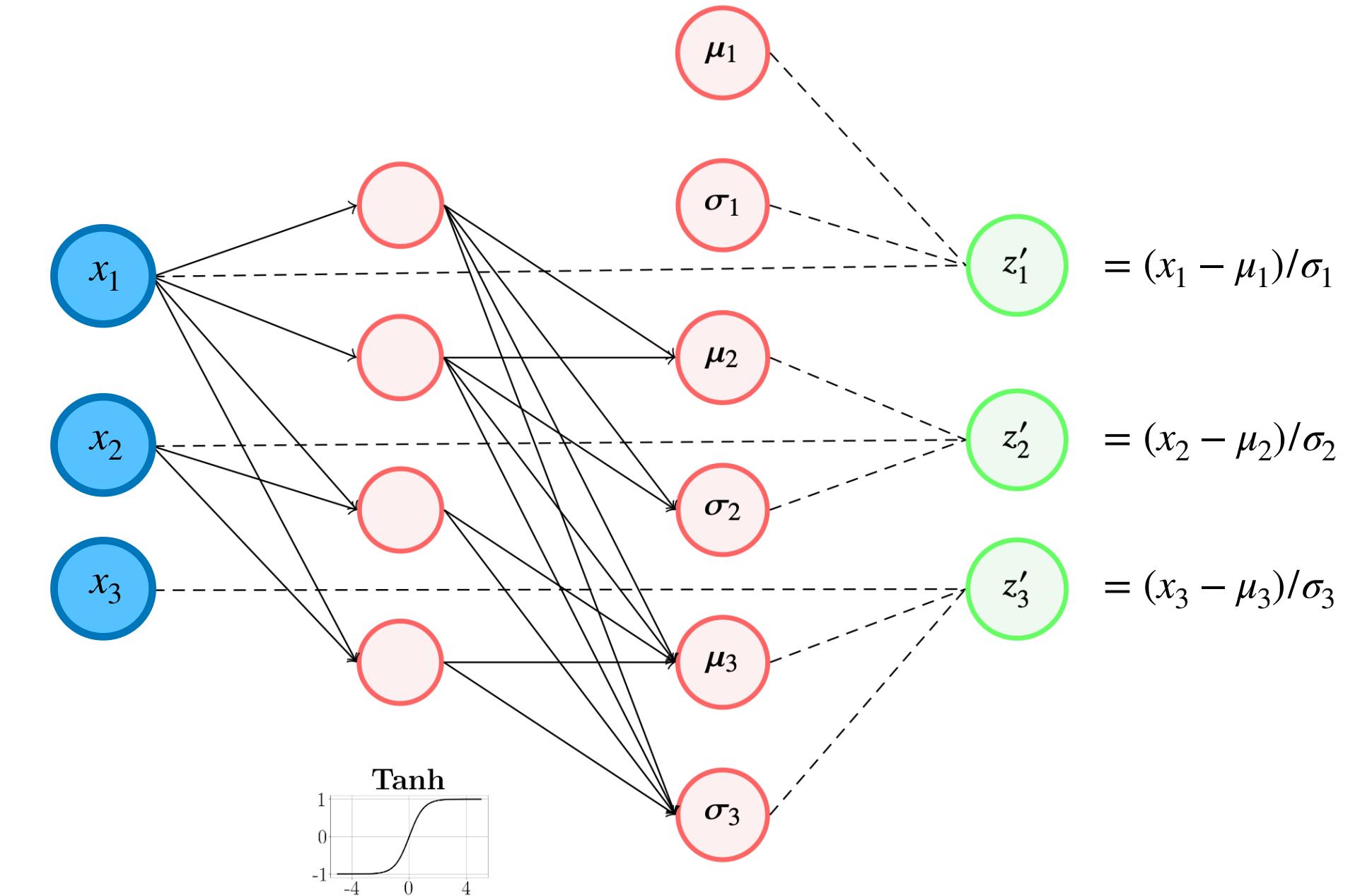


Masked Autoregressive Flow



Masked Autoregressive Flow

- Technically this is a Masked Autoencoder for Distribution Estimation (Germain+ [1502.03509])
- A Masked Autoregressive Flow (Papamakarios + [1705.07057]) is a chain of these transformations
- Change the mask in each network
- Encode complex target distributions

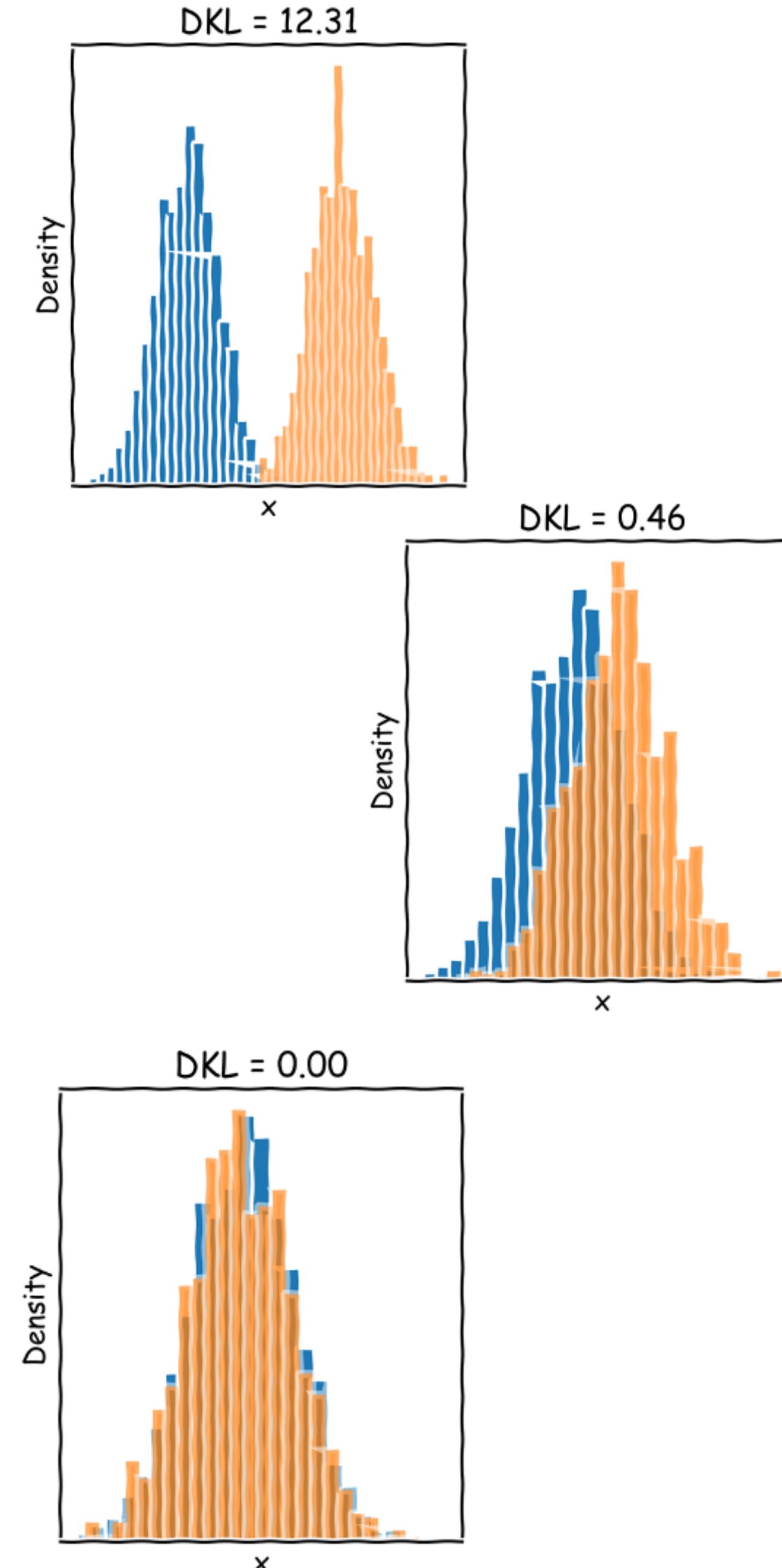


Masked Autoregressive Flow

- We train the network by minimising the KL divergence between the target $P(x)$ and the predicted distribution $P_\phi(x)$

$$D_{KL} = - \mathbb{E}_{P(x)}[\log P_\phi(x)] + \mathbb{E}_{P(x)}[\log P(x)]$$

- We do not know $\mathbb{E}_{P(x)}[\log P(x)]$ but it is independent of ϕ so we can ignore it for an optimisation problem



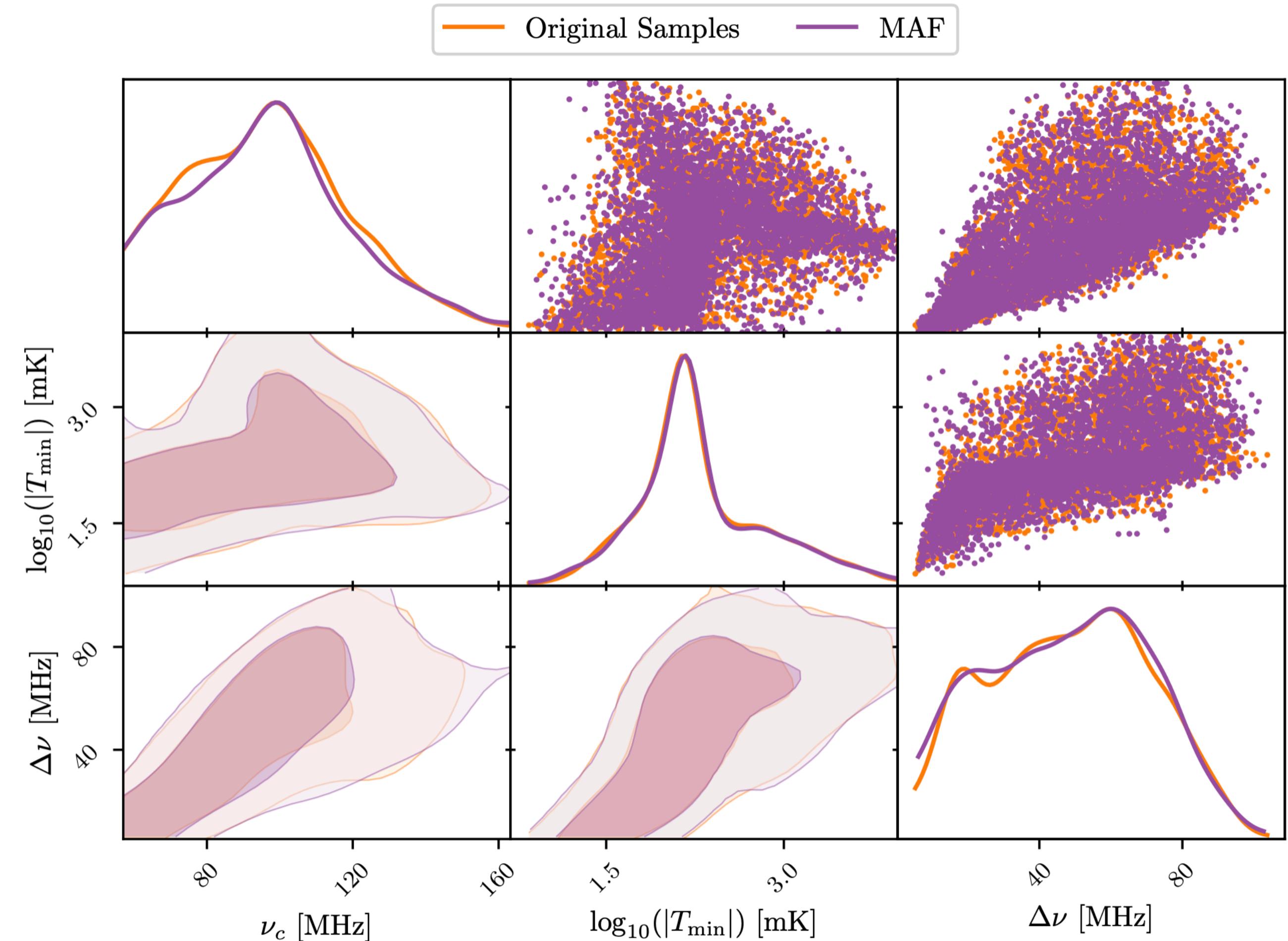
Masked Autoregressive Flow

- With a trained normalising flow we can draw samples from $P(x)$ via

$$x = z\sigma + \mu$$

- And calculate probabilities $P(x)$ via

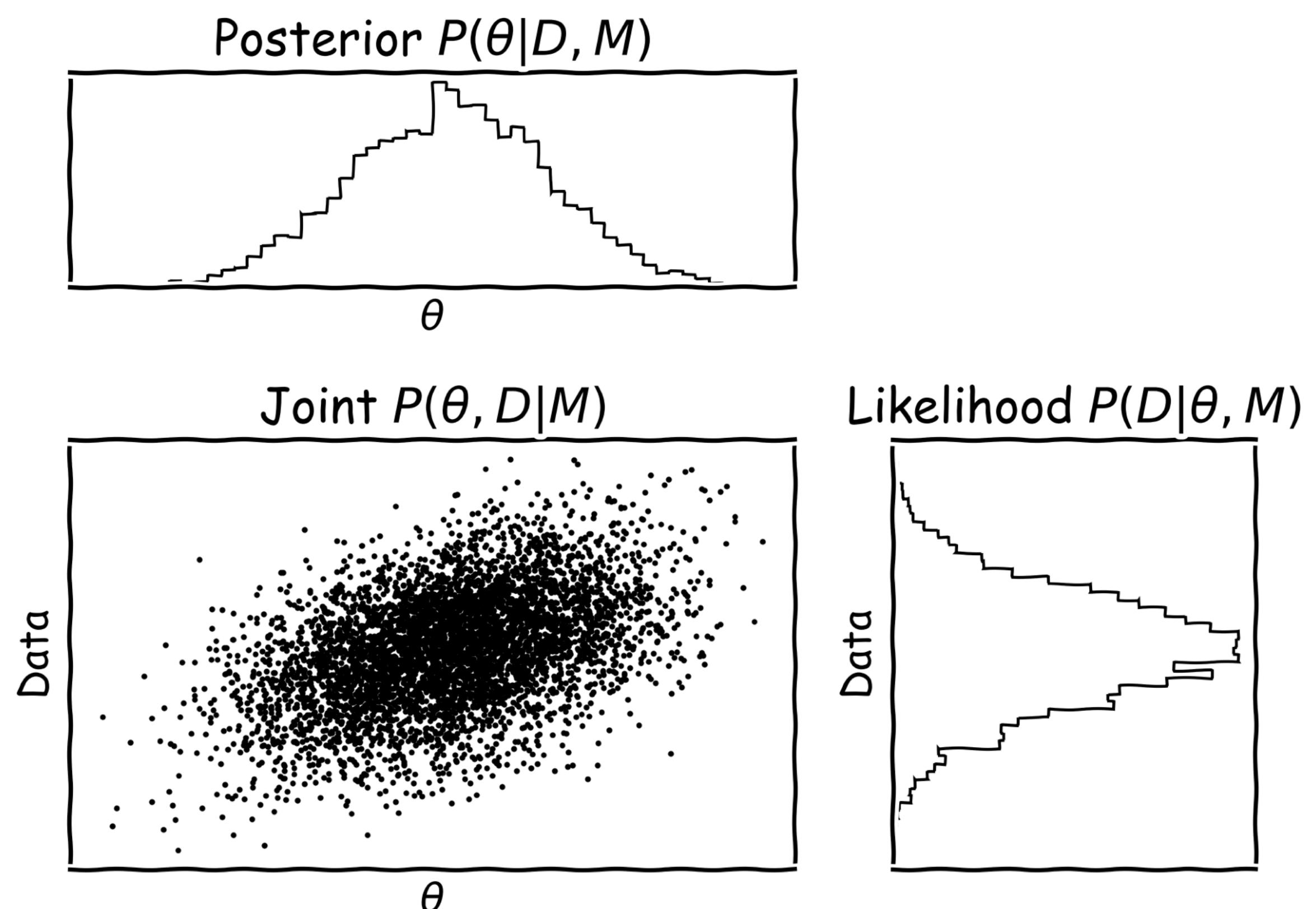
$$P_\phi(x) = \mathcal{N}(f_\phi^{-1}(x); \mu, \sigma) \left| \frac{df_\phi^{-1}(x)}{dx} \right|$$



Marginal Bayesian Workflows

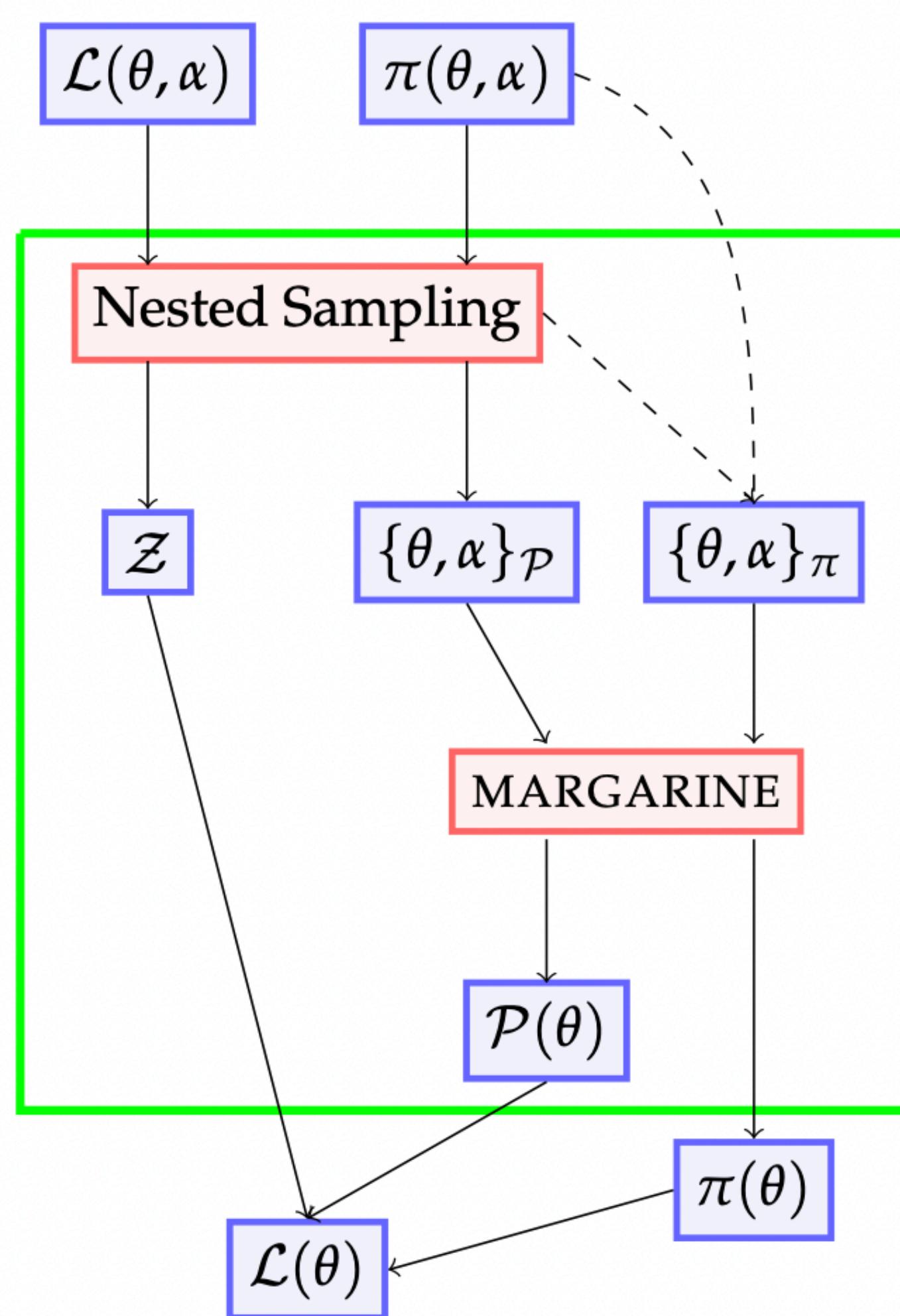
So how do we utilise normalising flows

- Up till now the discussion surrounding NFs has been quite general
- Focused on learning general densities $P(x)$
- But in Bayesian inference the distributions of interest are $P(\theta, \alpha | D, M)$ and $P(D | \theta, \alpha, M)$
- Or more specifically we want $P(\theta | D, M)$ and $P(D | \theta, M)$



So how do we utilise normalising flows

- We can use NFs to learn these distributions and implicitly marginalise out α by passing only samples from θ during training
- Since these models are generative they are effectively posterior and likelihood emulators
- And there are a number of things we can do with them

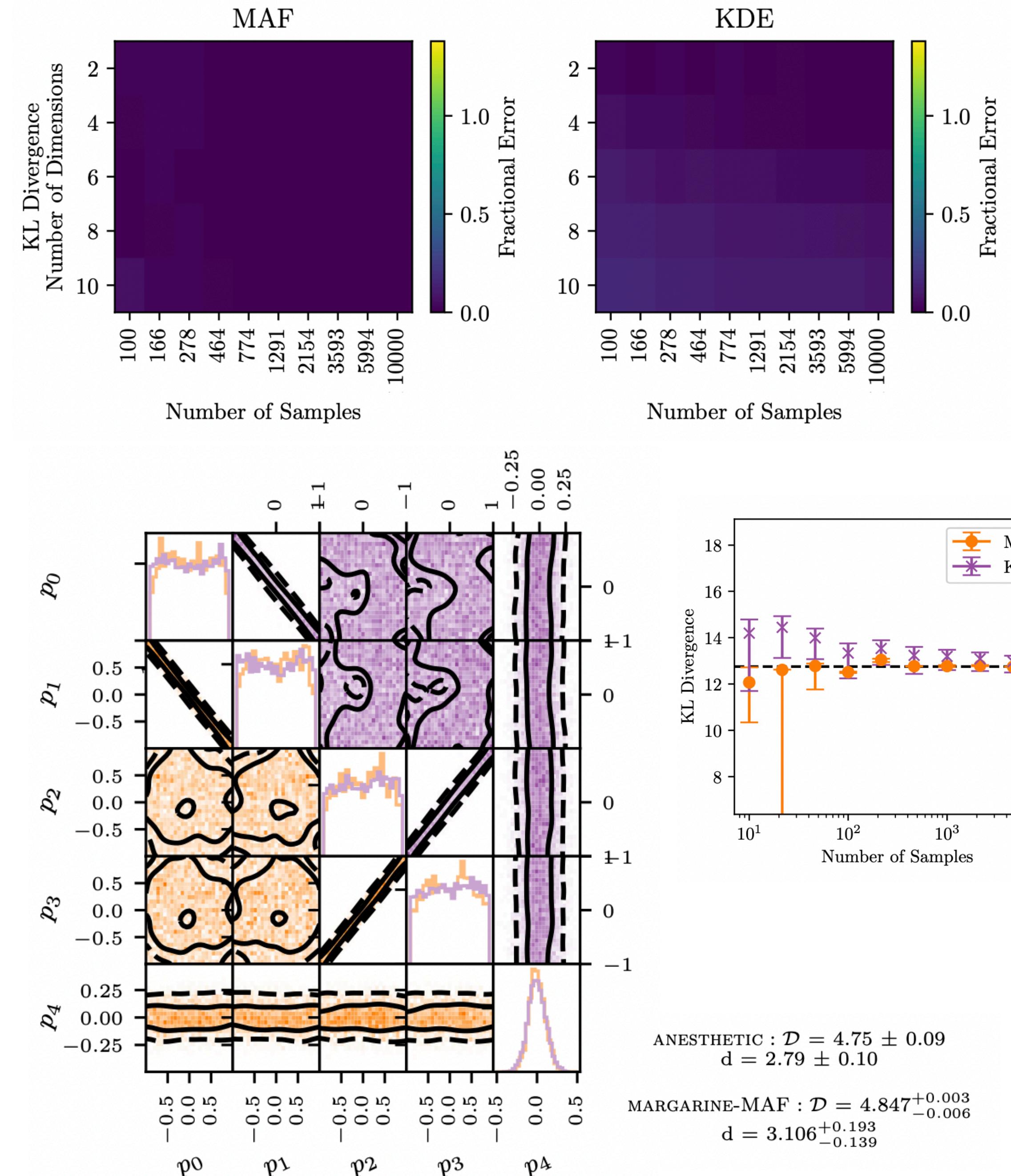


Marginal Statistics

- Firstly they give us access to analytically intractable quantities like $\log P(\theta | D, M)$
- This means we can calculate “marginal” or “nuisance-free” KL divergences

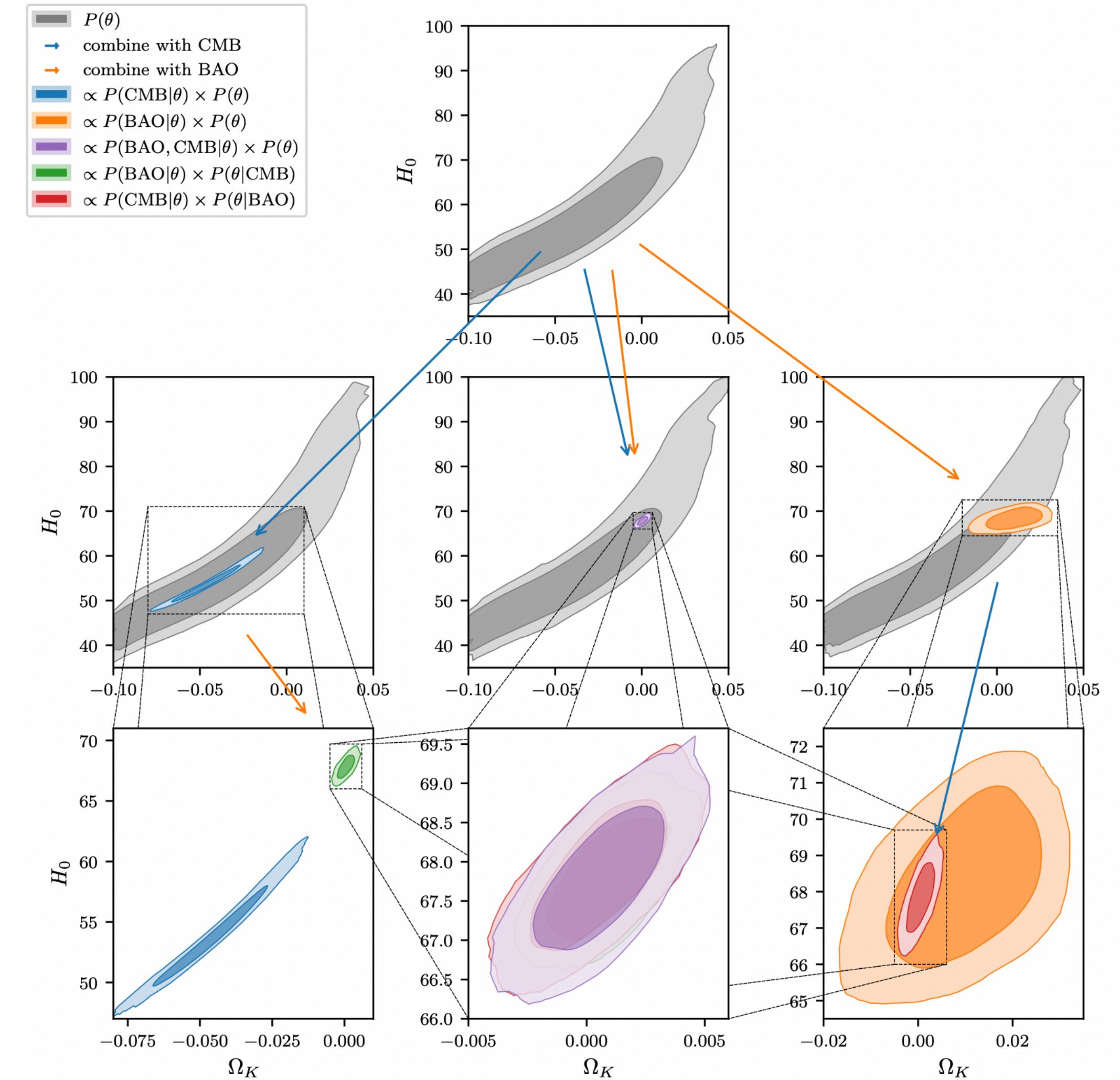
$$D_{KL}(P || \pi) = \int P(\theta | D, M) \log \frac{P(\theta | D, M)}{\pi(\theta | M)}$$

- And we can use this to tell us which experimental approaches give us the most information about the signal of interest



Any prior you like? Alsing and Handley [2102.12478]

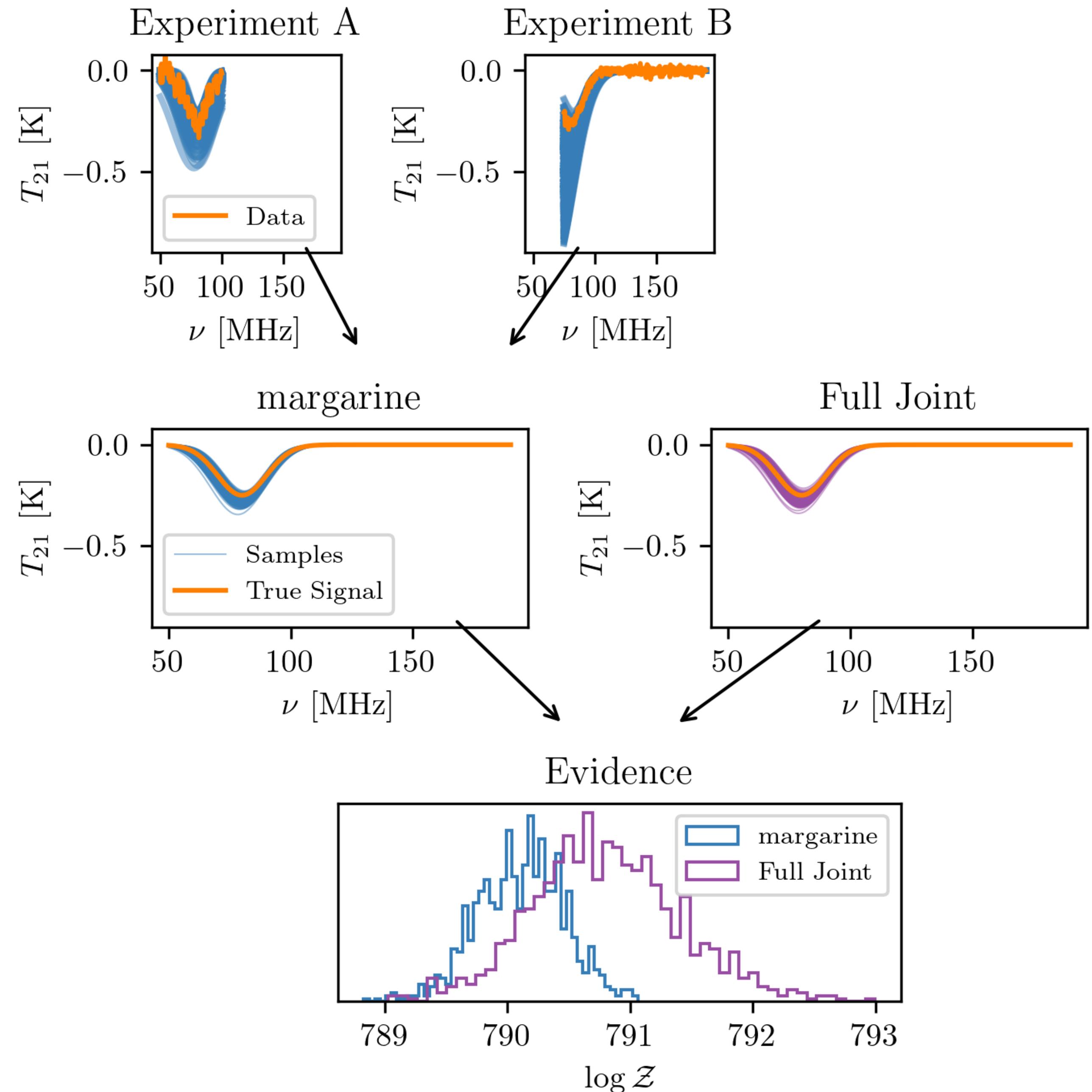
- Nested Sampling involves compressing the prior to the posterior
- If we can move the prior towards the posterior we reduce the run time of the sampling
- This is the premise behind the supernest algorithm (Petrosyan and Handley [2212.01760])
- With normalising flows we can use the posteriors from one analysis as the prior on another or derive more theoretically motivated priors



Marginal Likelihood Functions

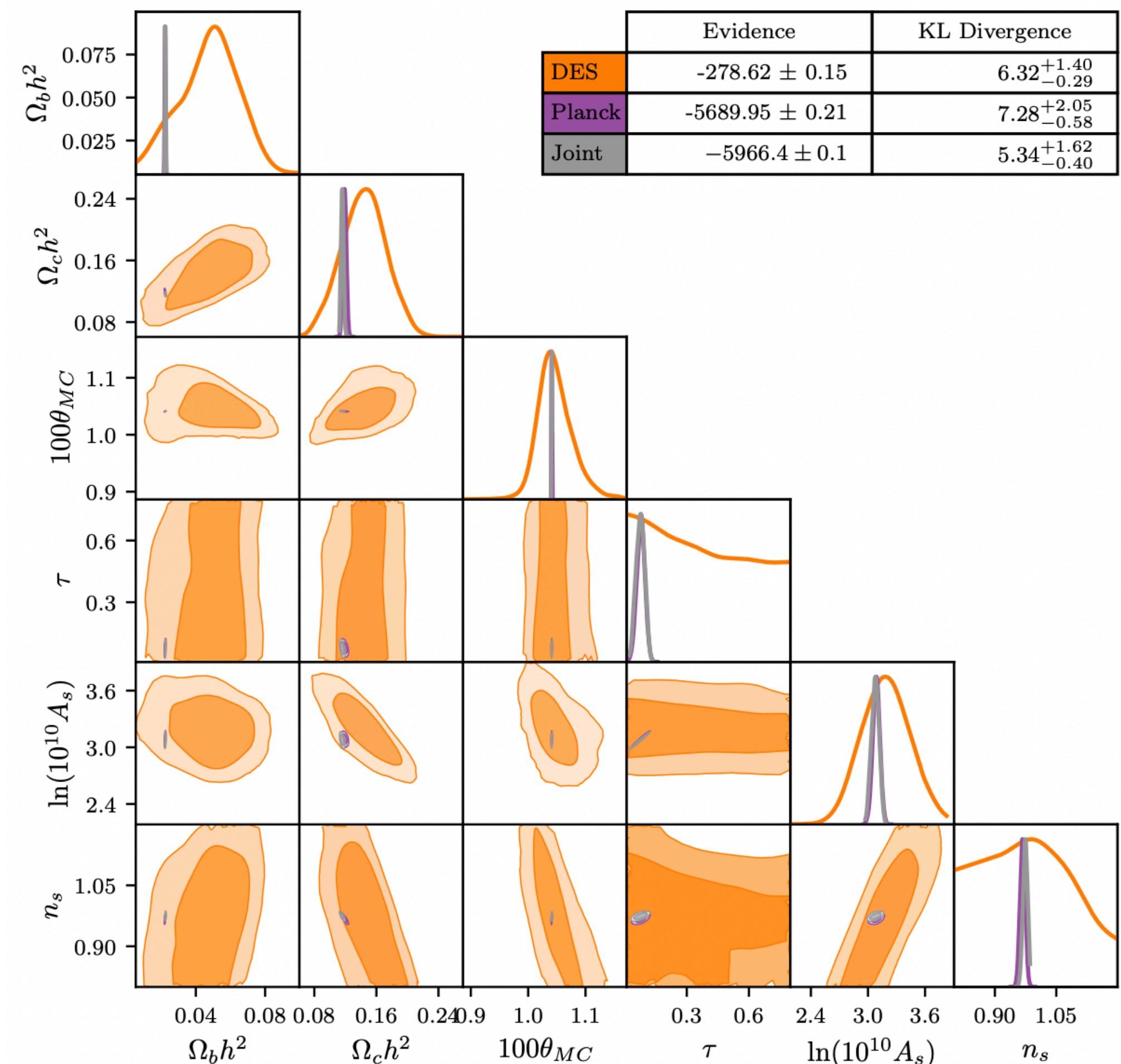
- Finally we can define marginal or nuisance-free likelihood functions
- Essentially this amounts to learning the posterior and prior marginal densities with normalising flows and combining this with the Bayesian evidence

$$L(\theta) = \frac{\int L(\theta, \alpha) \pi(\theta, \alpha) d\alpha}{\int \pi(\theta, \alpha) d\alpha} = \frac{P_\phi(\theta) Z}{\pi_\epsilon(\theta)}$$



Marginal Likelihood Functions

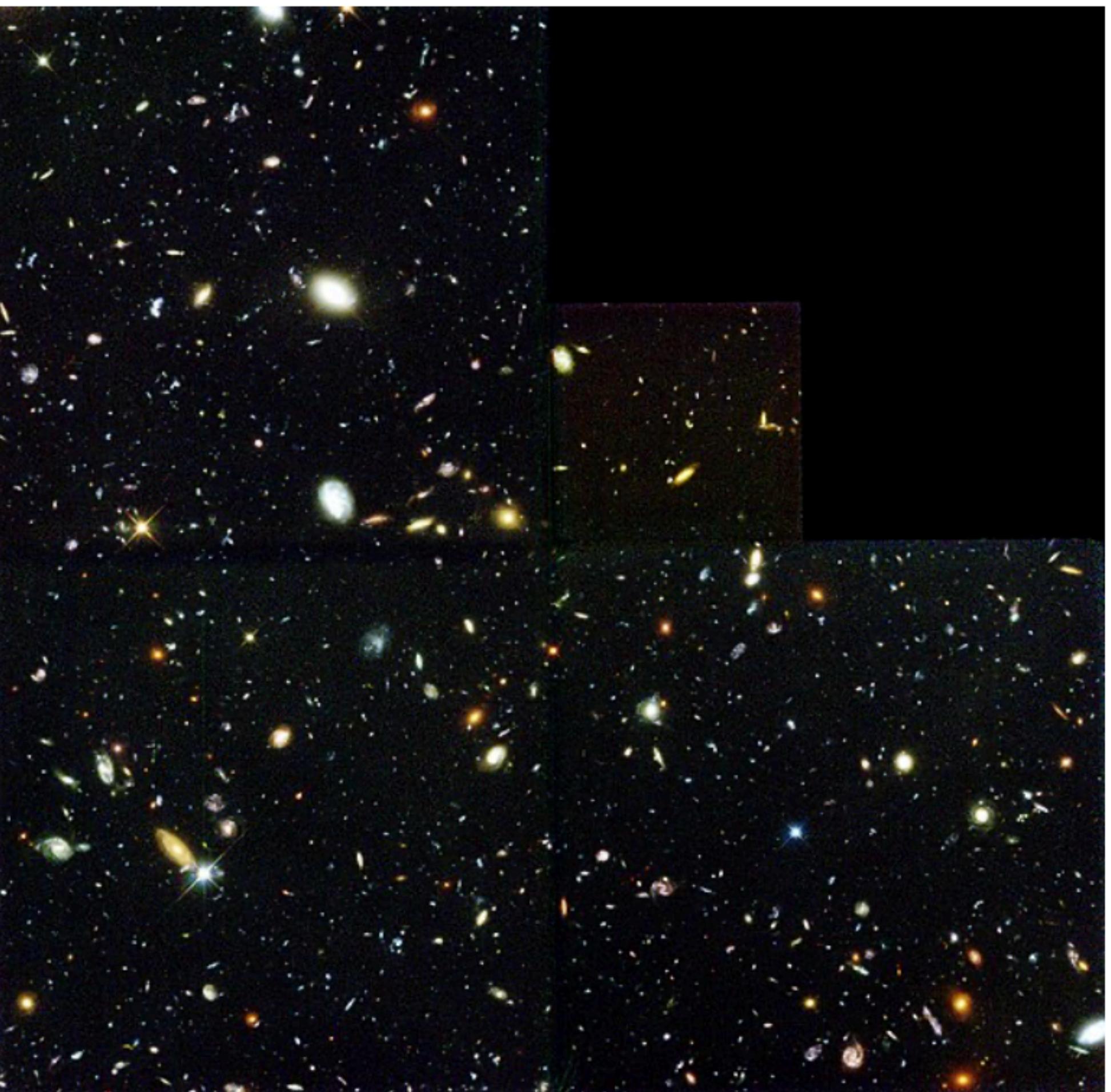
- But why is this important?
- Typically we want to do joint analysis of many different probes
- Using the nuisance-free likelihood function
 - Reduces the number of parameters that need sampling -> reduces the run time (scales as d^3)
 - e.g. for Planck plus DES we go from sampling 41 parameters to $26 + 21 + 6$ and $41^3 > 26^3 + 21^3 + 6^3$
 - Returns the correct Bayesian evidence (that we would have recovered sampling all of the parameters)
 - Correct posterior recovered without double counting priors



And (possible) Applications

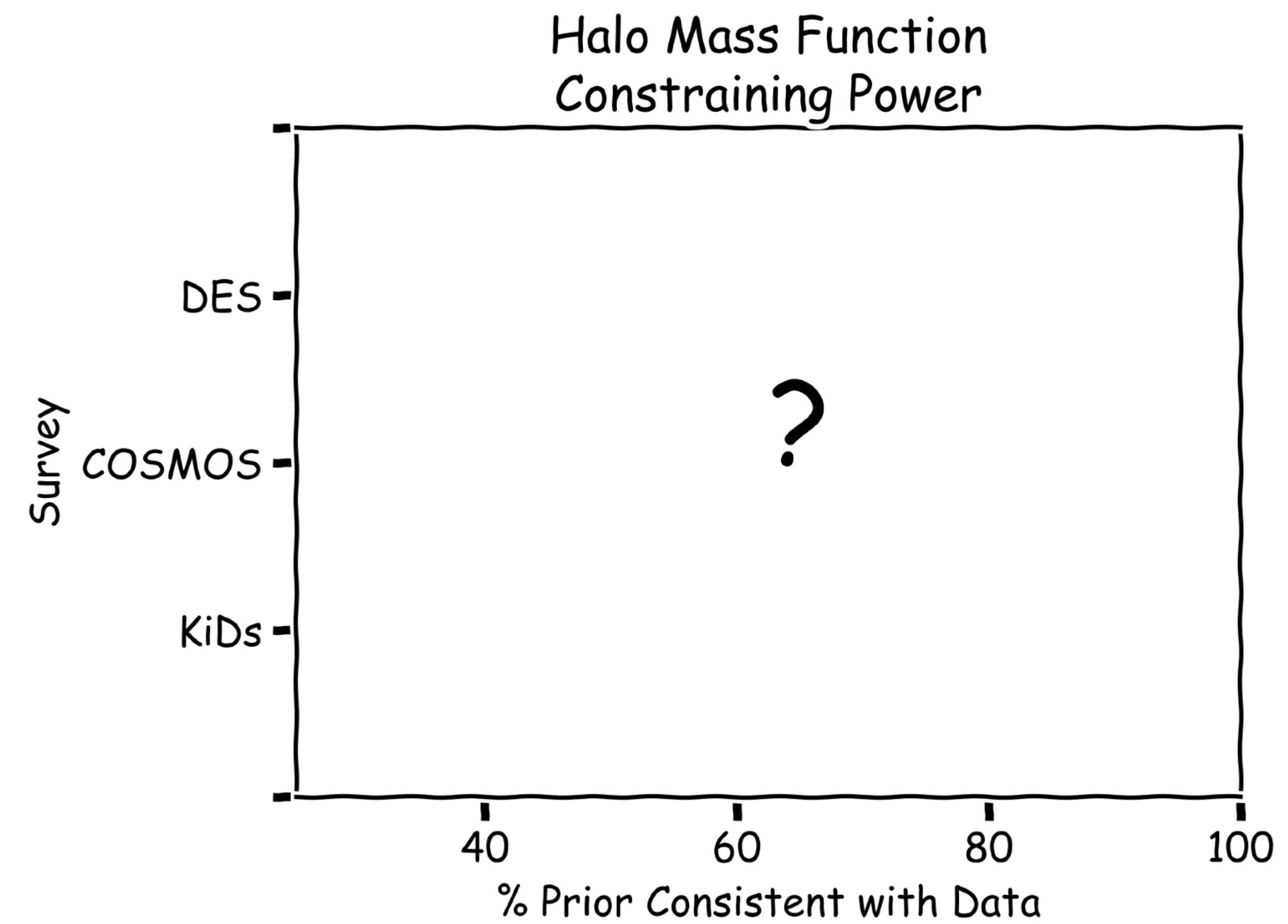
Applications to galaxies?

- None of this has been tested!
- Discussed three tools
 - Marginal statistics
 - Population level analysis and the constraining power of different surveys
- Any prior you like
 - AGN, SFG priors?
- Marginal Likelihood Functions
 - Population level analysis



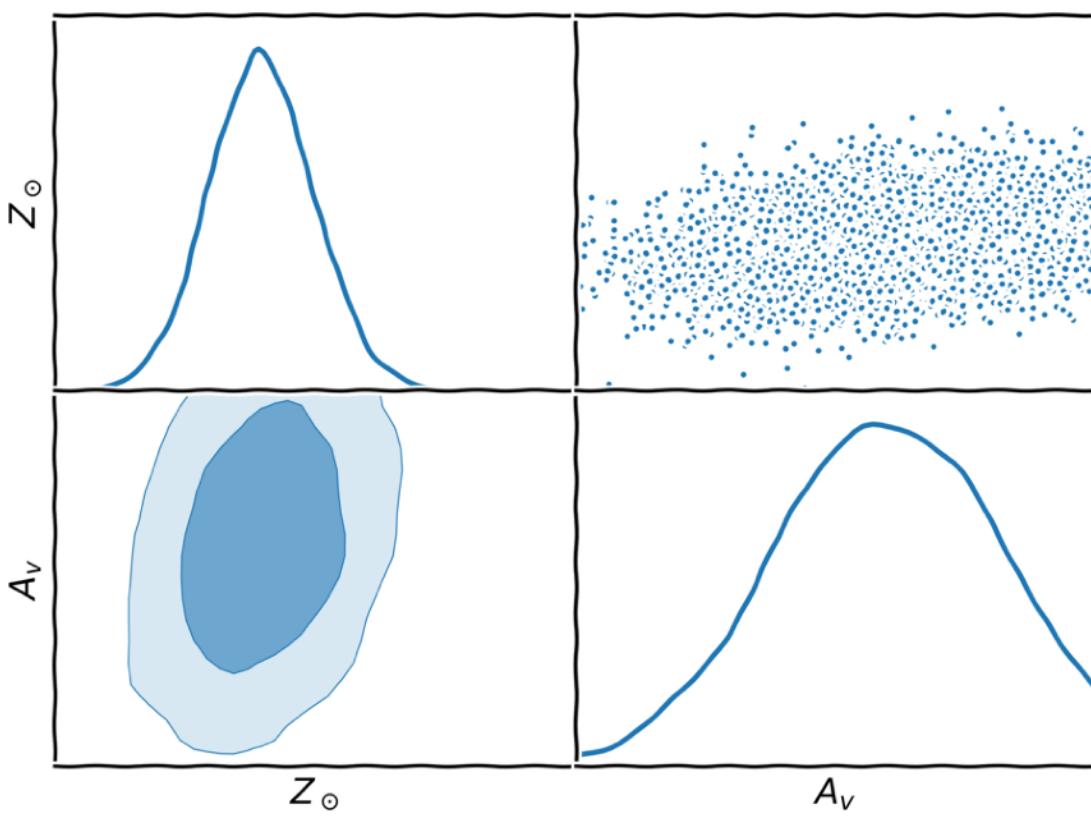
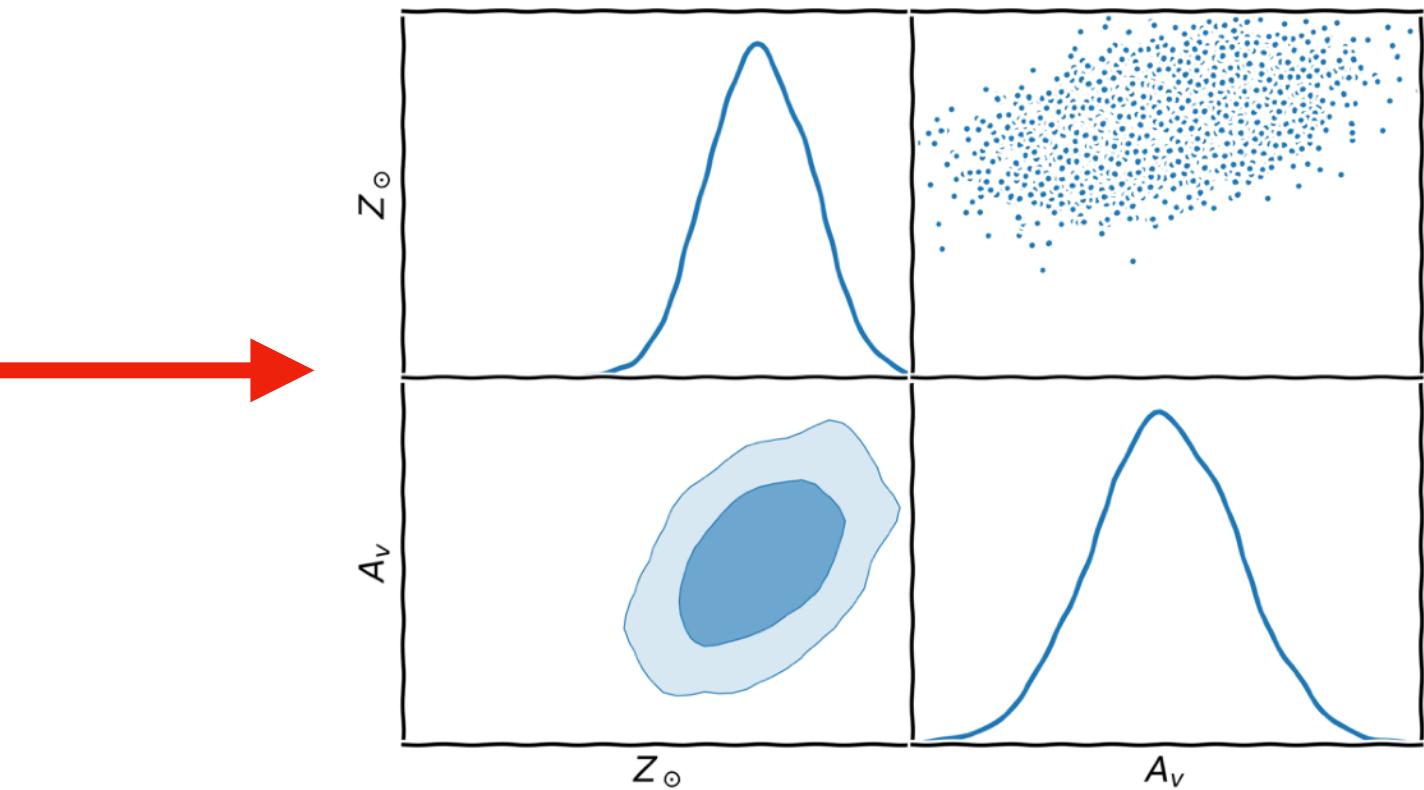
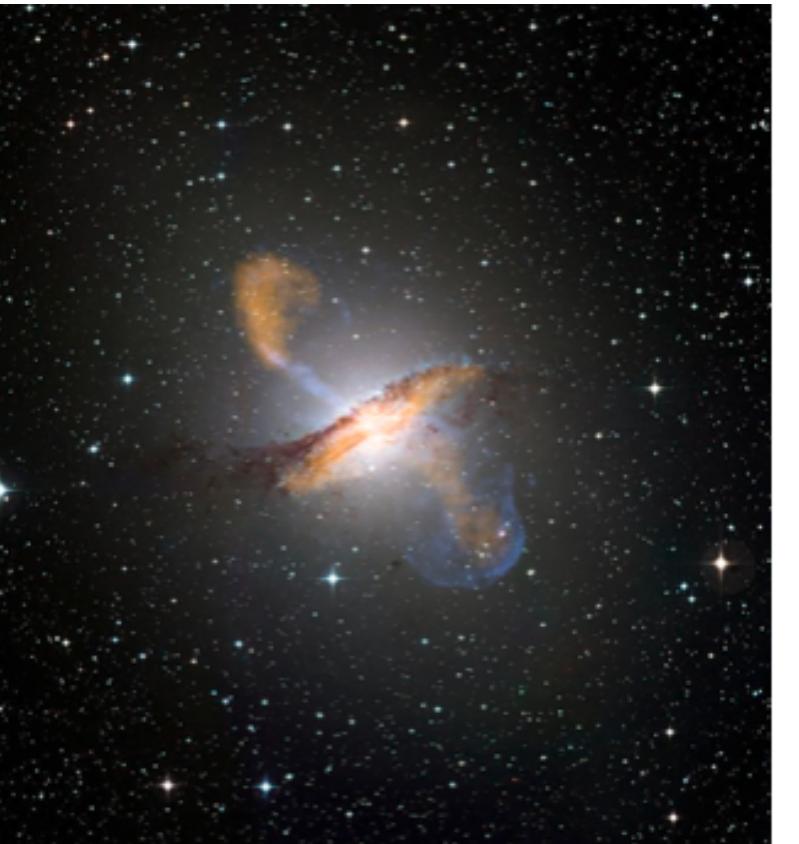
Marginal Statistics

- When we observe a set of galaxies we can think about constraining things like star formation rate density and halo mass functions
- We parameterise these functions with physically motivated models
- And then fit these models to galaxy surveys like SDSS, KiDs, COSMOS, DES, JADES etc
- Use marginal KL divergence to determine which surveys are the most informative



Any prior you like?

- We spend a lot of time classifying galaxies e.g. AGN, SFG etc
- Classification based on morphology or spectral properties e.g. BPT diagrams
- When fitting SEDs we might therefore expect particular classes of galaxies to have similar posterior distributions
- Can use archetypal galaxies and normalising flows to define an AGN or SFG prior at fixed redshifts



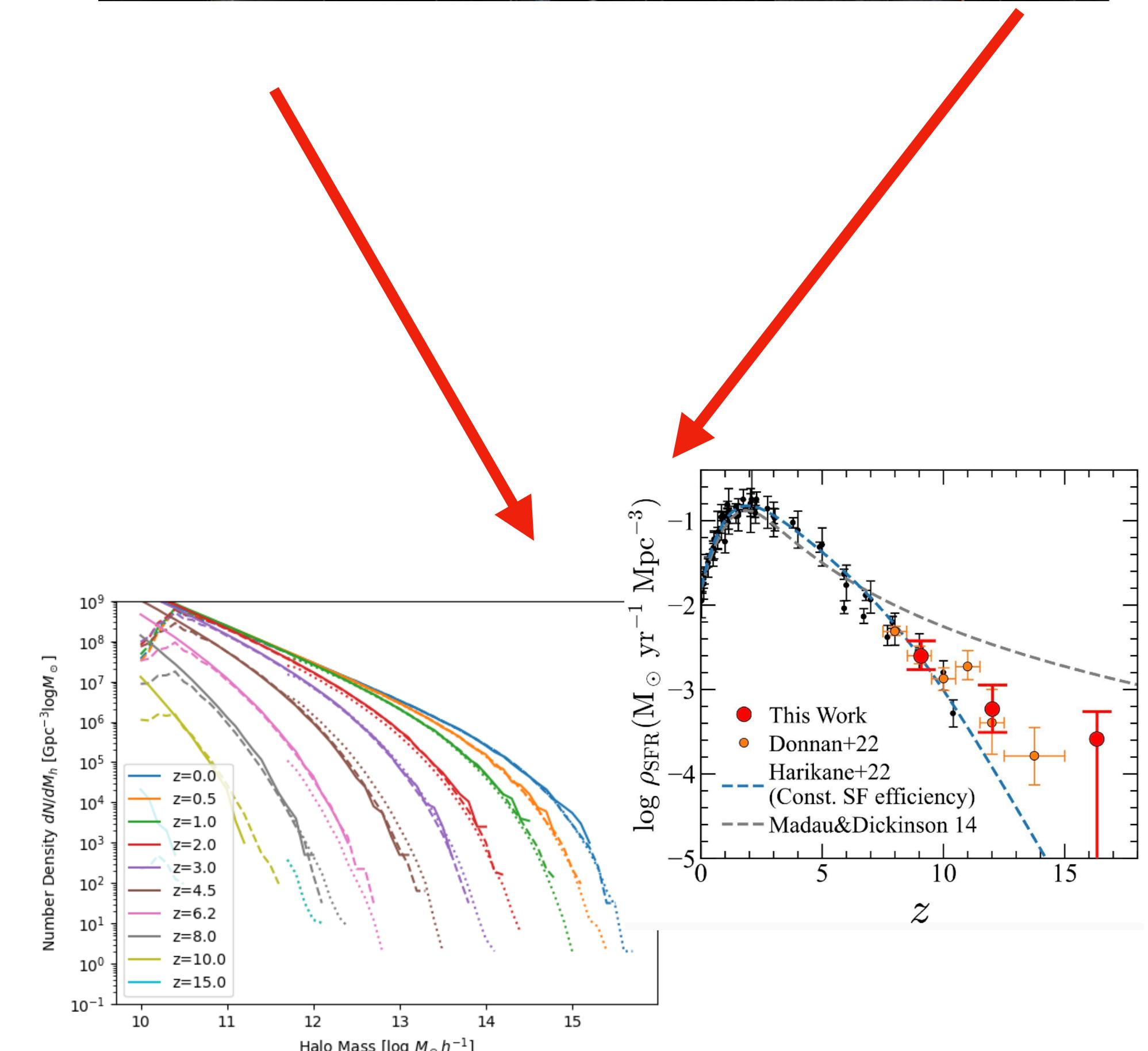
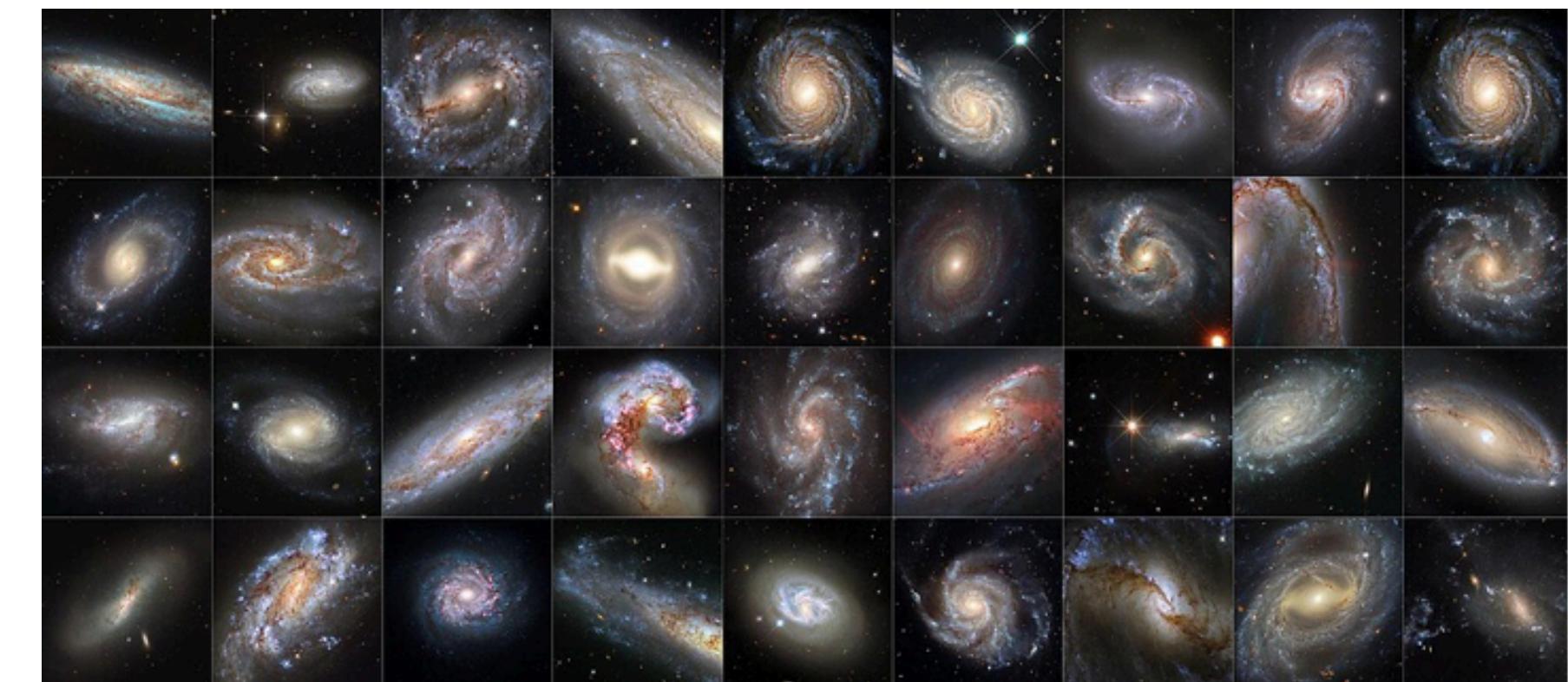
Marginal Likelihood Functions

- When studying galaxy populations one typically needs to perform a joint fit of many galaxy SEDs with likelihoods

$$P(D_{\text{survey}} | \theta_{\text{pop}}, \{\alpha\}^{N_{\text{gal}}}) = \prod_i^{N_{\text{gal}}} P(D_{\text{gal},i} | \theta_{\text{pop}}, \alpha_i)$$

- If all we are interested in is θ_{pop} then we can fit each galaxy individually, use NFs to emulate $P(D_{\text{gal},i} | \theta_{\text{pop}})$ and jointly sample

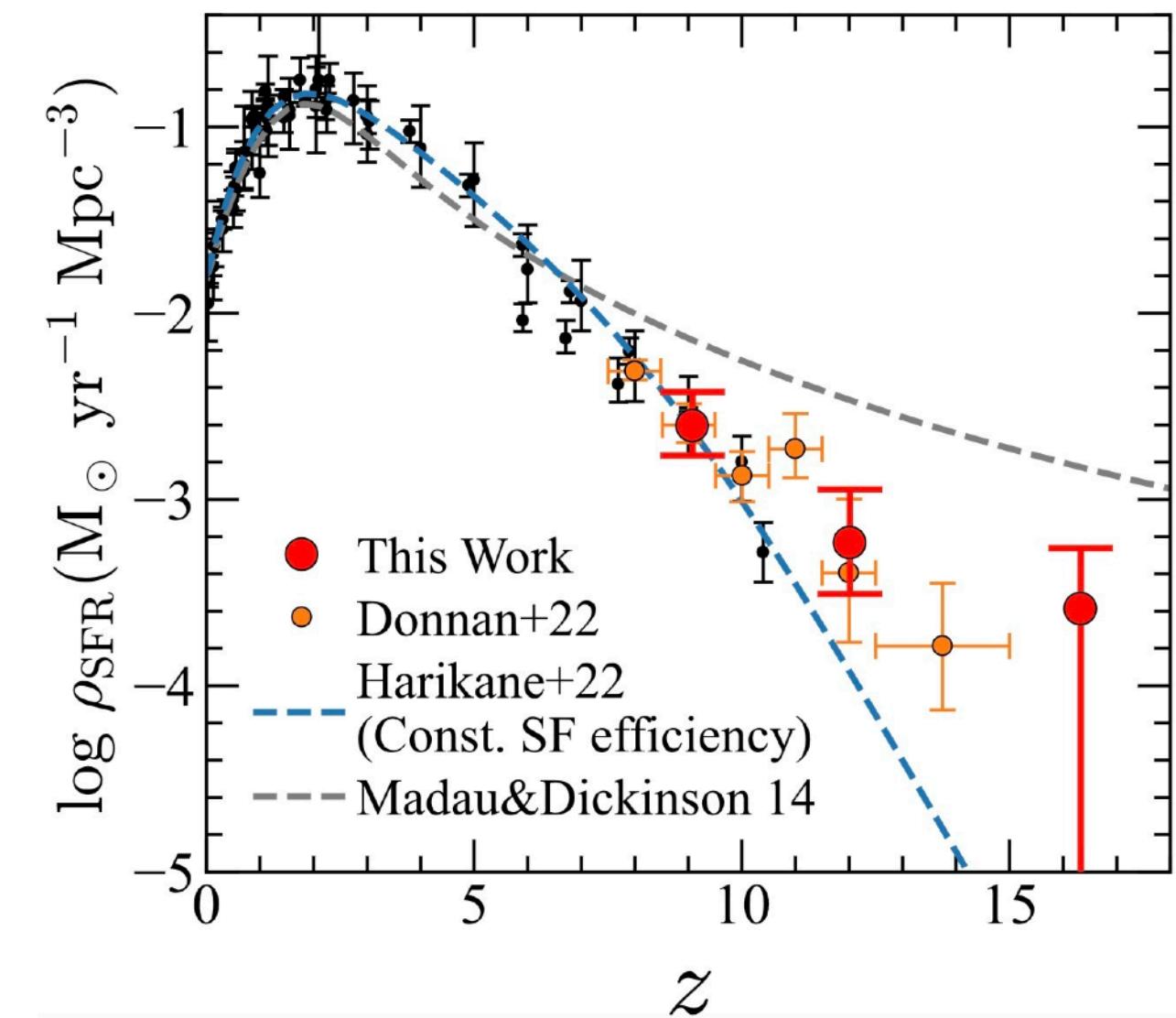
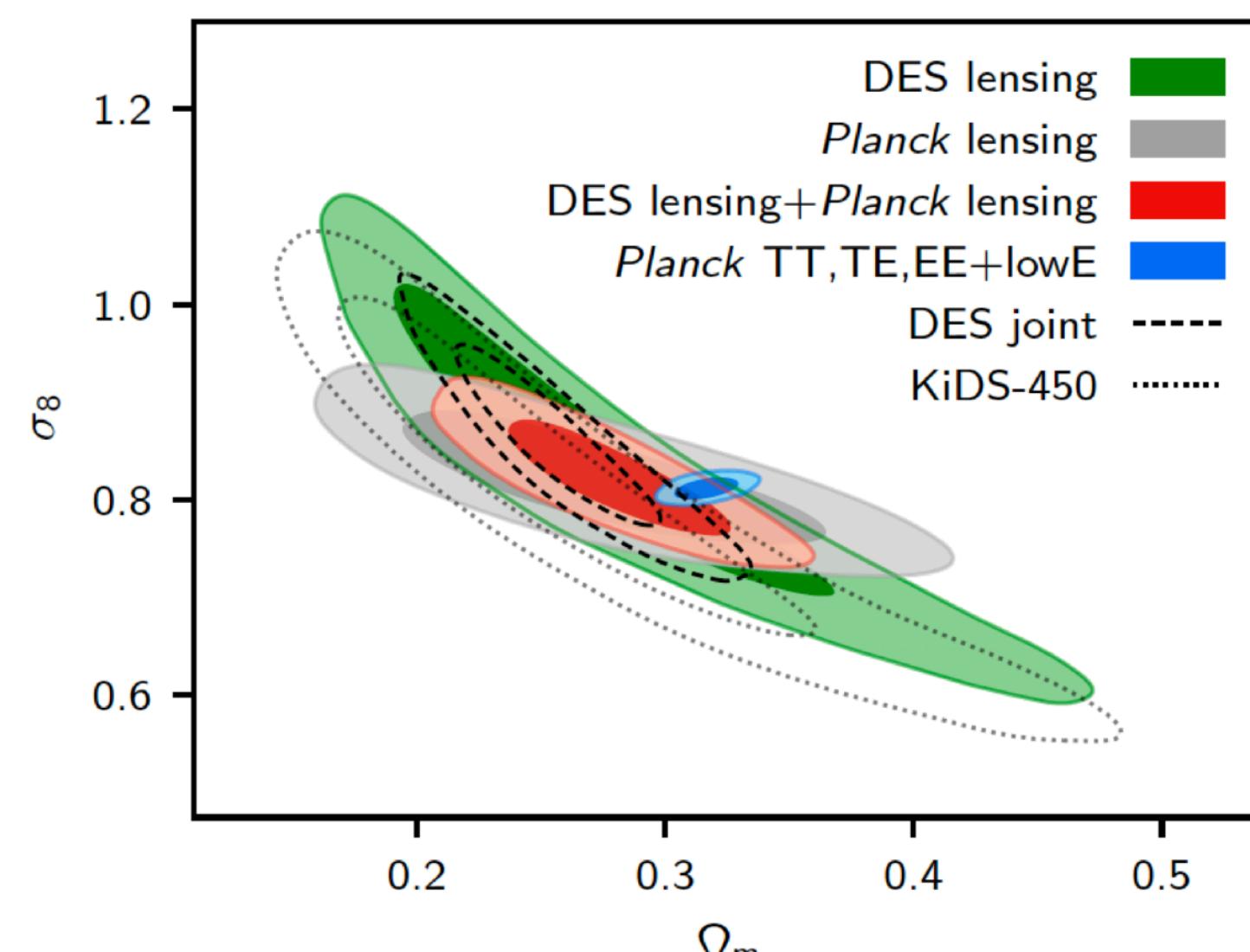
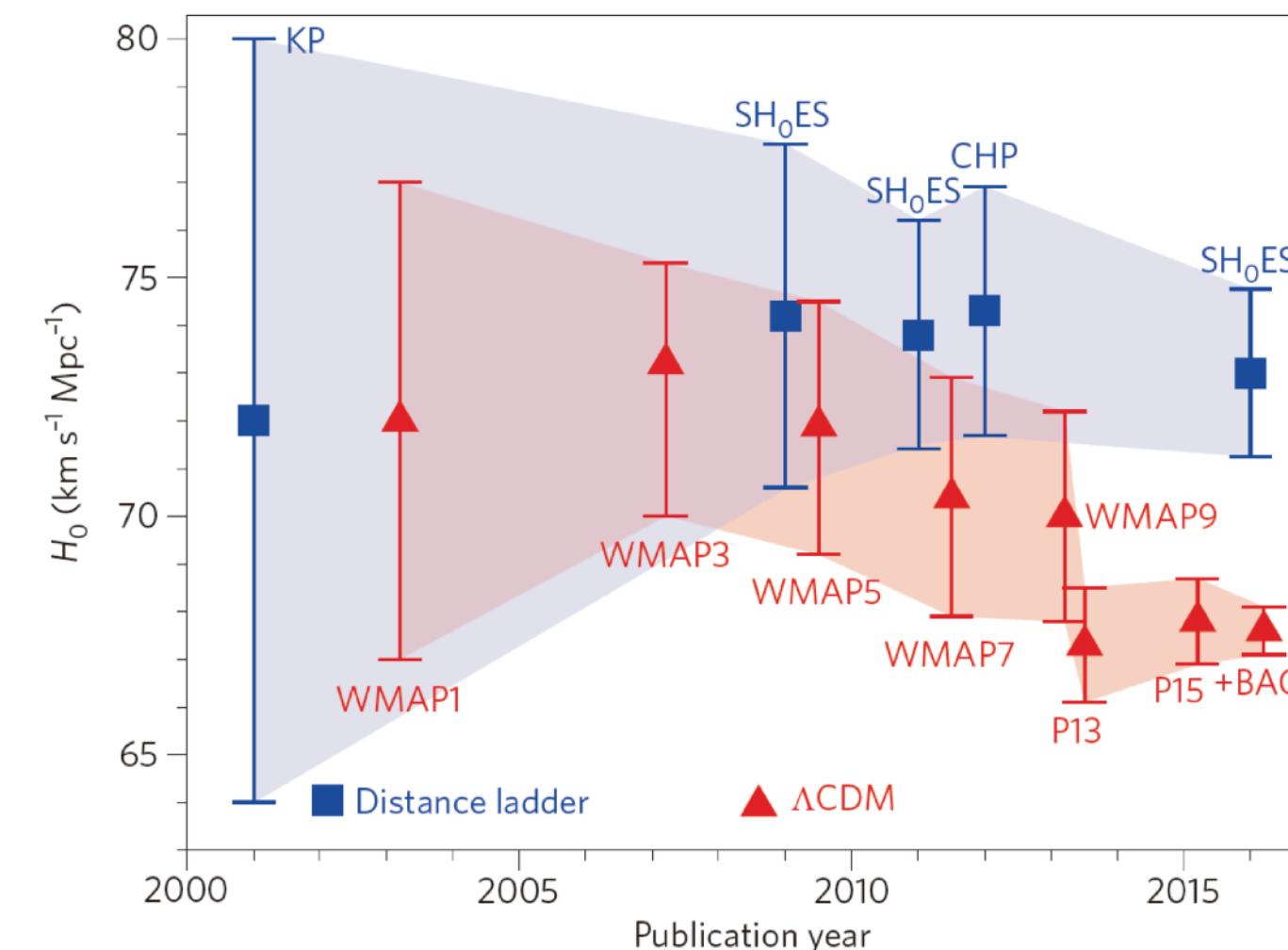
$$P(D_{\text{survey}} | \theta_{\text{pop}}) = \prod_i^{N_{\text{gal}}} P(D_{\text{gal},i} | \theta_{\text{pop}})$$



Calibrating Tension Statistics

Tension in Cosmology and Astrophysics

- We are all familiar with the idea of tensions e.g. H_0 and σ_8
- Understanding where tension comes from is important
- It can lead us to new physics and a better understanding of our instruments/systematics
- Quantifying tension properly is therefore very important

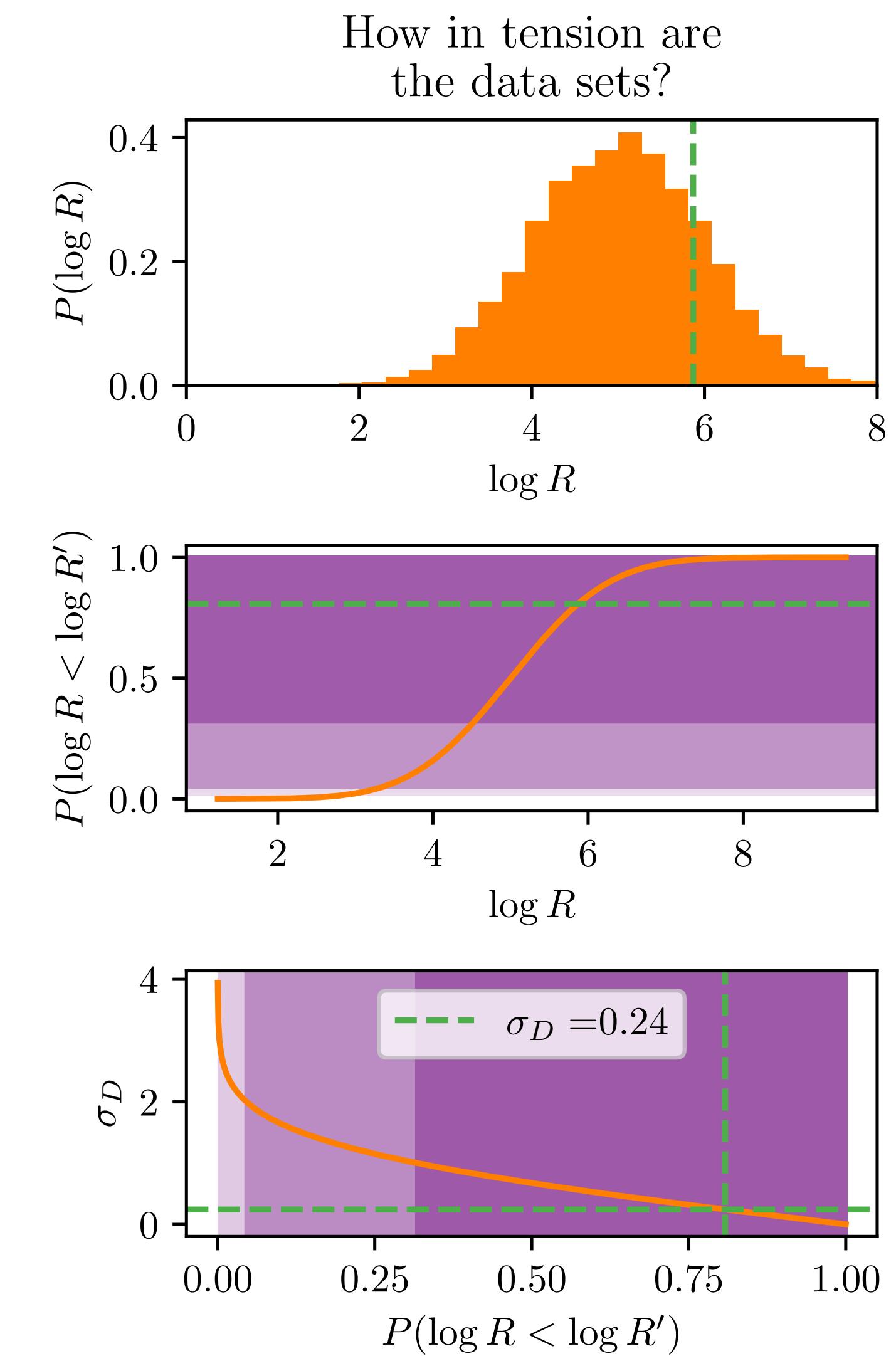


The R statistic

- An appropriately Bayesian way to do this is the R statistic

$$R = \frac{P(D_A, D_B)}{P(D_A)P(D_B)} = \frac{Z_{AB}}{Z_A Z_B}$$

- R is prior dependent and hard to interpret
- For every pair of experiments, prior and model there is a distribution of possible in concordance R statistic
- Having access to this distribution allows us to calibrate out the prior dependence of the true R for the real data



Approximating the distribution of R

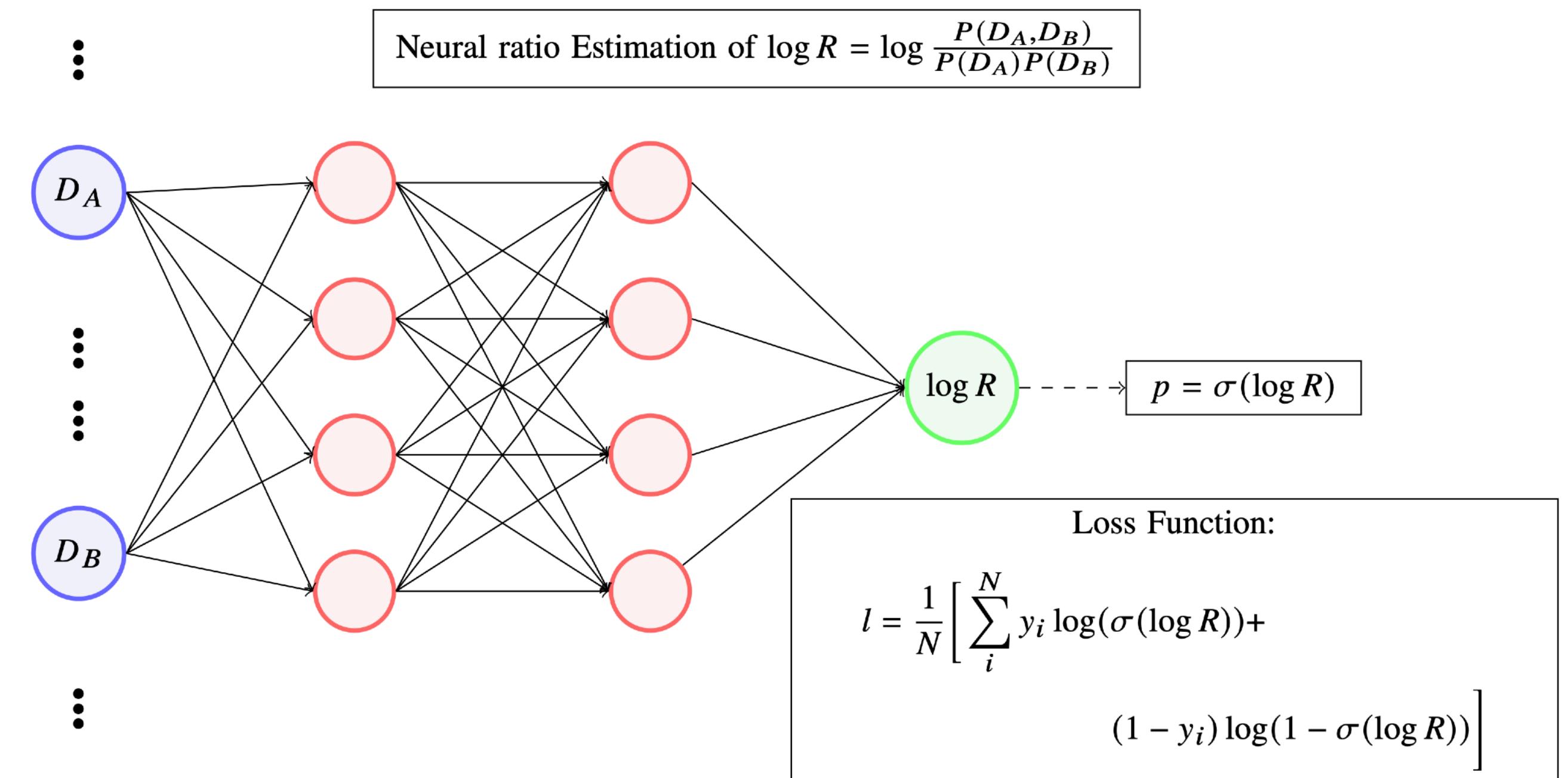
- We can approximate this distribution with neural ratio estimation

- Technique from simulation based inference

- Classifier that learns

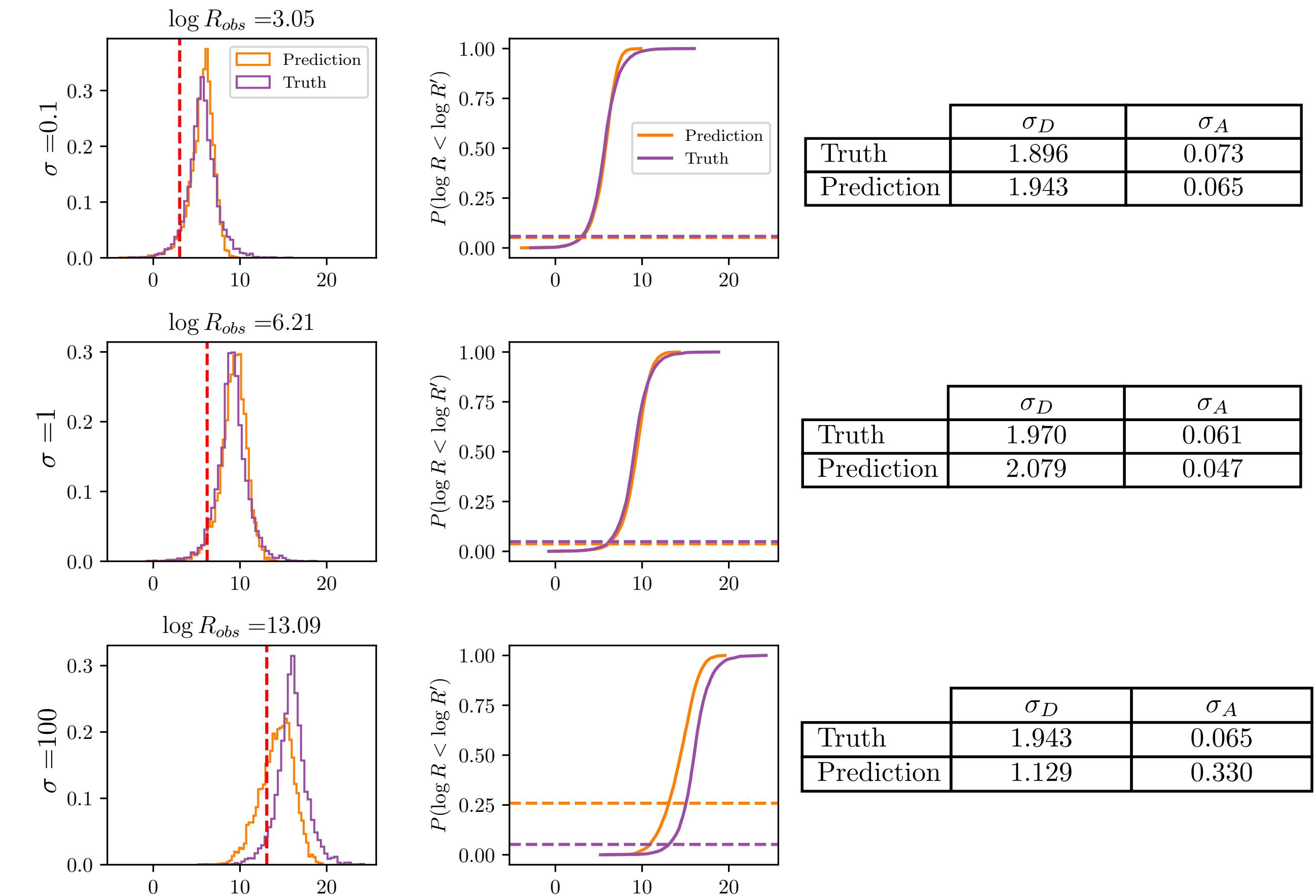
$$r = \frac{P(A, B)}{P(A)P(B)}$$

- So we can train the network to learn R with simulations of our observables drawn from the prior parameter space



Demonstration

- Simple linear model of two observables
- Gaussian prior and likelihood which means we can analytically calculate R to compare with predictions from our network
- We can see how R changes with the prior but σ_D and σ_A remain constant
- Can be used to assess tension between different galaxy surveys



Conclusions

- We can utilise machine learning tools to better interrogate the results of our Bayesian analysis
- Normalising flows can help us
 - better understand the constraining power of our data
 - Perform more efficient joint analysis of surveys
 - Perform more efficient SED fitting
- Neural Ratio Estimation can help us better understand tensions in our data
- Happy to discuss any of the ideas mentioned! Thanks for listening!

Papers: 2205.12841 and 2207.11457
Code: <https://github.com/htjb/margarine>

