Utilising Normalizing Flows to enhance our Bayesian workflows

Harry Bevins

On going work

UNIVERSITY OF CAMBRIDGE

CAVENDISH ASTROPHYSICS
RADIO COSMOLOGY
Science Technology Engineering Mathematics

Long list of people who have contributed to this work:

- Will Handley
- Justin Alsing
- Pablo Lemos
- Peter Sims
- Eloy de Lera Acedo
- Anastasia Fialkov

Papers:

- arXiv:2205.12841
- arXiv:2207.11457
- arXiv:2305.02930

Code:

- https://github.com/htjb/margarine
- https://github.com/htjb/piecewise_normalizing_flows

Removing the fat from your posterior samples with margarine

Harry T. J. Bevins, William J. Handley, Pablo Lemos, Peter H. Sims, Eloy de Lera Acedo, Anastasia Fialkov, Justin Alsing

Bayesian workflows often require the introduction of nuisance parameters, yet for core science modelling one needs access to a marginal posterior densit
we use masked autoregressive flows and kernel density estimators to encapsulate the marginal posterior, allowing us to compute marginal Kullback-Leib

Marginal Bayesian Statistics Using Masked Autoregressive Flows and Kernel Density Estimators with Examples in Cosmology

Harry Bevins, Will Handley, Pablo Lemos, Peter Sims, Eloy de Lera Acedo, Anastasia Fialkov

Cosmological experiments often employ Bayesian workflows to derive constraints on cosmological and astrophysical parameters from their data. It has been shown that these constraints can be combined across different probes such as Planck and the Dark Energy Survey and that this can be a valuable exercise to improve our

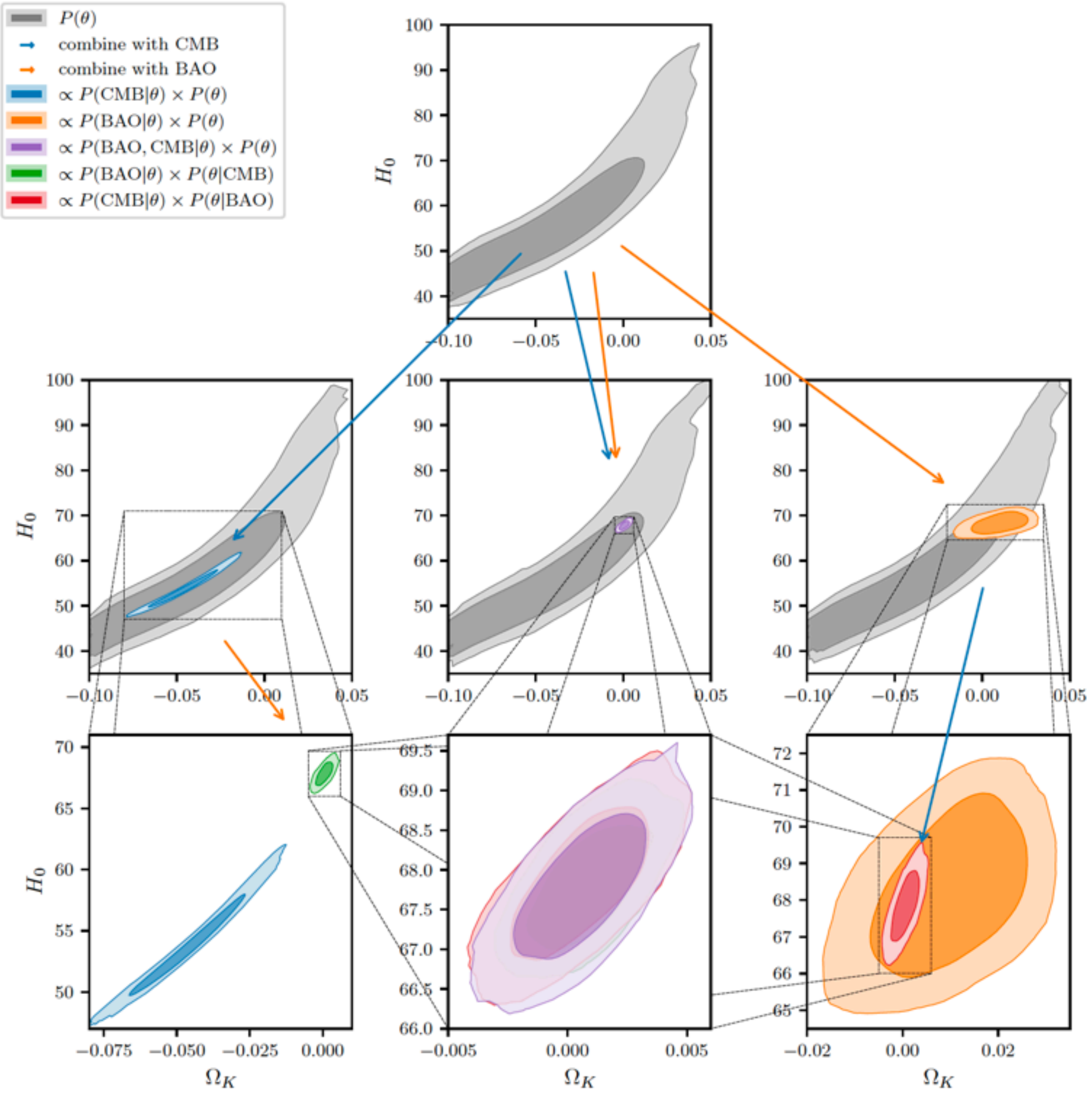
Piecewise Normalizing Flows

Harry Bevins, Will Handley

Normalizing flows are an established approach for modelling complex probability densities through
accuracy with which the target distribution can be captured by the normalizing flow is strongly influ

Any prior you like?

- Building on work done in Alsing and Handley 2021
- It was shown that we could use trained Normalizing Flows as priors in our Bayesian analysis
- Possible because Normalizing Flows are bijective and give access to probabilities



What is a Normalizing Flow?

Fundamentals

- A series of bijective transformations from one probability distribution to another

- Base distribution is usually a standard normal

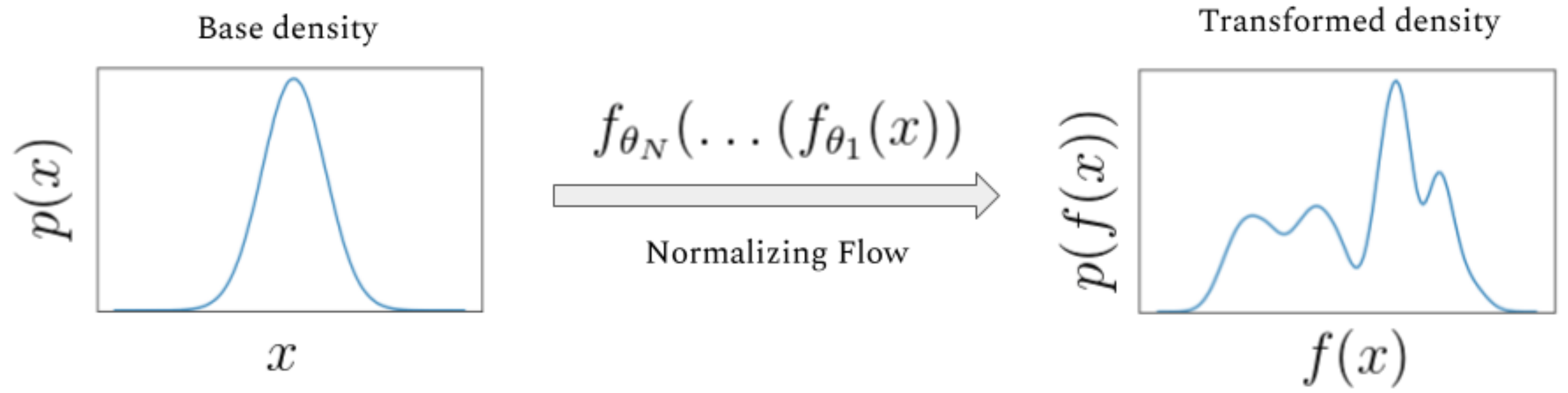
- Transformation is differentiable

- If we say $x' = f(x)$ then we can calculate

$$p(x') = p(f^{-1}(x')) \left| det\left(\frac{\delta f^{-1}(x')}{\delta x'}\right) \right|$$

Which is functionally equivalent to

$$p(x') = p(x) \left| det\left(\frac{\delta x}{\delta x'}\right) \right|$$



Masked Autoregressive Flows

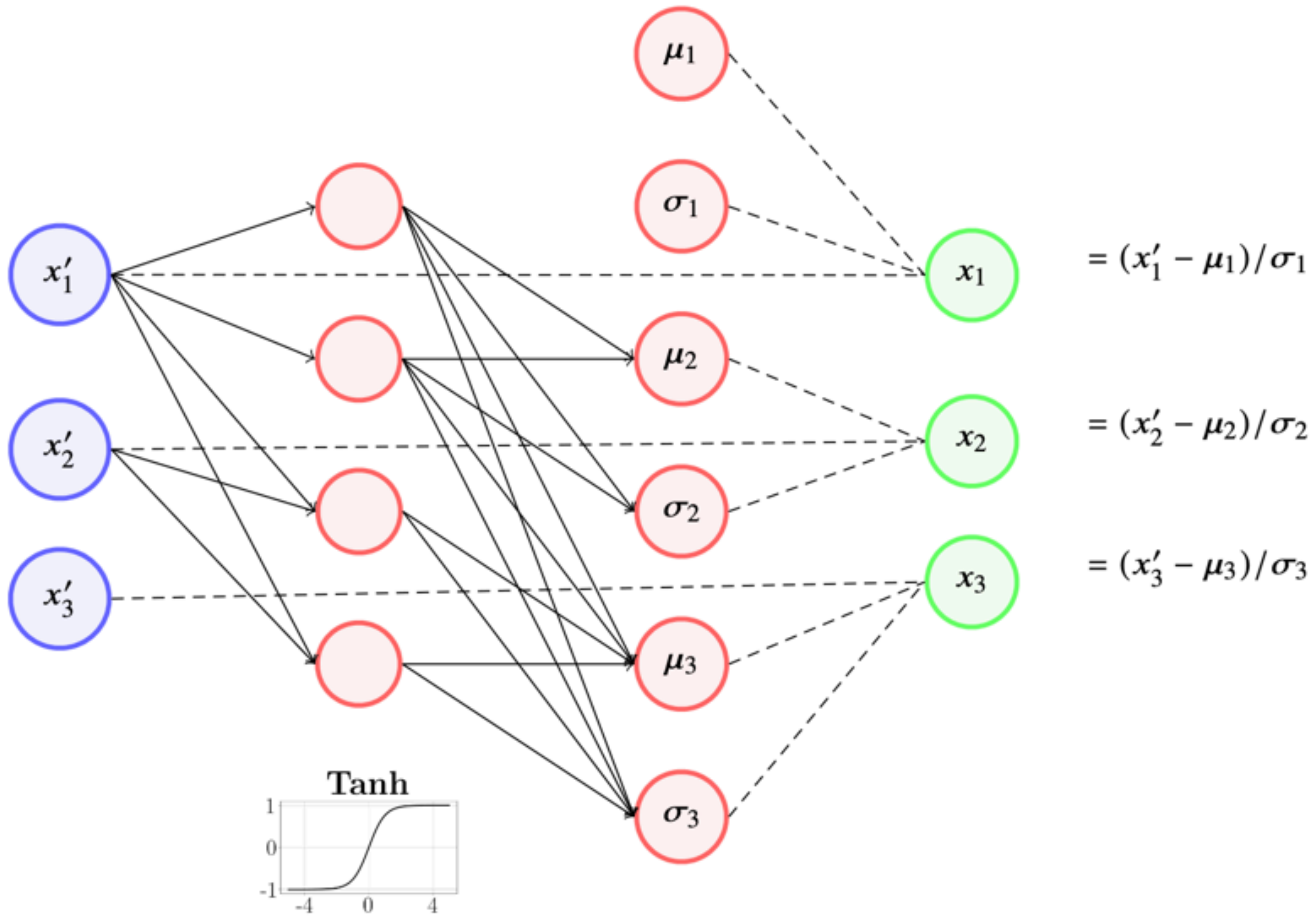
- Equate the function f to a trained neural network
- Use the Masked Autoencoder for Density Estimation [Germain et al. 2015 arXiv:1502.03509] architecture

$$p(x) = \prod_i p(x_i | x_1, x_2, \ldots, x_{i-1}) = \prod_i \mathcal{N}(\mu_i(x_1, x_2, \ldots, x_{i-1}), \sigma_i(x_1, x_2, \ldots, x_{i-1}))$$

- Learn μ and σ with the MADE network
- Train by minimising

$$\text{argmin}_\theta - \sum_{j=0}^{N} \log p_\theta(x_j)$$

- Increased expressibility when we use a series of networks chained together



margarine

UNIVERSITY OF CAMBRIDGE

CAVENDISH ASTROPHYSICS
RADIO COSMOLOGY
Science Technology Engineering Mathematics

- Python implementation with tensorflow, keras and scipy
- Density estimation through Normalizing Flows and KDEs
- Easy to use with tutorials and a customer help line (email me!)
- Continuously integrated tests
- pip installable
- https://github.com/htjb/margarine

README.rst

margarine: Posterior Sampling and Marginal Bayesian Statistics

Introduction

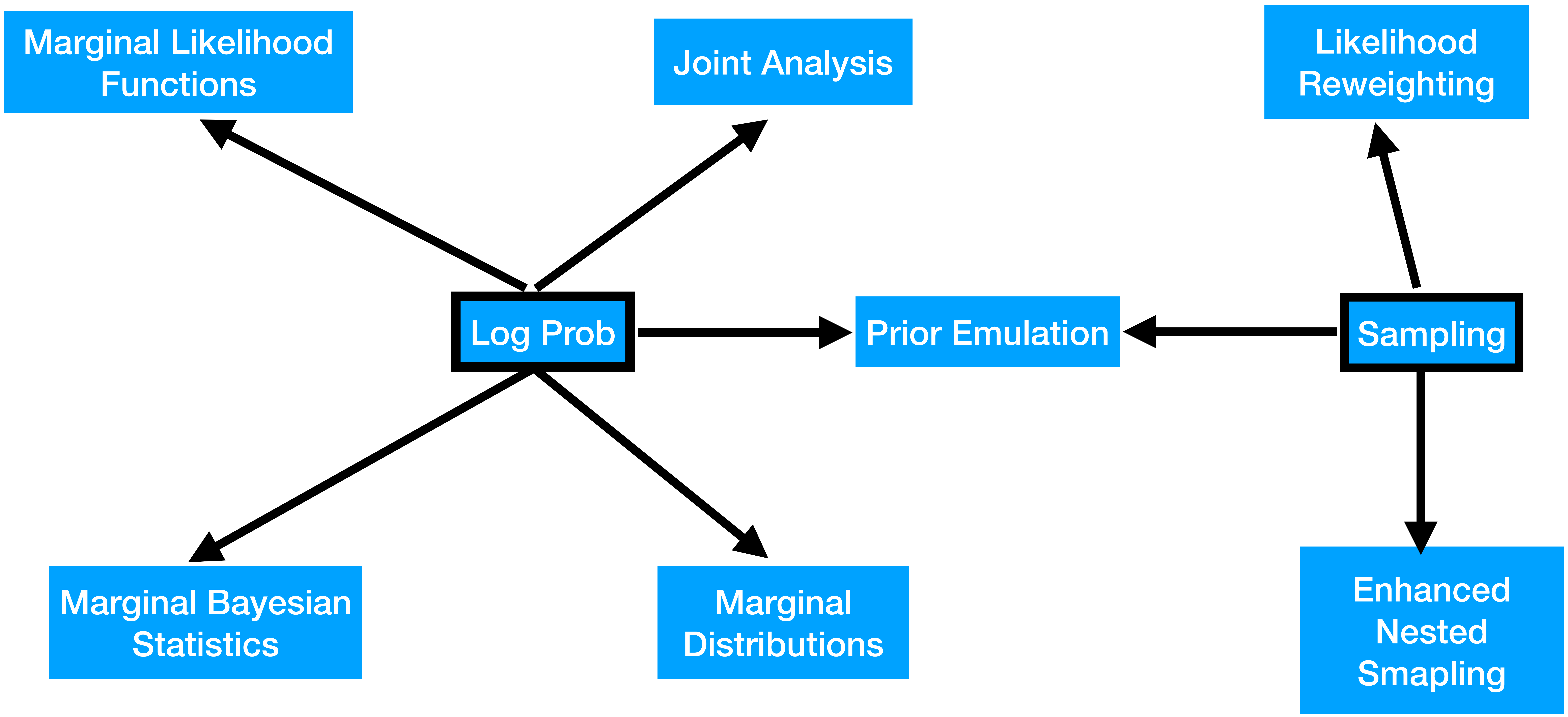
margarine:	Marginal Bayesian Statistics
Authors:	Harry T.J. Bevins
Version:	0.5.0
Homepage:	https://github.com/htjb/margarine
Documentation:	https://margarine.readthedocs.io/

docs passing | launch binder | astro.IM arXiv:2205.12841

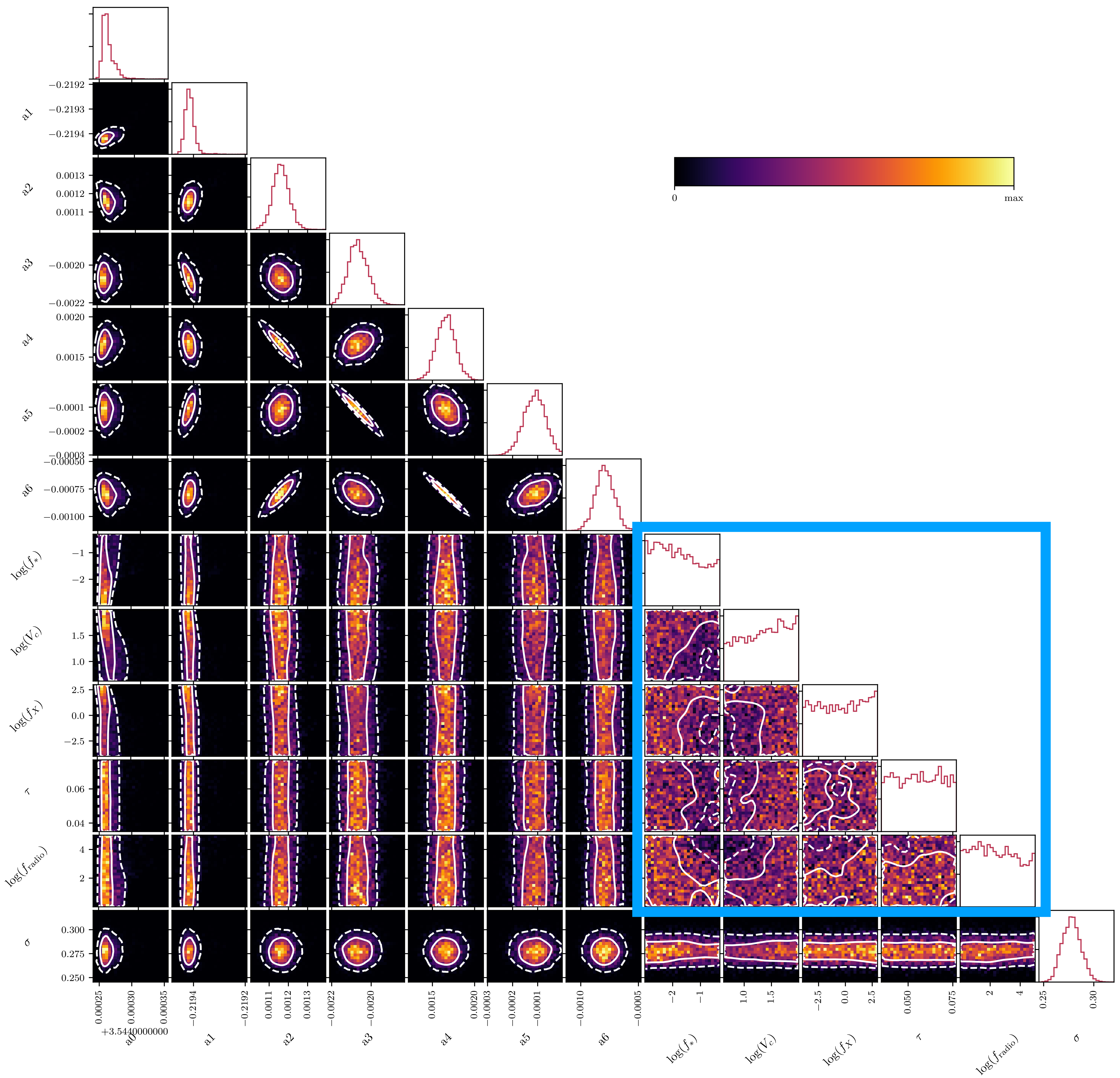
```
from margarine.maf import MAF
flow = MAF(data, weights)
flow.train(10000, early_stop=True)
samples = flow.sample(5000)
log_probs = flow.log_prob(samples)
```

Why are Normalizing Flows useful?

Samples and log probabilities



Marginal distributions



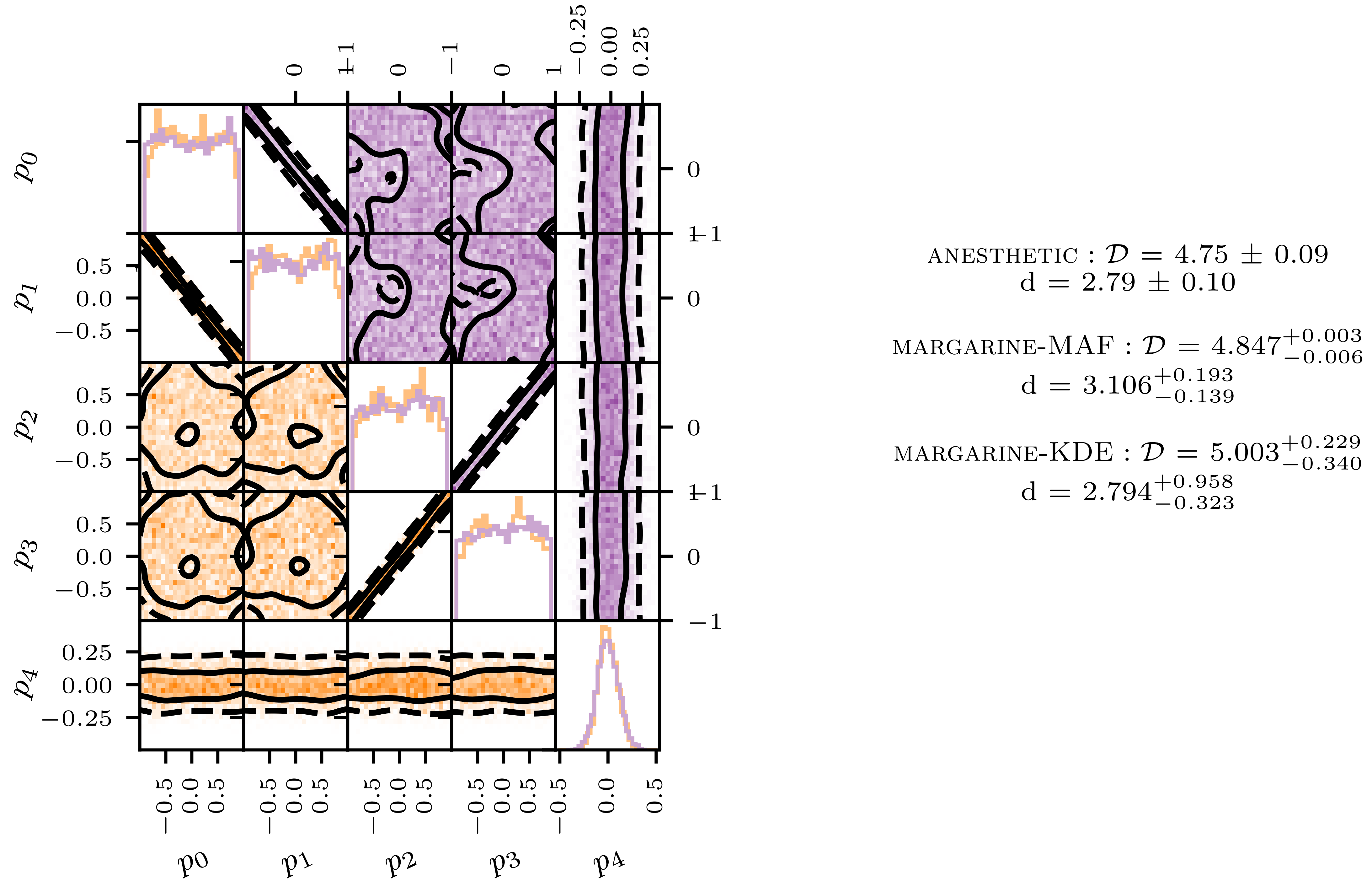
- Experimental data sets are described by nuisance parameters α and core science parameters θ

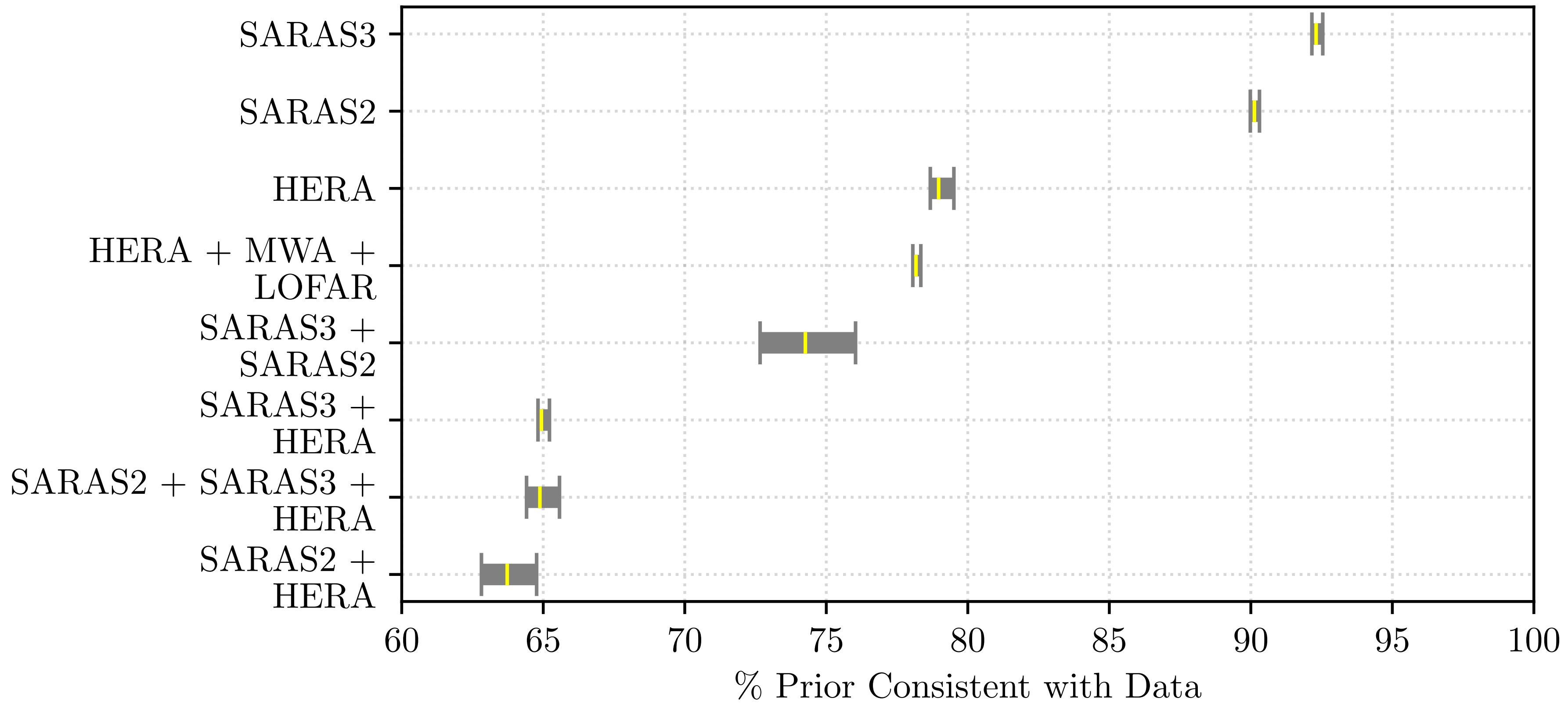
- Evaluating $P(\theta)$ is hard when we have samples on $P(\theta, \alpha)$

- Train density estimators on $\{\theta\}$ to get $P(\theta)$ marginalising over α

Marginal Bayesian Statistics

- Through access to $P(\theta)$ with our Normalizing Flows we can evaluate $\log P(\theta)$
- If we can model the posterior, $P(\theta)$, and prior, $\pi(\theta)$ with Normalizing Flows (or analytically in the case of the prior) we can evaluate marginal Bayesian statistics
- Marginal Kullback-Lieber Divergence and Marginal Bayesian Model Dimensionality
- Independent of the nuisance parameters
- Allows for comparisons across different experiments probing the same core science



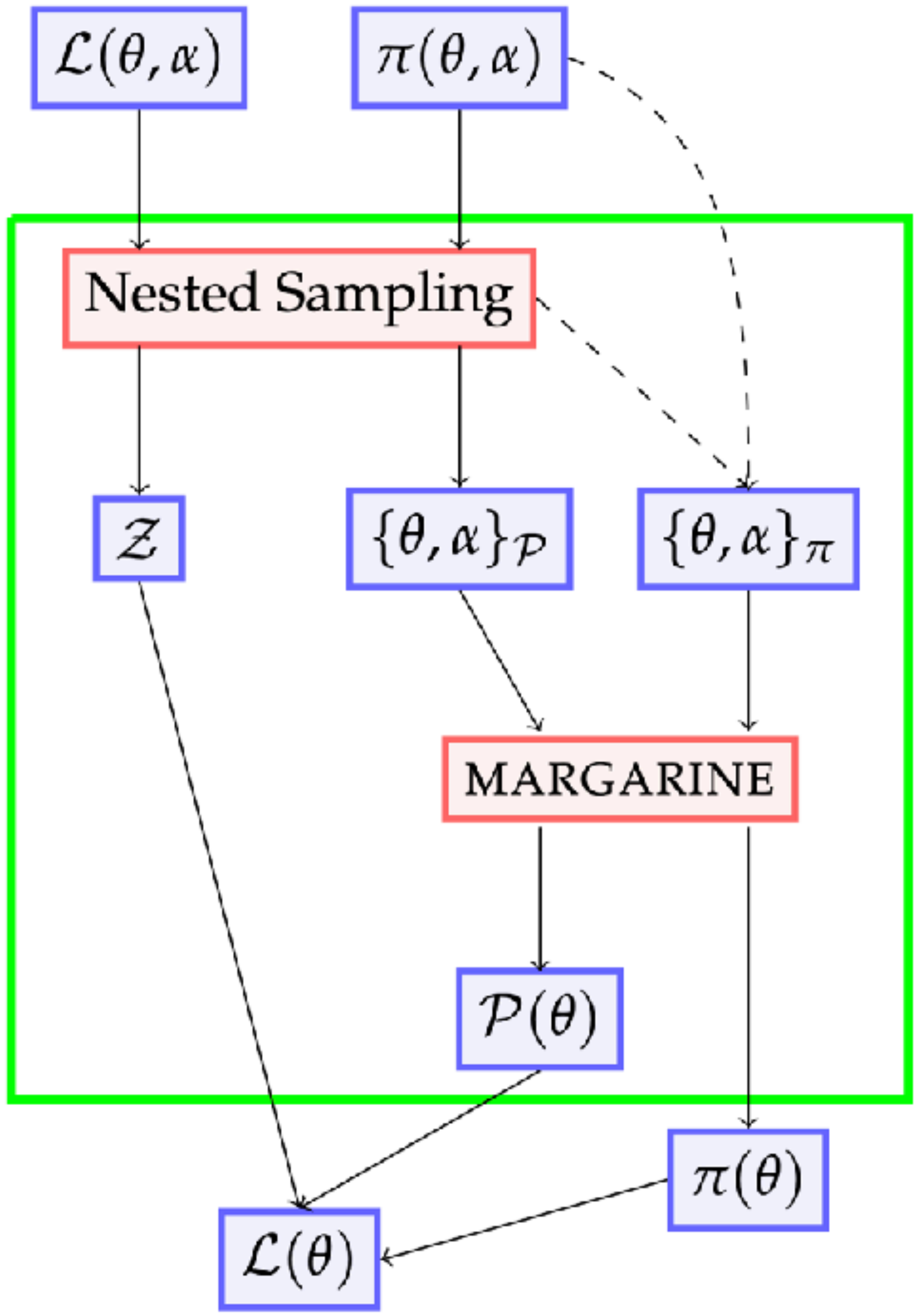


Marginal likelihood functions

- With samples $\{\theta, \alpha\}$ and a corresponding evidence Z we can define the marginal or nuisance-free likelihood as

$$L(\theta) = \frac{\int L(\theta, \alpha)\pi(\theta, \alpha) d\alpha}{\int \pi(\theta, \alpha) d\alpha} = \frac{P(\theta) Z}{\pi(\theta)}$$

- Use $\mathrm{margarine}$ to access $P(\theta)$ and $\pi(\theta)$



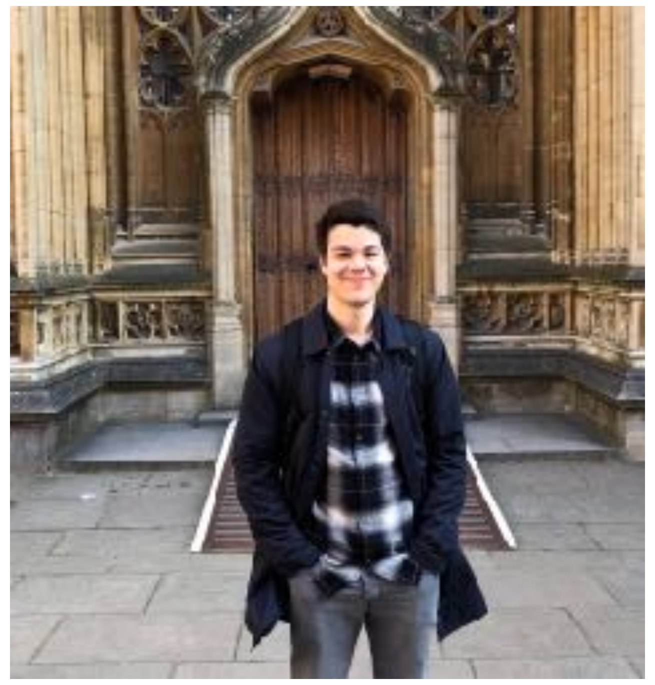
Joint analysis

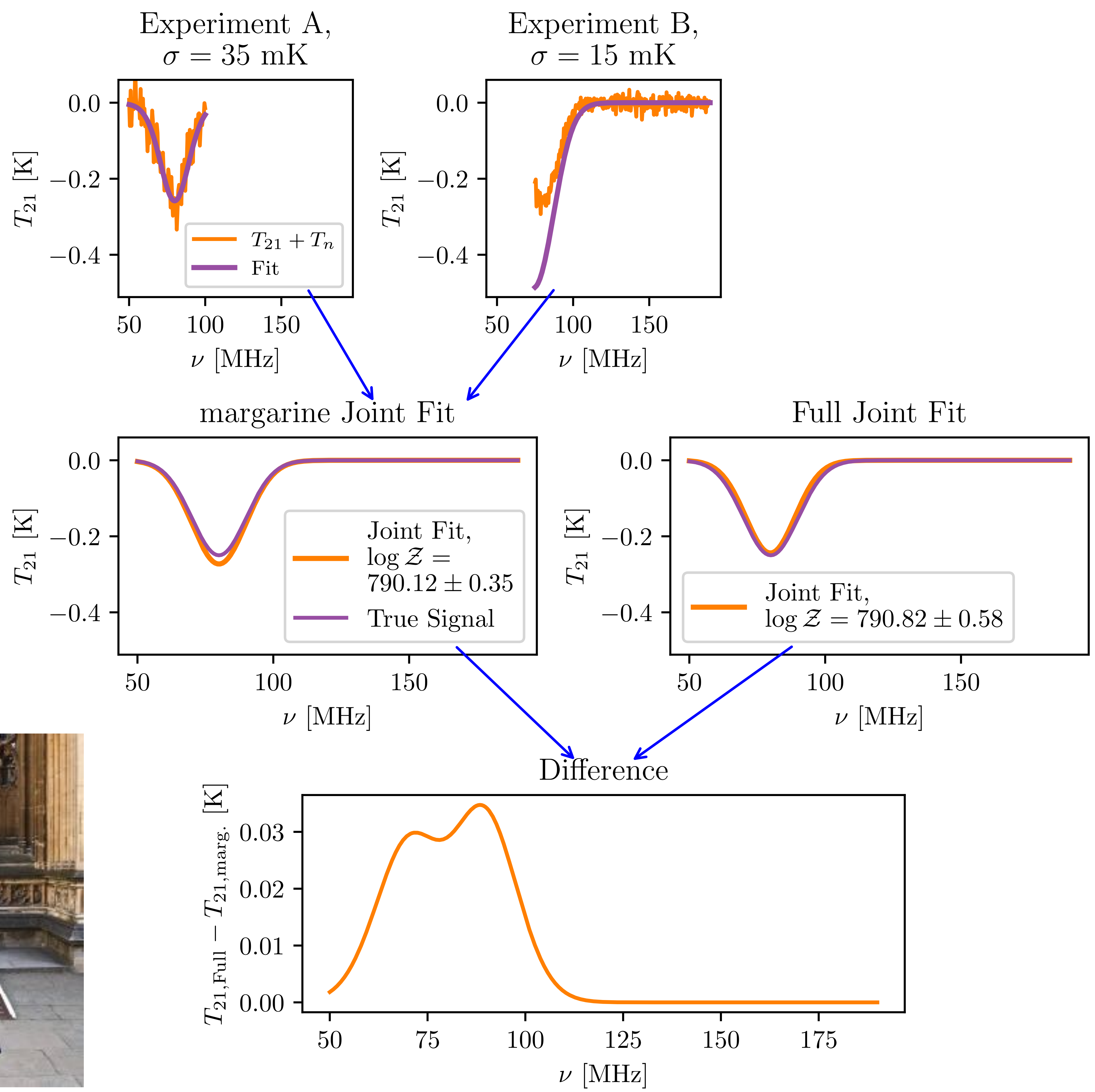
- If we have $L_A(\theta, \alpha_A)$ and $L_B(\theta, \alpha_B)$ and we want to perform joint analysis we can access $L_A(\theta)$ and $L_B(\theta)$ and sample

$$\log L_{AB}(\theta) = \log L_A(\theta) + \log L_B(\theta)$$

- Perform joint analysis without sampling nuisance parameters

- See Irene Abril-Cabezas' and Simon Pochinda's talks later today.





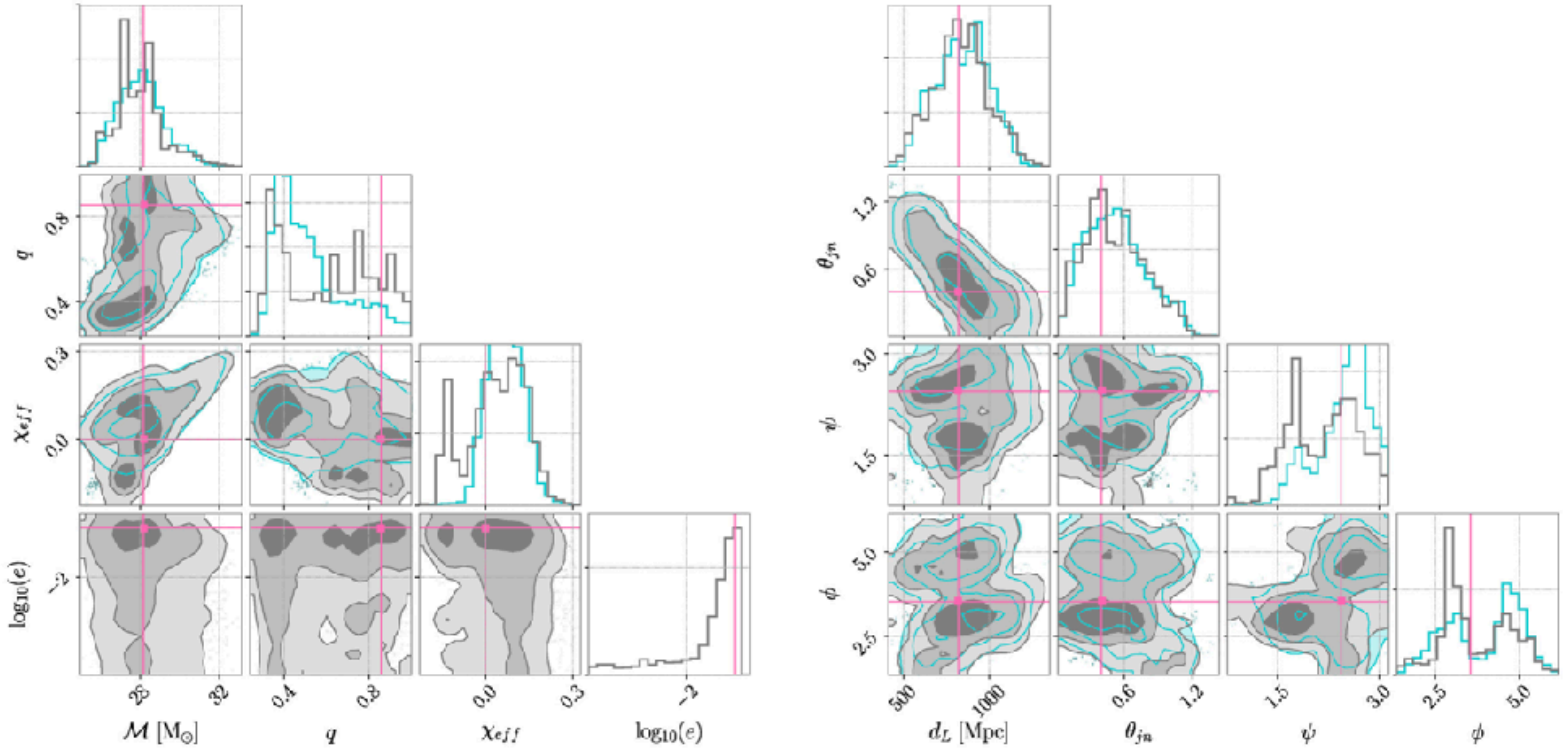
Future work?

Enhanced Likelihood Reweighting

- Slow likelihood B and a related fast likelihood A
- Sample fast likelihood and reweight samples onto slow likelihood

$$P_B(\theta) = P_A(\theta)\frac{L_B(\theta)}{L_A(\theta)}$$

- Only have to evaluate the slow likelihood a few thousand times rather than millions
- Pioneered for gravitational wave studies
- Can have too few samples in $P_A(\theta)$ to properly describe $P_B(\theta)$
- Emulate $P_A(\theta)$ and $L_A(\theta)$ with margarine and upsample until we have an appropriate n_{eff}



Romero-Shaw et al. 2019 arXiv:2108.01284

Enhanced Nested Sampling

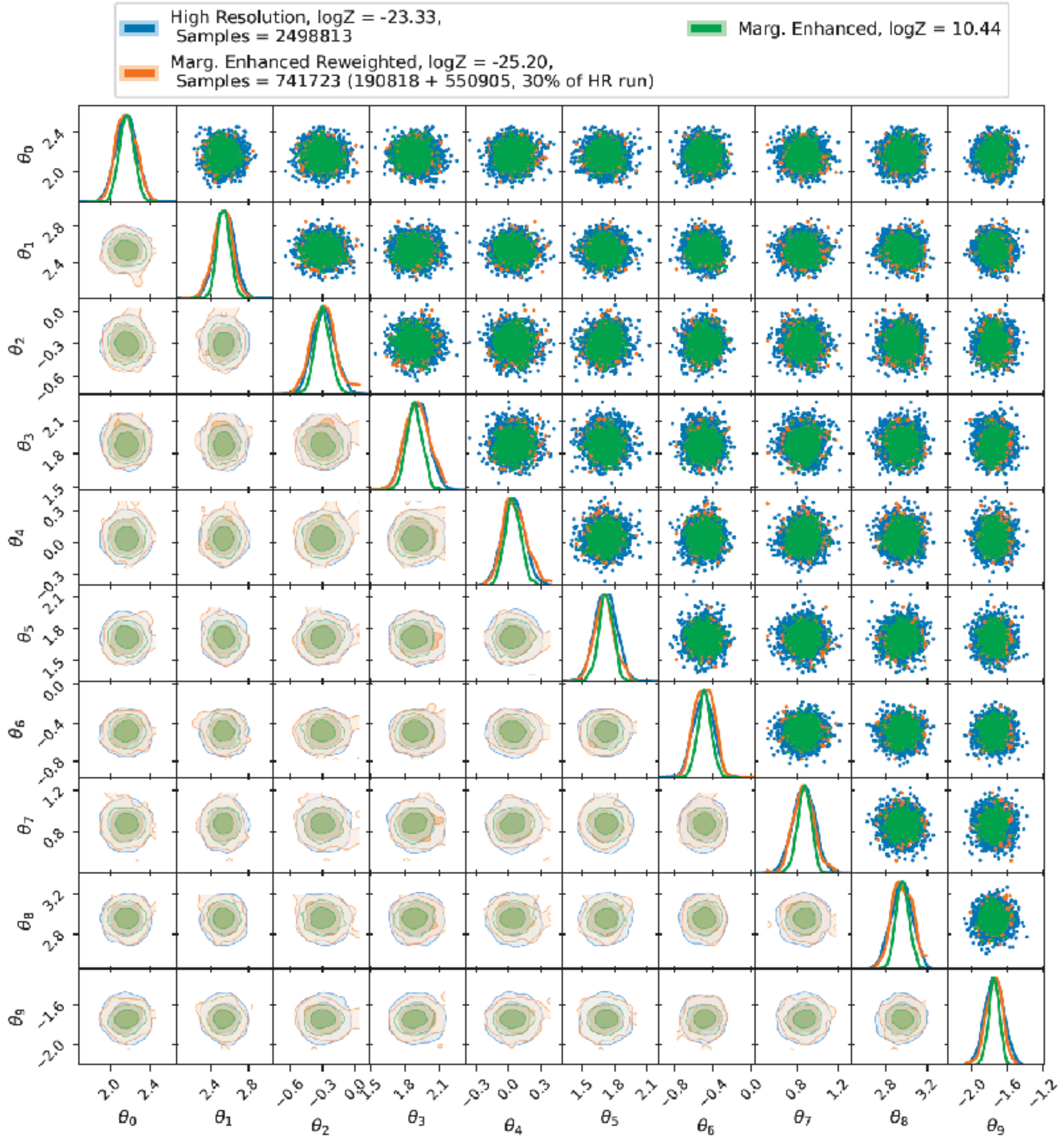
- We can speed up run time using better proposal distributions for the prior (reducing KL divergence)

$$t \propto \mathscr{D}_{\mathrm{KL}}$$

- Previously explored with `supernest` (Petrosyan and Handley 2022 arXiv:2212.01760)

- Low resolution (low n_{live}) sampling of the likelihood, train `margarine` on the resultant posterior and use that as the prior on a high resolution (high n_{live}) run

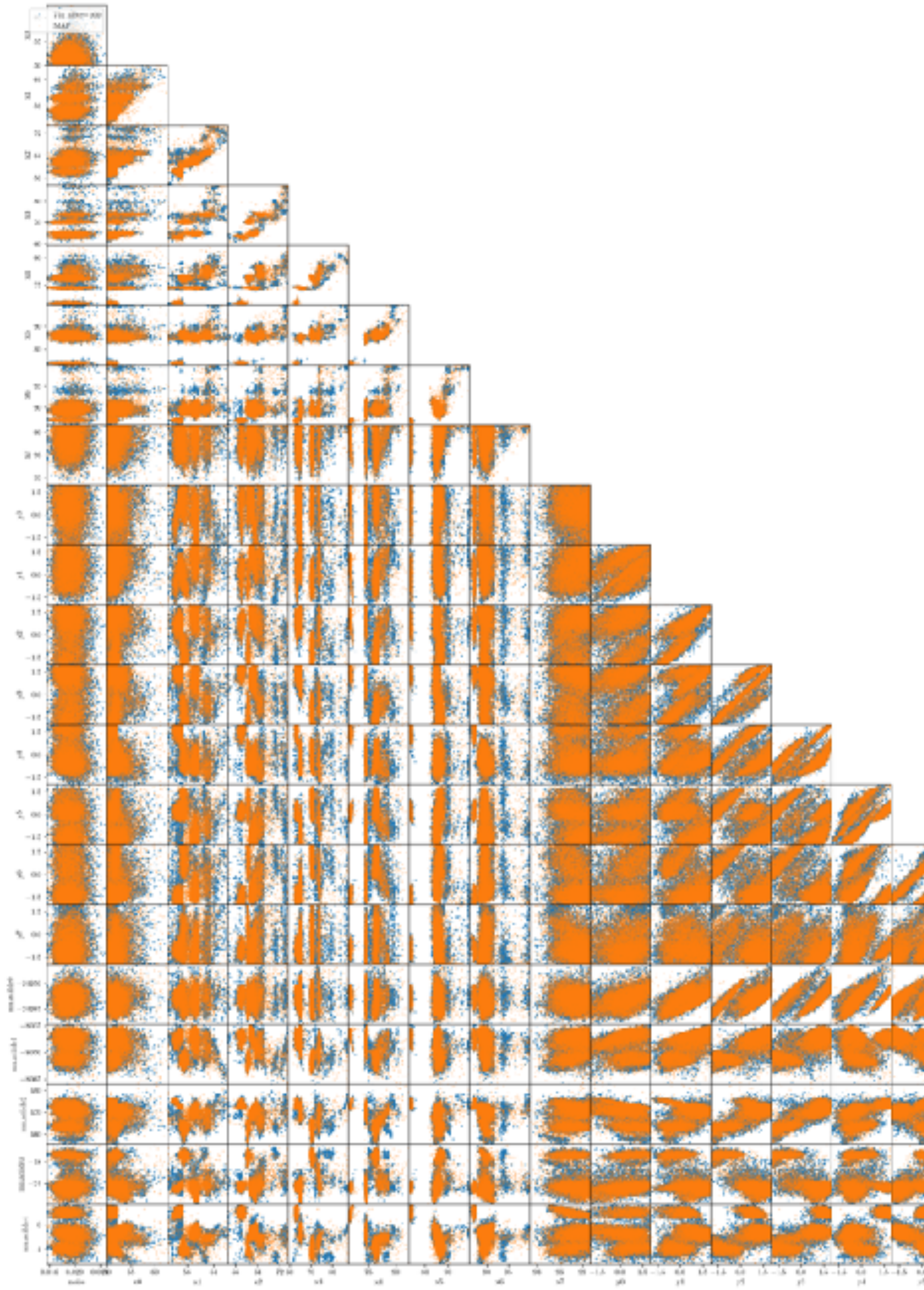
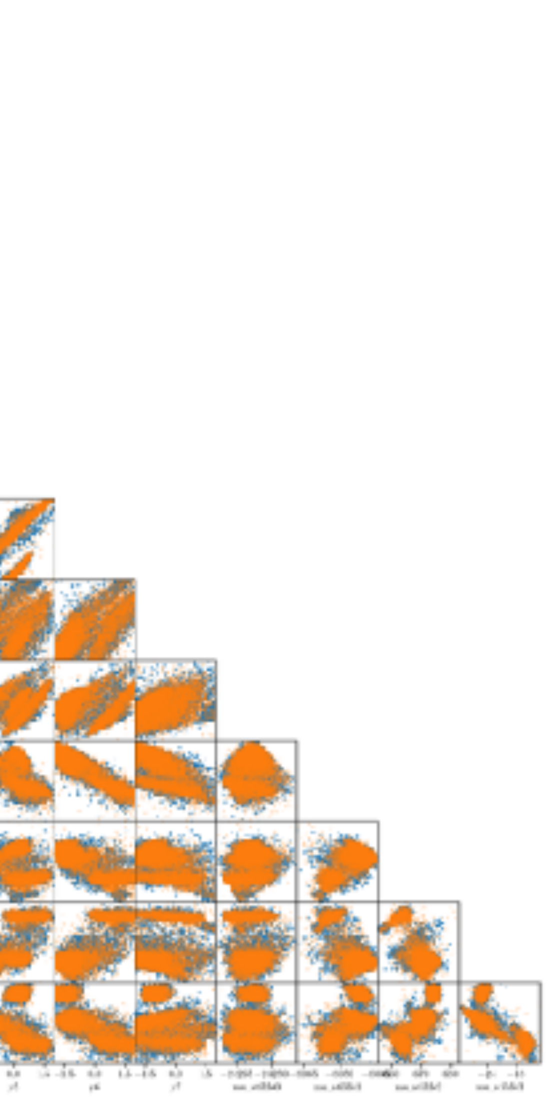
	High Resolution	Low Res -> `margarine` -> High Res
Likelihood Calls	2,498,813	741,723 (190,818 + 550,905, 30% of High Res run)
$\log Z$	-23.33	-25.20



Issues?

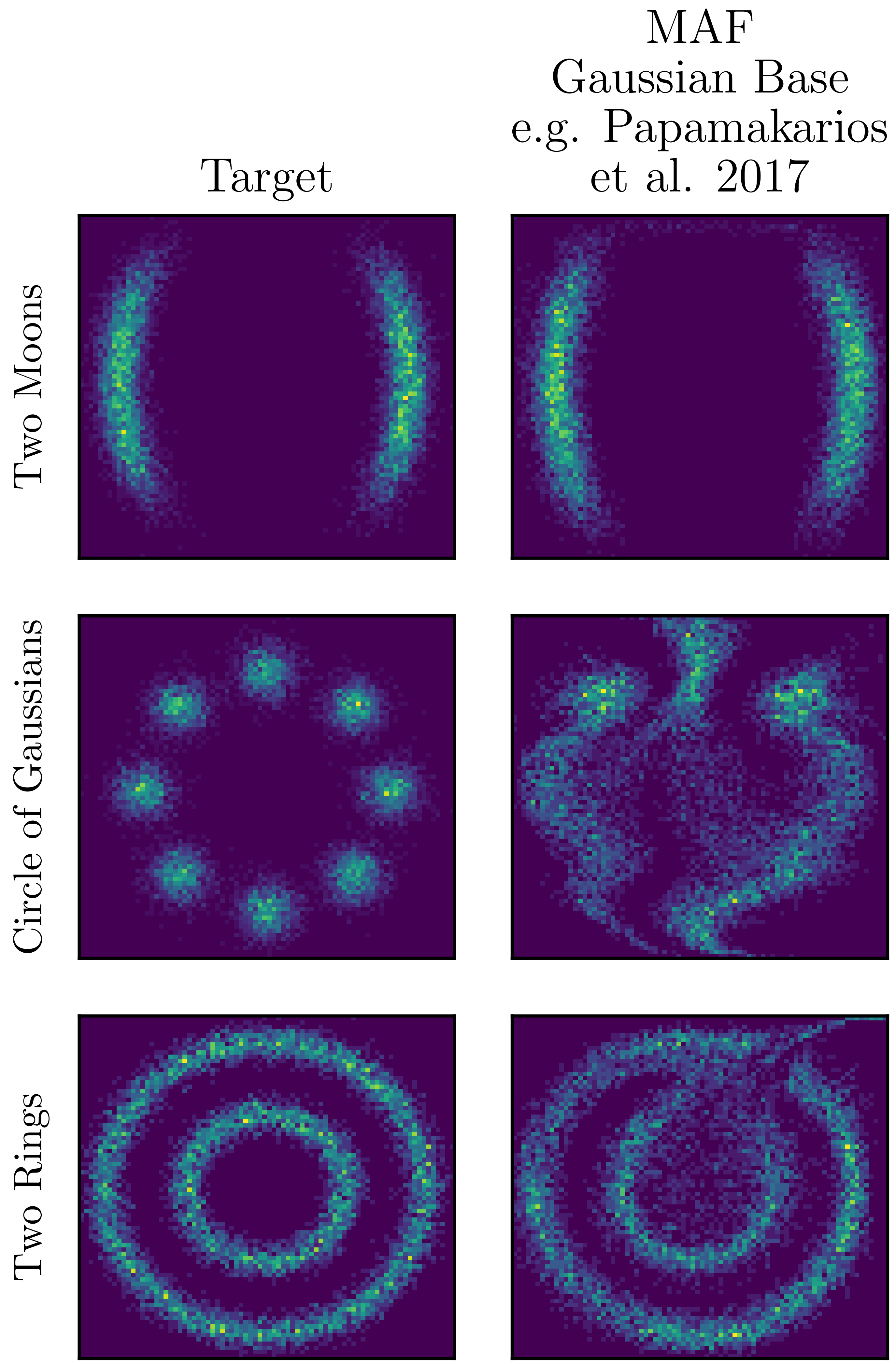
High dimensional distributions

- Normalizing Flows struggle to learn high dimensional distributions
- Exploring high dimensional problems in context of flex knot modelling with Stefan Heimersheim
- Potential to exploit independence of subspaces in the larger parameter space
- Train sets of MAFs on independent parts of parameter space and sample in unison

Multimodal Distributions

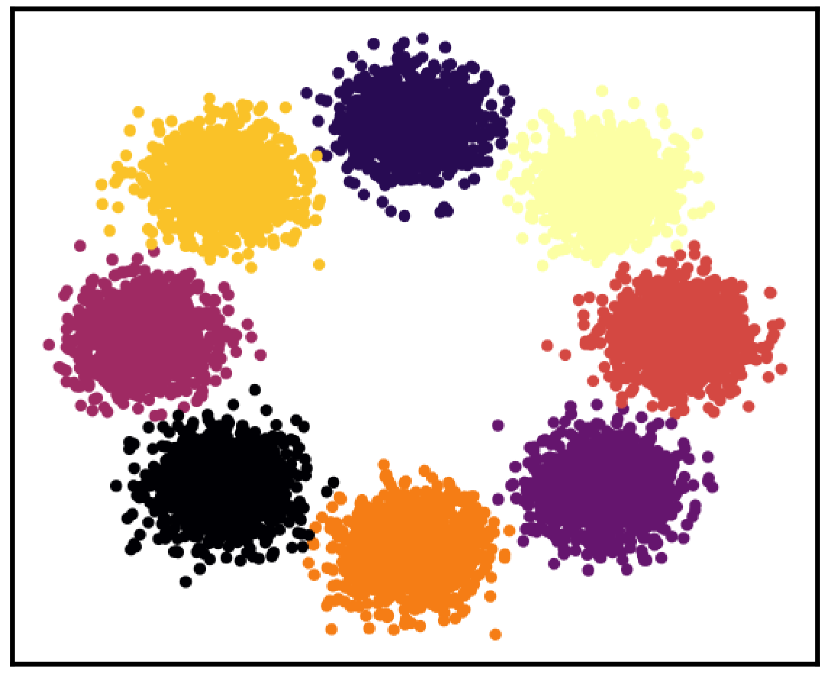
- Flows also struggle with multi-modal distributions
- Topology of the base distribution is different from the topology of the target distribution
- End up with bridges between the modes
- Many techniques have been developed to tackle this issue
- For example modifying the base distribution during training as in Stimper et al 2022 arXiv:2110.15828

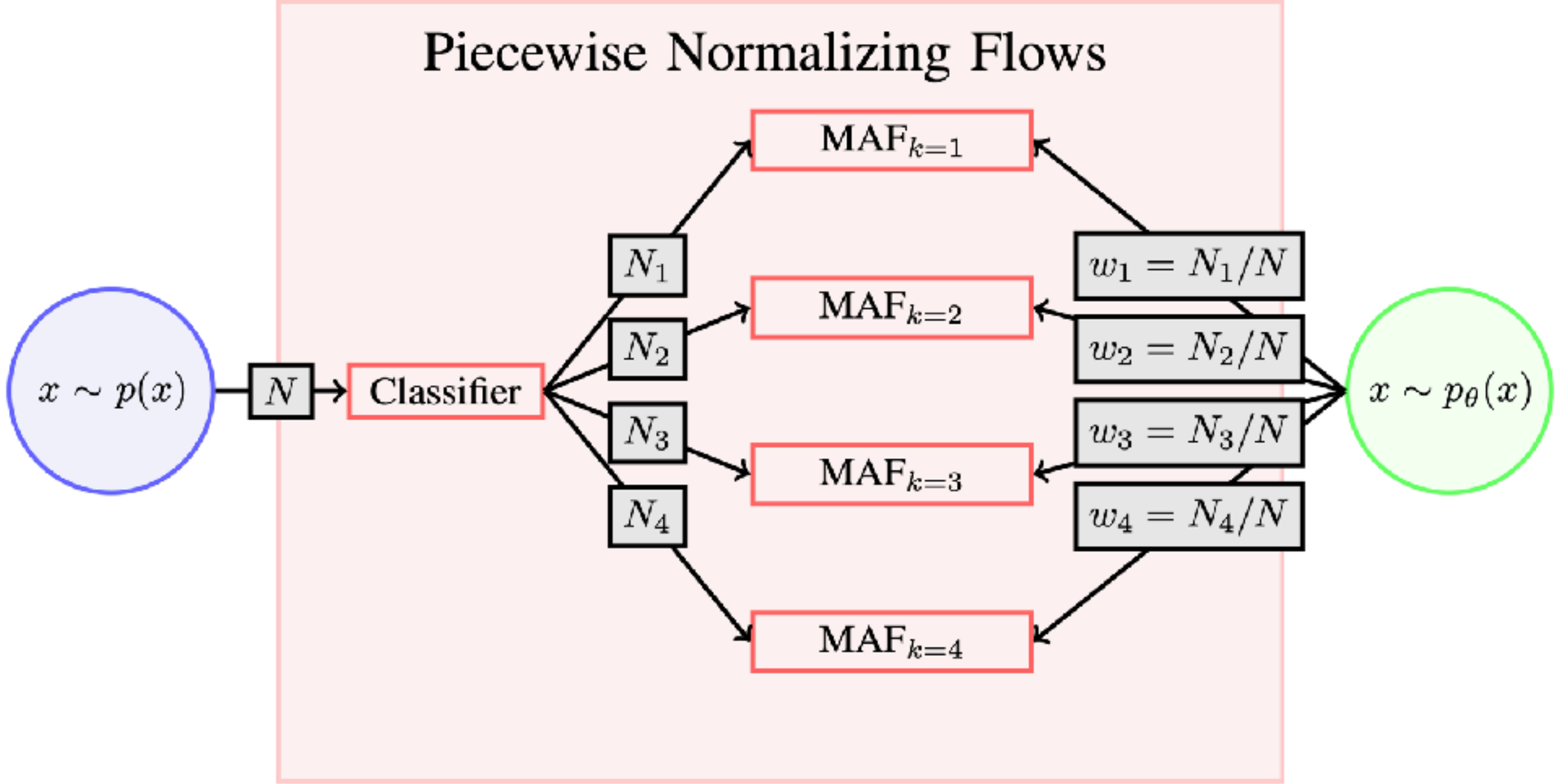


Piecewise Normalizing Flows

- Making progress with `margarine`
- Exploring the synergies between clustering algorithms and Normalizing Flows
- Divide the target into clusters with topologies closer to base distribution
- Train a MAF on each cluster
- Draw samples from MAFs based on size of cluster in target distribution
- Sum log-probabilities from each MAF

$k = 8, s = 0.667$



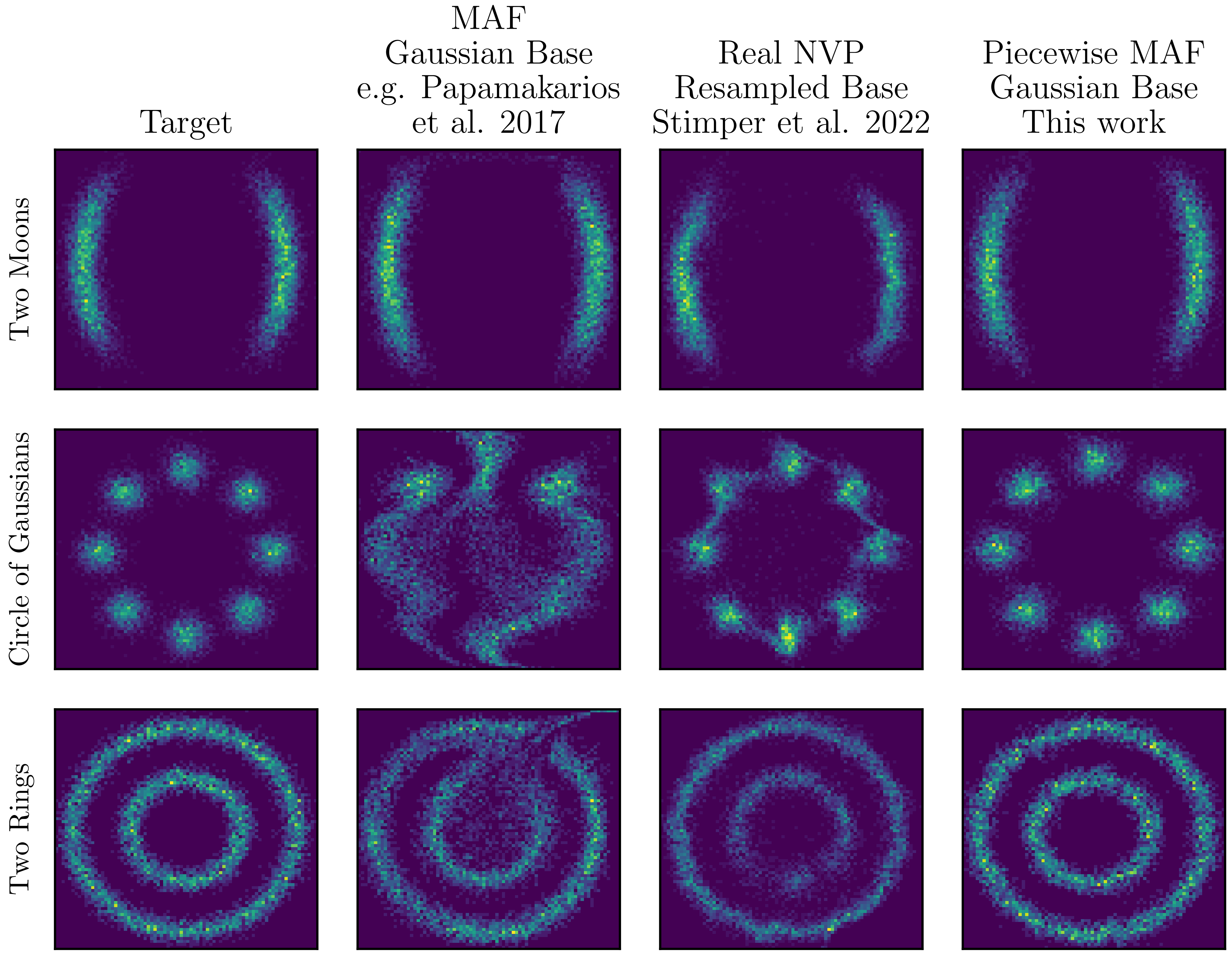


Piecewise Normalizing Flows

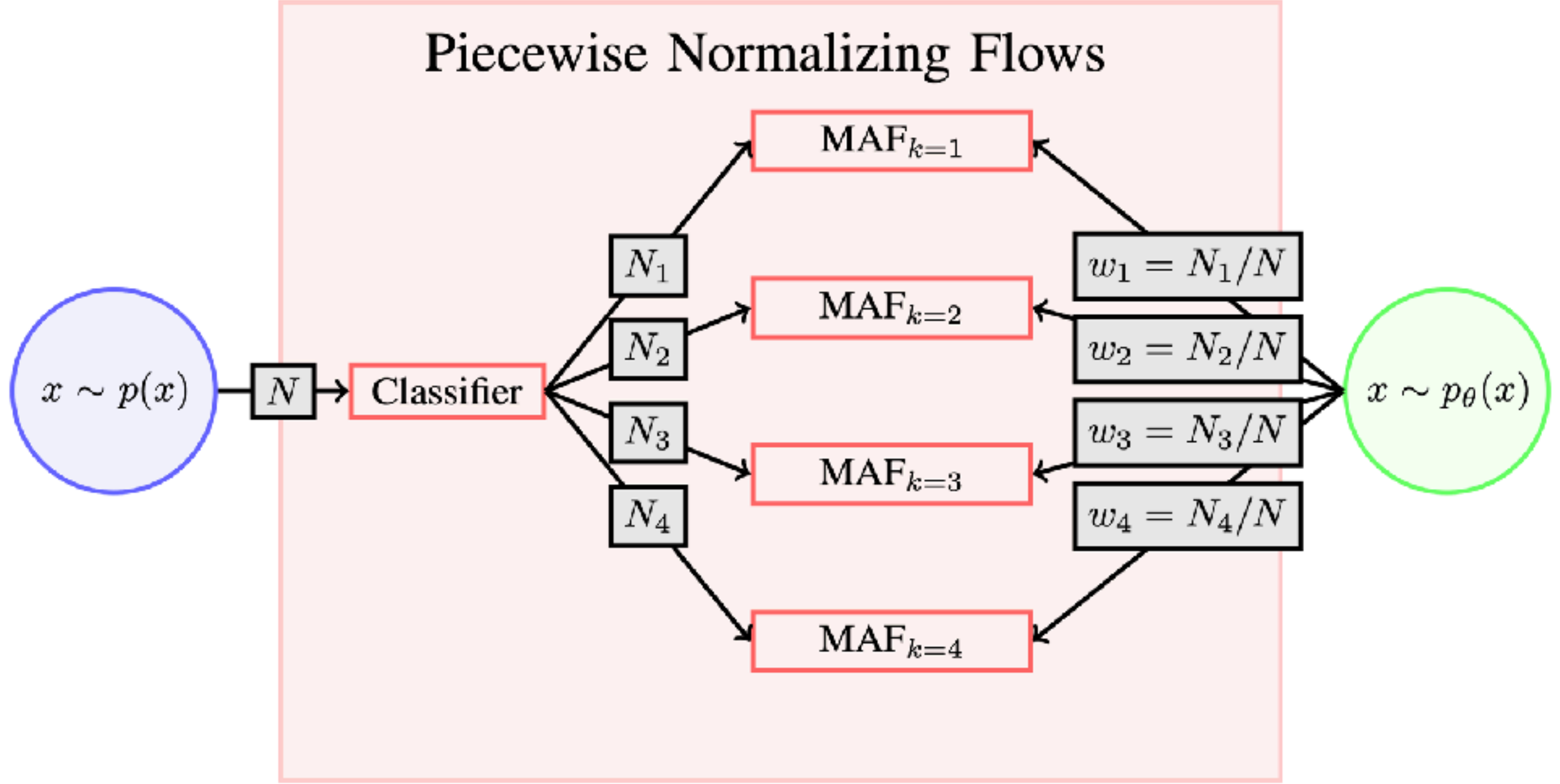
- Get better performance than existing algorithms
- Has the benefit that we can train each MAF in parallel
- Overcome the bridging issues seen with traditional approaches
- Implemented in `margarine` but parallelisation is still in the pipeline
- One to one mapping between the base and trained distribution plus a cluster number

$$\mathbb{R}^N_{\mathcal{N}(\mu=0,\sigma=1)} \Longleftrightarrow \mathbb{R}^N_{p_\theta(x)} \times \mathbb{N}$$

- Differentiable provided clustering algorithm is differentiable



Conclusions



- Normalizing Flows give us access to marginal probability distributions
- Allows us to calculate marginal Bayesian statistics
- Defined the marginal log-likelihood
- Enhanced joint analysis pipelines
- Potential for enhanced likelihood reweighting and enhanced Nested Sampling
- Challenges surrounding high dimensions and multi-modal distributions

Feel free to contact myself or Will if you think `margarine` could be useful in your work!

arXiv:2205.12841
arXiv:2207.11457
arXiv:2305.02930

https://github.com/htjb/margarine
https://github.com/htjb/piecewise_normalizing_flows