Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **536,** 190–202 (2025) https://doi.org/10.1093/mnras/stae2604
Advance Access publication 2024 November 21

EMUFLOW: normalizing flows for joint cosmological analysis

Arrykrishna Mootoovaloo,[1]★† Carlos García-García,[1] David Alonso[1] and Jaime Ruiz-Zapatero[2,3]

[1]*Oxford Astrophysics, Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK*
[2]*Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK*
[3]*Advanced Research Computing Centre, University College London, 90 High Holborn, London WC1V 6LJ, UK*

Accepted 2024 November 18. Received 2024 November 14; in original form 2024 September 6

ABSTRACT

Given the growth in the variety and precision of astronomical data sets of interest for cosmology, the best cosmological constraints are invariably obtained by combining data from different experiments. At the likelihood level, one complication in doing so is the need to marginalize over large-dimensional parameter models describing the data of each experiment. These include both the relatively small number of cosmological parameters of interest and a large number of 'nuisance' parameters. Sampling over the joint parameter space for multiple experiments can thus become a very computationally expensive operation. This can be significantly simplified if one could sample directly from the marginal cosmological posterior distribution of preceding experiments, depending only on the common set of cosmological parameters. We show that this can be achieved by emulating marginal posterior distributions via normalizing flows. The resulting trained normalizing flow models can be used to efficiently combine cosmological constraints from independent data sets without increasing the dimensionality of the parameter space under study. The method is able to accurately describe the posterior distribution of real cosmological data sets, as well as the joint distribution of different data sets, even when significant tension exists between experiments. The resulting joint constraints can be obtained in a fraction of the time it would take to combine the same data sets at the level of their likelihoods. We construct normalizing flow models for a set of public cosmological data sets of general interests and make them available, together with the software used to train them, and to exploit them in cosmological parameter inference.

Key words: methods: data analysis – methods: statistical – cosmology: cosmological parameters.

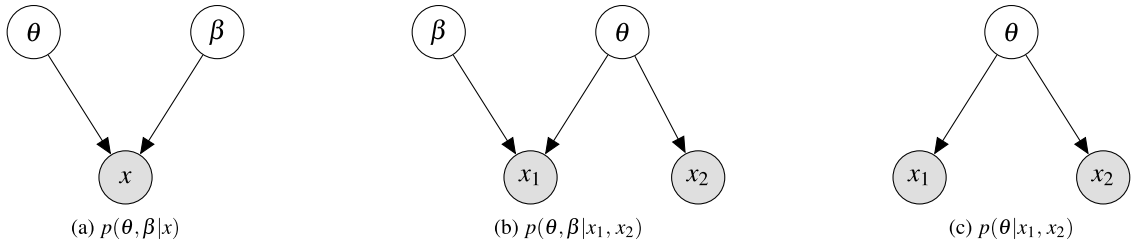
1 INTRODUCTION

Data analysis in cosmology is rapidly evolving. With data from past and current experiments such as Planck (Planck Collaboration VI 2020b), Kilo–Degree Survey (KiDS) (Asgari et al. 2021), and Dark Energy Survey (DES) (Abbott et al. 2022), as well as forthcoming surveys like Euclid (Amendola et al. 2018), Simons Observatory (Ade et al. 2019), Dark Energy Spectroscopic Instrument (DESI) (DESI Collaboration 2024), and the Vera Rubin Observatory, formerly known as the Large Synoptic Survey Telescope (LSST) (Ivezić et al. 2019) amongst others, there is a growing need for the development of tools that can accelerate the analysis of cosmological data.

Various techniques have been developed to accelerate computations in cosmology, depending on the tasks being investigated. For instance, generative models have been extensively used in field-level analysis in Cosmology. Kodi Ramanah et al. (2020) built a Generative Adversarial Network (GAN) emulator for low-resolution cosmological simulations. Tröster et al. (2019) explored deep generative models to identify an accurate representation of the large-scale distribution of gas and its temperature. Jamieson et al. (2024) also developed a field-level emulator for large-scale structure structure. Emulators can also be developed at the power-spectrum level, though this can be computationally expensive due to the need for numerous forward simulations and power spectrum computations in joint analyses. Recently, symbolic regression techniques have been used to derive mathematical expressions for linear and non-linear matter power spectra (Bartlett et al. 2024a, b). In the same spirit, Aricò et al. (2021) and Spurio Mancini et al. (2022) developed power-spectrum emulators based on deep learning. On the other hand, Mootoovaloo et al. (2022) developed a Gaussian Process (GP) emulator for linear and non-linear matter power spectra, which can also be used for computing weak lensing power spectra. While these techniques effectively accelerate computations, a major challenge is defining the region of parameter space before building the emulator. A naive broad prior can result in power spectra computations where the cosmological data does not constrain the parameters.

In short, the aim of these machine learning methods is to learn an effective model that is able to describe the data, using only the data as input. Data in this context can be any key quantity we are interested in, for example cosmological samples, bandpowers, and other forms of compressed data (Alsing, Wandelt & Feeney 2018; Mootoovaloo et al. 2020). On the one hand, we have generative models such as GAN, variational auto-encoders (VAE), normalizing flows, and GP which can learn the data directly. On the other hand,

★ E-mail: arrykrishna.mootoovaloo@physics.ox.ac.uk
† LSST-DA Catalyst Fellow



(a) $p(\theta, \beta|x)$ (b) $p(\theta, \beta|x_1, x_2)$ (c) $p(\theta|x_1, x_2)$

Figure 1. Directed acyclic graphs (DAGs) showing the typical inference problem in cosmology in panel (a). Panel (b) shows the DAG for a joint analysis in the case where the forward model in experiment 1 also has nuisance parameters, $\boldsymbol{\beta}$ and for experiment 2, we have access to an approximate distribution, $p(\boldsymbol{\theta}|\boldsymbol{x}_2)$. In panel (c), we have marginalized over all the nuisance parameters and we have approximate $p(\boldsymbol{\theta}|\boldsymbol{x}_1)$ and $p(\boldsymbol{\theta}|\boldsymbol{x}_2)$. Note that we are working with independent data sets, hence, there is no link between any two data sets, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

we have simulation-based inference (SBI) techniques which learn the parameters of the model by simulating data (see e.g. Bayesian optimization for likelihood-free inference – BOLFI (Leclercq 2018). While the two classes of machine learning methods (generative models and SBI) are related by the fact that they both adopt a probabilistic approach, their implementations and goals are different. For example generative models have a loss function to learn the data distribution, while SBI uses a forward model for the data and the goal is to learn the parameters of the model. In this work, we focus solely on generative model, in particular, normalizing flow models.

In Cosmology, numerous publicly available chains for cosmological and nuisance parameters have been obtained using Markov Chain Monte Carlo (MCMC) approaches from different data sets. The question is whether we can exploit these chains (rather than the likelihood and theory prediction codes that generated them) to perform joint analysis of different probes efficiently. A similar concept was investigated by Heavens et al. (2017a, b), who utilized publicly available MCMC chains to estimate the marginal likelihood. Moreover, Bevins et al. (2023) developed a technique which uses normalizing flows and kernel density estimators to learn marginal posterior distribution of the scientific parameters in cosmology. Bevins et al. (2022, 2024) further applied this technique in 21cm experiments. In this work, we demonstrate how normalizing flows can be employed to learn these marginal probability distributions and subsequently use them to perform joint analyses combining different experiments. This allows us to bypass having to sample computationally expensive and slow joint posterior distributions, as is often the standard approach.

Normalizing flows have been used in various applications in Cosmology. For example Alsing & Handley (2021) combined normalizing flow models with nested sampling. Recently, Srinivasan et al. (2024) developed a codebase, FLOWZ, to estimate the Bayesian evidence from posterior samples. Normalizing flows have also been used in the estimation of Bayesian evidence via the harmonic mean estimation (McEwen et al. 2021; Polanska et al. 2024). Recently, Taylor et al. (2024) studied the approach of training weighted ensembles of normalizing flows to emulate individual and joint distribution. Our contributions in this work are as follows. (1) We show that pre-trained normalizing flows can be used to effectively and precisely sample joint posterior distributions without increasing the dimensionality of the parameter space or the computational cost of including new complex likelihoods. We show that this is true even for combinations of experiments that are in relatively large tension with each other. (2) We also make many pre-trained models publicly available, together with an Application Programming Interface (API) that makes it easy for anyone to include them in their likelihoods.The main difference between our approach and that of Taylor et al. (2024) is our aim to employ the trained normalizing flows as emulated likelihoods that can be used as effective priors in the joint analysis of past experiments with new data within an MCMC scheme.

This paper is organized as follows. In Section 2, we describe the normalizing flow procedures before applying the concepts to simple, toy examples in Section 3. In Section 4, we apply the method to infer the cosmological parameters and nuisance parameters. We take two approaches, the first where a normalizing flow model is used as a prior and the second case, where we simply use two normalizing flows to sample the joint distribution. Furthermore, we use existing publicly available MCMC chains to construct these normalizing flow models, as discussed in Section 5. We also briefly cover how the code works in Section 6. We discuss our results in Section 4.3 before concluding in Section 7.

2 NORMALIZING FLOWS

Normalizing flows are a class of generative model which transform simple distributions into complex ones via invertible functions. They are efficient tools for density estimation and sampling. In effect, our goal is to employ normalizing flows to learn the density function of publicly available MCMC chains of cosmological interest over a few parameters of interest, marginalized over the rest. This is useful in the task of joint analysis as well as specifying a prior before sampling parameters of a model of choice.

2.1 Motivation

Suppose we have N experiments, each having its own set of cosmological parameters, $\boldsymbol{\theta}_i$, nuisance parameters, $\boldsymbol{\beta}_i$, and data $\boldsymbol{x}_i$. We are also assuming that the data, $\boldsymbol{x}_i$ is independent from each other. For simplicity, we will assume that we have a common set of cosmological parameters across all experiments. Let us also assume that we have samples of $\{\boldsymbol{\theta}_i, \boldsymbol{\beta}_i\}$, which are obtained by sampling the posterior of all parameters in each experiment. The marginalized posterior distribution of the cosmological parameters

$$p(\boldsymbol{\theta}_i|\boldsymbol{x}_i) \equiv \int p(\boldsymbol{\theta}_i, \boldsymbol{\beta}_i|\boldsymbol{x}_i)\, \mathrm{d}\boldsymbol{\beta}_i, \tag{1}$$

can be obtained by considering only the values of those parameters in the MCMC chain, ignoring the values of β_i.

Panel (a) of Fig. 1 shows the directed acyclic graph (DAG) for this set-up. $\boldsymbol{\theta}_i$ and $\boldsymbol{\beta}_i$ are the latent variables and $\boldsymbol{x}_i$ is the fixed data. We are now interested in finding the joint posterior of the cosmological parameters, $\boldsymbol{\theta}$ given the different experiments. In the first case, we can think of a scenario where we want to enforce a more informative prior in the analysis, for example a case where we have the likelihood for a large-scale structure data ($\boldsymbol{x}_1$) and we want to use the posterior distribution of cosmological parameters inferred by Planck ($\boldsymbol{x}_2$) as a

prior. In this case,

$$p(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2) = \int p(\boldsymbol{x}_1|\boldsymbol{\theta},\ \boldsymbol{\beta})\ p(\boldsymbol{\theta}|\boldsymbol{x}_2)\ p(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta} \tag{2}$$

where $p(\boldsymbol{x}_1|\boldsymbol{\theta},\ \boldsymbol{\beta})$ is the likelihood of $\boldsymbol{x}_1$ and $p(\boldsymbol{\theta}|\boldsymbol{x}_2)$ being a more informative prior based on the second data set, $\boldsymbol{x}_2$. See panel (b) in Fig. 1 for this set-up. In the third scenario, we can also have a case where we simply use nuisance-marginalized models for joint inference of the cosmological parameters. See panel (c) in Fig. 1.

If we were to do a joint analysis among the different experiments, the total dimensionality of the problem can become large. For example if we assume we have b cosmological parameters and each experiment E_i has c_i nuisance parameters, the total number of parameters is $b + \sum_i c_i$. Standard sampling schemes such as Metropolis–Hastings may struggle to learn the full posterior distribution of all parameters. Furthermore, as we incorporate more experiments into the analysis, it may become increasingly computationally intensive. Our proposal is to sample the parameters in each experiment (or use publicly available MCMC chains), followed by data fusion, which we discuss in the next section.

2.2 Data fusion

The process by which multiple data and knowledge is combined together is known as data fusion (Wu et al. 2024). In machine learning, a common practice to augment the knowledge of a single model is via federated learning. In this scenario, a model is trained on a set of data, generating a local agent. This agent can further be trained on another similar data set in a different location, thereby augmenting its knowledge and capability. In a Bayesian setting, each experiment has its own local estimate of its parameters. There are different ways in which this fusion process can be carried out, and its performance of this fusion process depends on the way the priors are used. For example in non-parametric Bayesian methods such as GP, techniques such as Product of Expert (PoE) and Bayesian committee machine (BCM) have been developed to fuse local estimates of the GP posterior (Tresp 2000). This is particularly helpful in scaling GP to millions of training points. In what follows, we will only cover parametric Bayesian methods, that is, a scenario where we have a forward model with its cosmological and nuisance parameters which are learnt from data.

In the first case, assuming a common prior, $p(\boldsymbol{\theta})$, the joint posterior is given by Bayes' rule as:

$$p(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N) = \frac{p(\boldsymbol{\theta})\ \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta})}{p(\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N)}, \tag{3}$$

and we have assumed that the joint likelihood can be factorized into their local, individual likelihood as a result of the conditional independence. Under this formalism, where the prior is common across all experiments, each experiment returns a local posterior in the data fusion process. One can also think of a scenario where each experiment has its local prior and we have a global prior for data fusion. In this case, we can write

$$p(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N) = Z\ p(\boldsymbol{\theta}) \prod_{i=1}^{N} \frac{p(\boldsymbol{\theta}|\boldsymbol{x}_i)}{p_i(\boldsymbol{\theta})}, \tag{4}$$

where Z is

$$Z = \frac{\prod_{i=1}^{N} p(\boldsymbol{x}_i)}{p(\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N)}, \tag{5}$$

and $p_i(\boldsymbol{\theta})$ is the local prior for each experiment. In the case where the data are independent from each other, $Z = 1$. These data fusion techniques are known as *conditionally independent likelihood* (CIL) data fusion (Wu et al. 2024).

On the other hand, it could also be possible that the N multiple different experiments, have been performed separately and we do not have access to the individual priors but only the local posteriors. One approach to data fusion in this scenario is the PoE, where these local posteriors are multiplied together to generate a global posterior given by

$$\tilde{p}(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N) = c \prod_{i=1}^{N} p(\boldsymbol{\theta}_i|\boldsymbol{x}_i), \tag{6}$$

where c is some normalization constant and $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N)$ is accurate compared to the true posterior, $p(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N)$ only when the assumption holds, that is, the case where the priors play an important role generating the global posterior. This data fusion technique is often referred to as the *conditionally independent posterior* (CIP) data fusion (Wu et al. 2024). Interestingly, in complex data analysis problems which involve expensive and non-linear models, it is highly unlikely that we can find the local, joint posterior of the cosmological parameters only (marginalized over the nuisance parameters), that is, $p(\boldsymbol{\theta}_i|\boldsymbol{x}_i)$. However, one can try to approximate this local joint posterior of the cosmological parameters via generative modelling frameworks. In this work, given that we have samples of $\boldsymbol{\theta}$, the density $p(\boldsymbol{\theta}_i|\boldsymbol{x}_i)$ is approximated using a normalizing flow model. Hence, the approximate joint posterior is:

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N) = k \prod_{i=1}^{N} p_{\mathrm{nf}}(\boldsymbol{\theta}_i|\boldsymbol{x}_i), \tag{7}$$

where $p_{\mathrm{nf}}(\boldsymbol{\theta}_i|\boldsymbol{x}_i)$ is the learned normalizing flow model for each experiment and k is just a normalization constant. Also note that $\hat{p}$ is different from $\tilde{p}$. The assumption that we can combine each individual posterior via the PoE rule, together with approximating the individual posterior with a normalizing flow model, can lead to a less accurate joint posterior compared to the true joint posterior, $p(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N)$. The approximate joint log-posterior is:

$$\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_N) = \sum_{i=1}^{N} \log p_{\mathrm{nf}}(\boldsymbol{\theta}_i|\boldsymbol{x}_i) + \log k. \tag{8}$$

If we have access to $\log p_{\mathrm{nf}}(\boldsymbol{\theta}_i|\boldsymbol{x}_i)$, we can draw samples from this approximate distribution and we can also compute the log-density. This is crucial because we can then (1) use the log-density for joint analysis and (2) use the learned density as a prior in a completely new cosmological data analysis problem.

Moreover, if we have a pre-trained normalizing flow model, it is also possible to combine it in the analysis of a new data set. If $\boldsymbol{x}_{\mathrm{new}}$ denotes the new data set and given a pre-trained flow model, $p_{\mathrm{nf}}(\boldsymbol{\theta}|\boldsymbol{x}_{\mathrm{old}})$ of an old data set, $\boldsymbol{x}_{\mathrm{old}}$, the new posterior due to the joint analysis is:

$$p(\boldsymbol{\theta},\ \boldsymbol{\beta}|\boldsymbol{x}_{\mathrm{new}},\ \boldsymbol{x}_{\mathrm{old}}) \propto p(\boldsymbol{x}_{\mathrm{new}}|\boldsymbol{\theta},\ \boldsymbol{\beta})\ p_{\mathrm{nf}}(\boldsymbol{\theta}|\boldsymbol{x}_{\mathrm{old}})\ p(\boldsymbol{\beta}), \tag{9}$$

where β are the nuisance parameters of the new experiment. $p(\boldsymbol{x}_{\mathrm{new}}|\boldsymbol{\theta},\ \boldsymbol{\beta})$ and $p(\boldsymbol{\beta})$ are likelihood and priors of the nuisance parameters in the new experiment, respectively. This is interesting for various reasons. $p_{\mathrm{nf}}(\boldsymbol{\theta}|\boldsymbol{x}_{\mathrm{old}})$ captures all the information about $\boldsymbol{x}_{\mathrm{old}}$ in the cosmological parameters. In this regard, we are enforcing a more informative prior on the cosmological parameters. Moreover, we no longer have to explicitly evaluate the likelihood due to the old data, which can be computationally expensive.

2.3 Normalizing flow theory

Given n samples of $\boldsymbol{\theta} \in \mathbb{R}^d$, a normalizing flow provides a simple way to construct a flexible distribution over the cosmological parameters. The idea is to express $\boldsymbol{\theta}$ as a transformation,

$$\boldsymbol{\theta} = f(\boldsymbol{z}) \qquad \boldsymbol{z} \sim p(\boldsymbol{z}), \tag{10}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is sampled from a distribution, $p(\boldsymbol{z})$. $p(\boldsymbol{z})$ is also known as the base distribution. These base distributions are usually simple distributions such as normal distribution or multivariate normal distribution.

The function f, also known as a bijector, has its own set of unknown parameters which we denote as ϕ. These unknown parameters are learnt via optimization (see Section 2.3.2 below). An important characteristic of f is that it should be invertible and both f and f^{-1} should be differentiable. This also implies that the function is bijective, that is, there is a one-to-one correspondence between elements in the domain of θ and elements in the codomain of $\boldsymbol{z}$. An important property of these types of transformation is that they are also composable. For example if we have two functions, f_1 and f_2, the composition $f_1 \circ f_2$ is also invertible and differentiable. The inverse is given by

$$(f_1 \circ f_2)^{-1} = f_2^{-1} \circ f_1^{-1}. \tag{11}$$

In general, it is a common practice to combine multiple transformations (bijectors), that is, $f = f_M \circ \ldots \circ f_2 \circ f_1$ and each bijector transforms $\boldsymbol{z}_{m-1}$ into $\boldsymbol{z}_m$ and $\boldsymbol{z}_M = \boldsymbol{\theta}$.

2.3.1 Change of variables

Let us consider a 1D example. Suppose we have the continuous random variable, x and its probability density function is $p(\theta)$. In order to change variables, we can write the following:

$$\int_\Theta p(\theta)\,\mathrm{d}\theta = \int_{\mathcal{Z}} p(z) \left|\frac{\mathrm{d}z}{\mathrm{d}\theta}\right| \mathrm{d}\theta, \tag{12}$$

where Θ and $\mathcal{Z}$ are the support of θ and z, respectively. Therefore, the probability density function of θ can be written as:

$$\log p(\theta) = \log p(z) + \log \left|\frac{\mathrm{d}z}{d\theta}\right|. \tag{13}$$

In the high dimensional case ($d > 1$), we can write the probability density function as:

$$\log p(\boldsymbol{\theta}) = \log p(\boldsymbol{z}) + \log \left|\det\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\theta}}\right)\right|, \tag{14}$$

where $\mathsf{J} \equiv \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{d\times d}$ is the Jacobian. Intuitively, we can think of the function, f as warping the space $\mathbb{R}^d$ by moulding the density $p(\boldsymbol{z})$ into $p(\boldsymbol{\theta})$. The absolute Jacobian determinant term accounts for the volume correction factor. If instead we have a series of transformation, that is, $f = f_M \circ \ldots \circ f_2 \circ f_1$, then

$$\log p(\boldsymbol{\theta}) = \log p(\boldsymbol{z}) + \sum_{m=1}^{M} \log \left|\det\left(\frac{\partial \boldsymbol{z}_{m-1}}{\partial \boldsymbol{z}_{\boldsymbol{m}}}\right)\right|, \tag{15}$$

where $\boldsymbol{z}_M = \boldsymbol{\theta}$, $\boldsymbol{z}_0 = \boldsymbol{z}$ and $\boldsymbol{z} = f_1^{-1} \circ \ldots \circ f_M^{-1}(\boldsymbol{\theta})$. Ideal normalizing flows should be expressive, invertible to ensure precise reconstruction of inputs, and have computationally efficient Jacobian determinants to enable quick evaluation and optimization of probability densities, making them amenable to model complex data distributions.

2.3.2 Optimization

Suppose $p_*(\theta)$ is the unknown target distribution and we have samples $\{\theta\}_{j=1}^n$. The Kullback–Leibler (KL) divergence between the target distribution, $p_*(\theta)$ and the flow-based model, $p(\boldsymbol{\theta}|\phi)$ is

$$\mathcal{L}(\phi) \equiv D_{KL}[p_*(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\phi)]. \tag{16}$$

Simplifying the above, we can write the KL-divergence as:

$$\begin{aligned}\mathcal{L}(\phi) &= -\int p_*(\boldsymbol{\theta}) \log p(\boldsymbol{\theta}|\phi)\, d\boldsymbol{\theta} + \text{constant} \\ &= -\mathop{\mathbb{E}}_{p_*(\boldsymbol{\theta})}\left[\log p(\boldsymbol{z}) + \log\left|\det\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\theta}}\right|\right] + \text{constant} \\ &\approx -\frac{1}{n}\sum_{j=1}^{n}\left[\log p(\boldsymbol{z}_j) + \log\left|\det\frac{\partial \boldsymbol{z}_j}{\partial \boldsymbol{\theta}_j}\right|\right].\end{aligned} \tag{17}$$

Recall that $z = f^{-1}(\boldsymbol{\theta};\phi)$, that is, the bijector f, modelled using using neural networks, has unknown parameters ϕ. Interestingly, minimizing the KL-divergence (equation 17) via the Monte Carlo method is equivalent to fitting the flow model via maximum likelihood estimation.

In short, once the normalizing flow model is trained, this means that we can do two important tasks. First, we can draw samples of $\boldsymbol{z}$ from the base distribution and transform them into $\boldsymbol{\theta}$ via $\boldsymbol{\theta} = f(\boldsymbol{z})$. Second, we can calculate the probability density at any point in the Θ domain via equation (14). The latter is the most crucial aspect for conducting the joint analysis in this work.

2.4 Flow models

Throughout this work, we will make use of Affine autoregressive flow model to learn the complex distribution, $p(\boldsymbol{\theta})$. Typically, the base distribution, $p(\boldsymbol{z})$ is mapped to the $p(\boldsymbol{\theta})$ distribution in the forward transformation and is reversed in the backward transformation. In short, in the forward transformation, $\boldsymbol{z} \to \boldsymbol{\theta}$ and in the reverse transformation, $\boldsymbol{\theta} \to \boldsymbol{z}$. A straightforward example is a scale-location transformation that meets the monotonicity criterion, that is,

$$z' = sz + t, \tag{18}$$

where s and t are the scale and location parameters, respectively. As described in Section 2.3.1, it is possible to apply a series of M bijective transformations. For a single training point $\boldsymbol{\theta}_j$,

$$\boldsymbol{z}_m^{(k)} = s_m^{(k)}\left(\boldsymbol{z}_{m-1}^{(<k)}\right) \cdot \boldsymbol{z}_{m-1}^{(k)} + t_m^{(k)}\left(\boldsymbol{z}_{m-1}^{(<k)}\right), \tag{19}$$

where k is the index denoting the kth dimension of the vector $\boldsymbol{\theta}$ and m is the mth transformation. Moreover, $z_0 = z$ and $\boldsymbol{z}_M = \boldsymbol{\theta}$. Both s and t are parametrized by neural networks. Moreover, for each transformation m and each training point, j, the absolute Jacobian determinant is

$$\log\,|\det(\mathsf{J}_m)| = \sum_{k=1}^{d} \log\left|s_m^{(k)}\left(\boldsymbol{z}_{m-1}^{(<k)}\right)\right|, \tag{20}$$

and the total absolute determinant due to M transformations is simply the sum of the above, that is,

$$\log\,|\det(\mathsf{J}_{\rm tot})| = \sum_{m=1}^{M} \log\,|\det(\mathsf{J}_m)|. \tag{21}$$

This type of affine transformation yields a lower triangular Jacobian matrix, allowing the determinant to be computed as the product of its diagonal elements. This process is repeated for all training points, and the loss is calculated using equation (17). By

Table 1. The different metrics (as discussed in Section 2.5) for the different analyses performed. The columns correspond to δ_μ, δ_σ, and δ_q for the joint analysis involving the CGG21 and P18 data sets. Columns 2 to 4 present the metrics when the P18 flow is used as a prior in the analysis, while the final three columns show the metrics when two normalizing flows are used. The last row gives the energy metric, which measures the degree of similarity of two distributions from their respective samples.

	P18 flow as a prior			CGG21 flow and P18 flow		
	δ_μ	δ_σ	δ_q	δ_μ	δ_σ	δ_q
Amplitude of density fluctuations, σ_8	0.000	0.006	0.002	0.002	0.021	0.273
CDM density, $\Omega_{\rm cdm}$	0.000	0.015	0.017	0.006	0.107	0.254
Baryon density, $\Omega_{\rm b}$	0.000	0.007	0.021	0.002	0.107	0.187
Hubble parameter, h	0.000	0.027	0.011	0.002	0.096	0.252
Scalar spectral index, n_s	0.000	0.031	0.041	0.000	0.036	0.012
Optical depth to reionization, τ	0.005	0.025	0.023	0.067	0.052	0.318
Energy distance, $\tilde{D}(p,\ \hat{p})$		~ 0.001			~ 0.01	

minimizing the KL-divergence, the neural network parameters, ϕ can be optimized. In this work, we employ three transformations, each with a dense architecture consisting of three layers, each having 32 hidden units and using the `tanh` activation function.

If we are given a test point, $\boldsymbol{\theta}_{\rm test}$ and we want to compute the log-density, it is mapped to $\boldsymbol{z}$ via the reverse transformations, f_m^{-1} and the log-density of the base distribution is calculated. Moreover, using equation (21), the log-determinant of the Jacobian is computed, followed by the computation of $\log p(\boldsymbol{\theta}_{\rm test})$. On the other hand, if we want to draw N samples from the normalizing flow model, N samples from the base distributions are drawn and the forward transformations are applied to map them to the $\boldsymbol{\theta}$ space.

2.5 Metrics

In order to quantify the difference between the approximate posterior (built upon the normalizing flow models) and the known joint posterior, we can compute metrics related to the statistics of marginalized posterior distribution of each parameter in 1D. For example we can compare the mean, $\mu_{\rm nf}$, obtained using the normalizing flow models with the expected mean, μ, that is,

$$\delta_\mu = \left| \frac{\mu - \mu_{\rm nf}}{\mu} \right|. \tag{22}$$

This essentially gives a measure of how accurate the samples from the normalizing flow model is compared to known samples from the joint posterior. Moreover, we also compare the width of the distribution, that is, the standard deviation using

$$\delta_\sigma = \frac{|\sigma - \sigma_{\rm nf}|}{\sigma}. \tag{23}$$

This effectively quantifies whether the precisions of the two set of samples are comparable. We can also take the difference of the means divided by the quadrature sum of the errors in both experiments, that is,

$$\delta_q = \frac{|\mu - \mu_{\rm nf}|}{\sqrt{\sigma^2 + \sigma_{\rm nf}^2}}. \tag{24}$$

Note that these metrics apply only in the 1D case. As discussed by Lemos et al. (2021), there is no universal method for quantifying the differences in multidimensional parameter spaces. However, the lower the values of these metrics, the better the reconstructed posteriors with the normalizing flow models. The question is whether we can have a metric to assess the similarity of the two distributions (known joint posterior and the joint posterior due to the normalizing flow models). One could use the analytic expression for the KL divergence or the Bhattacharyya distance or other variant such as the Jensen–Shannon (JS) divergence to quantify the similarity of the two distributions. However, these expressions apply only in the multivariate normal case and in our case, we can have non-Gaussian-like posteriors – see Fig. A1.

In our case, we have samples from the posteriors and one option is to use the maximum mean discrepancy (MMD) which is commonly used in GANs (Bińkowski et al. 2018). The idea is to embed the distributions into the reproducing kernel Hilbert space (RKHS) and measure the distance between the means in the embedded space, that is,

$$\mathrm{MMD}(p, \hat{p}) = \left\| \mu_p - \mu_{\hat{p}} \right\|_{\mathcal{H}}. \tag{25}$$

For example we can use the Gaussian kernel to embed the samples and compute the distance using equation (25). However, the resulting distance can be inconsistent due to its dependence on the bandwidth of the Gaussian kernel. Instead, one can use the energy distance which does not depend on any hyperparameter at all. It is given by

$$D^2(p, \hat{p}) = 2\mathbb{E}\left\| \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right\| - \mathbb{E}\left\| \boldsymbol{\theta} - \boldsymbol{\theta}' \right\| - \mathbb{E}\left\| \hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}' \right\|, \tag{26}$$

where $\|\cdot\|$ is the Euclidean norm. $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are samples from p and $\hat{p}$, respectively. The energy metric is inspired by Newton's concept of gravitational potential energy, where the potential energy becomes zero when the gravitational centres of two particles coincide. If the two distributions (p and $\hat{p}$) are exactly the same, then $D = 0$. However, as argued by Rizzo & Székely (2016), the above distance statistics is not standardized and in order to interpret the value, one can use

$$\tilde{D}(p, \hat{p}) = \frac{2\mathbb{E}\left\| \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right\| - \mathbb{E}\left\| \boldsymbol{\theta} - \boldsymbol{\theta}' \right\| - \mathbb{E}\left\| \hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}' \right\|}{2\mathbb{E}\left\| \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \right\|}, \tag{27}$$

where $0 \leq \tilde{D}(p, \hat{p}) \leq 1$. A low value of $\tilde{D}(p, \hat{p})$ indicates a higher degree of similarity between the distributions. There is no clear consensus on what constitutes a good energy metric; relative comparisons are generally more meaningful. We take a random set of 3000 samples from each distribution and compute $\tilde{D}(p, \hat{p})$ in the two experiments we have performed. The results are quoted in Table 1.

3 TOY EXAMPLES

In this section, we will look at two examples (1D and 2D) to demonstrate (1) how the normalizing flow model works and (2) how it can be used to sample the joint posterior of parameters of interest without requiring the original data sets and likelihoods.

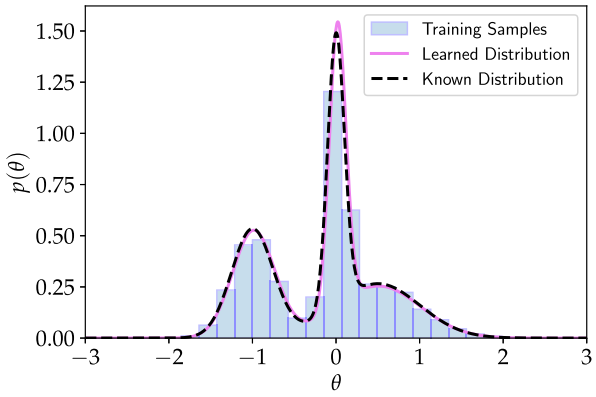


Figure 2. The plot shows the training samples in blue. They are generated using a mixture of three normal distributions, with means, $[-1.0, 0.5, 0.0]$ and standard deviations $[0.25, 0.50, 0.10]$. Therefore, $p(\theta) = \sum_{i=1}^{3} w_i \mathcal{N}(\mu_i, \sigma_i)$, where $w_i = \frac{1}{3}$ is fixed. The probability distribution learned by the normalizing flow model is shown in violet, while the dashed black curve shows the known distribution. The flow model accurately captures the distribution of the generated samples. See explanation in Section 2.3.1 for further details on the implementation.

3.1 1D distribution

Let us consider a mixture of three Gaussian distributions,

$$p(\theta) = \sum_{i=1}^{3} w_i \mathcal{N}\left(\mu_i,\, \sigma_i^2\right), \tag{28}$$

where $\sum_{i=1}^{3} w_i = 1$. The means and the standard deviations assumed are $\boldsymbol{\mu} = (-1.0,\, 0.5,\, 0.0)$ and $\boldsymbol{\sigma} = (0.25,\, 0.50,\, 0.10)$. We will assume a uniform distribution as the base distribution, that is, $p(z) = \mathcal{U}[0,\, 1]$. We will define the bijector, $z = f^{-1}(\theta)$ as a linear combination of cumulative density function, $\Phi(\mu,\, \sigma^2)$, of the normal distributions, that is,

$$z = \sum_{c=1}^{C} w_c \Phi\left(\theta;\, \mu_c,\, \sigma_c^2\right), \tag{29}$$

where C is the number of components which we are free to choose. In this case, we fix $C = 3$. In total, there are nine parameters in this model: $\{\mu_c,\, \sigma_c,\, w_c\}_{c=1}^{3}$. The derivative of the above function with respect to θ is analytical and can be written as

$$\frac{\mathrm{d}z}{\mathrm{d}\theta} = \sum_{c=1}^{C} w_c \mathcal{N}\left(\theta;\, \mu_c,\, \sigma_c^2\right). \tag{30}$$

We draw 10 000 samples from the true underlying distribution (equation 28) and fit for the nine parameters $\{\mu_c,\, \sigma_c,\, w_c\}_{c=1}^{3}$ by maximizing equation (13) (equivalent to minimizing the negative of the log density) using all the samples. The final probability density learnt is shown in violet in Fig. 2. The same technique explained for this toy 1D example can be extended to higher dimensional scenarios, with the exception, that the bijector is now composed of neural network blocks to accommodate for more expressive functions.

3.2 Gaussian linear model and banana posterior

Before building the normalizing flow models for complex posteriors in Cosmology, in this example, we show that we can recover the joint posterior distribution, solely using the normalizing flow models, without needing the original data sets and likelihood functions.

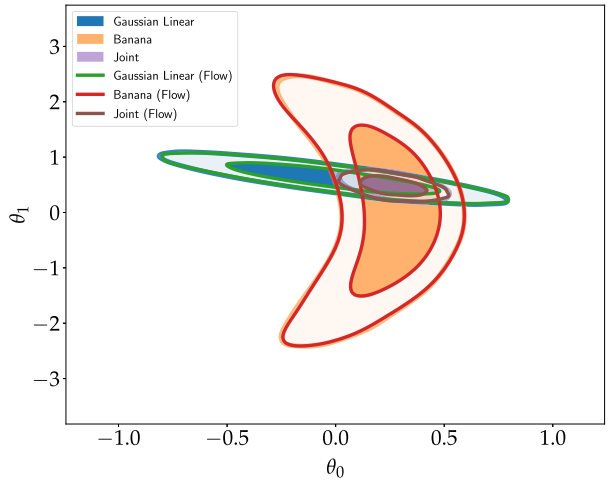


Figure 3. The figure shows the joint posterior distribution of a Gaussian posterior, obtained from a Gaussian Linear Model and a banana posterior. See Section 3.2 for implementation details. The blue and orange colours show the posteriors of the two parameters θ_0 and θ_1, sampled using MCMC, for the Gaussian Linear Model and the banana, respectively. The normalizing flows are built using these samples and are shown in green and red, respectively. The purple shaded region shows the joint distribution using the individual likelihoods, while the brown contour shows the joint distribution using only the normalizing flow models.

Let us consider a Gaussian Linear Model (GLM) of the form $f(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$. The fiducial point is $\boldsymbol{\theta} = [0.25,\, 0.25]$ and we generate data points, $y = f(x; \boldsymbol{\theta}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0,\, 1)$. The next function we consider is a banana-shape posterior, whose functional form is $g(\boldsymbol{\theta}) = \theta_0 + 0.1\theta_1^2$. As in the GLM, we generate 100 data points, that is, $w = g(\boldsymbol{\theta}) + \epsilon$. We assume independent normal priors – with mean centred on zero and standard deviation equal to one – on the parameters, $\boldsymbol{\theta}$. $p(\boldsymbol{y}|\boldsymbol{\theta})$ and $p(\boldsymbol{w}|\boldsymbol{\theta})$ are the Gaussian likelihoods for the two data sets, $\boldsymbol{y}$ and $\boldsymbol{w}$, respectively.

We sample the posterior of the individual data using EMCEE (Foreman-Mackey et al. 2013) and the joint posterior between θ_0 and θ_1 is shown in Fig. 3. The blue one corresponds to the GLM while the orange one corresponds to the banana function. As explained in Section 2.1, we can also do a joint analysis by combining the likelihoods, that is,

$$p(\boldsymbol{\theta}|\boldsymbol{y},\, \boldsymbol{w}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})\, p(\boldsymbol{w}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}). \tag{31}$$

This joint posterior is shown in Fig. 3 in purple. Given that we have MCMC samples, that is, $\boldsymbol{\theta}_y \leftarrow p(\boldsymbol{\theta}|\boldsymbol{y})$ and $\boldsymbol{\theta}_w \leftarrow p(\boldsymbol{\theta}|\boldsymbol{w})$, we can use a subset of these samples to fit a normalizing flow model to learn the posterior probability distribution independently. We first apply an affine transformation (rotation and translation) to the original samples, that is,

$$\boldsymbol{\theta}' = \mathsf{L}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}), \tag{32}$$

where L is the Cholesky factor of the covariance of the samples, $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ is the mean of the samples. We then choose independent normal distributions, $p(z;\, z_{\rm med},\, \sigma')$, where $z_{\rm med}$ and σ' are the median and standard deviation of the $\boldsymbol{\theta}'$ samples, respectively. We choose the median because posteriors may have complex shapes, for example the banana posterior, and hence the median is a more representative measure of central tendency. Note that the normalizing flow will output the density of $p(\boldsymbol{\theta}')$. We can draw samples from the original distribution by first sampling, $\boldsymbol{\theta}'$, followed by applying the inverse transformation, that is, $\boldsymbol{\theta} = \mathsf{L}\boldsymbol{\theta}' + \boldsymbol{\mu}$. We can calculate the density

using

$$p(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}')}{|\det(\mathrm{L})|}, \tag{33}$$

where $|\det(\mathrm{L})|$ is the absolute determinant of the Cholesky factor and accounts for the volume correction factor.

Once the normalizing flow models are trained, we can draw samples in green and red for the GLM and banana functions, respectively, in Fig. 3. By inspecting the blue and green posteriors for the GLM, and the orange and red posteriors for the banana function, it is evident that the flow models excel at learning the distribution. Our next task is to learn the joint distribution using the normalizing flow models only, that is, $p(\boldsymbol{\theta}|\boldsymbol{y},\ \boldsymbol{w}) \propto p_{\rm nf}(\boldsymbol{\theta}|\boldsymbol{y})\, p_{\rm nf}(\boldsymbol{\theta}|\boldsymbol{w})$. We sample this joint distribution using EMCEE and the joint posterior is shown in brown in Fig. 3. This is an important result as it demonstrates the ability to recover the joint distribution using the flow models alone, with MCMC samples from individual experiments serving as training points.

4 VALIDATION

An obvious application of emulated marginalized posterior distributions using normalizing flows is the possibility of using them as effective priors in the joint analysis of past experiments with new data sets without having to sample the full parameter space of the past data sets (including their nuisance parameters). The aim of this section is validating this approach, applying it to two real data sets that are in mild tension with one another (and for which, therefore, the normalizing flow emulator must be able to capture the outskirts of the distributions). We present the two data sets used [a large suite of large-scale structure data from García-García et al. (2021), and CMB data from *Planck*], the methodology used for parameter inference, and the results of this validation exercise.

4.1 Data

We make use of two cosmological data sets. The first one is the 2018 *Planck* data set (P18 hereafter), combining auto and cross-correlations between temperature and E-mode polarization. We use data from the COMMANDER component separation algorithm for the low-ℓ data ($2 \leq \ell \leq 29$) and the PLIK likelihood for the high-ℓ data in the range $30 \leq \ell \leq 2508$ for TT and $30 \leq \ell \leq 1996$ for TE and EE (Planck Collaboration V 2020a). We also include the CMB lensing autocorrelation, considering the range of scales $8 \leq L \leq 400$ (Planck Collaboration VI). We generate MCMC chains using the public likelihoods implemented in COBAYA (Torrado & Lewis 2021), marginalizing over nuisance parameters with priors as recommended in Planck Collaboration V (2020a).

The second data set consists of a large combination of projected large-scale structure data, analysed in García-García et al. (2021) (CGG21 from now on). Details about the data set itself and the model used to analyse it are described in detail in CGG21, and we provide only a short summary here. The data set is composed of the angular power spectrum of the auto and cross-correlation of galaxy clustering from DES-Y1 RedMaGiC (Rozo et al. 2016), DESI Legacy Survey (DELS) (specifically, the sample defined in Hang et al. 2021), and extended Baryon Oscillation Spectroscopic Survey (eBOSS) quasars (QSO) targets (Hou et al. 2021); weak lensing from the DES-Y1 METACALIBRATION sample (Zuntz et al. 2018) and KiDS-1000 (Asgari et al. 2021); and Cosmic Microwave Background (CMB) lensing from *Planck* 2018 (Planck Collaboration VIII 2020c).We only consider correlations between pairs of data sets that have non-zero sky overlap, we discard cross-correlations between different galaxy clustering samples, as well as the CMB lensing autocorrelation. To avoid modelling the covariance between DELS and DES RedMaGiC, all the region declination $\delta < -36$ deg in DELS was removed.

The CGG21 data set was analysed starting at the catalogue level in order to ensure a consistent measurement and analysis pipeline for the different data sets, and to properly account for the correlations between them. Maps of all large-scale structure tracers were generated using with a HEALPIX resolution parameter $N_{\rm side} = 4096$, and power spectra and their covariance were estimated using the pseudo-C_ℓ approach with NAMASTER (Alonso et al. 2019) method and the Gaussian part of the covariance with the improved Narrow Kernel Approximation. Power spectra involving galaxy clustering were analysed on scales $k < 0.15\,\mathrm{Mpc}^{-1}$, and weak-lensing data was analysed with a small-scale cut $\ell < 2048$. Different large-scale cuts were used for different tracers depending on the evidence of large-scale systematics in them.

The model used to analyse these data in CGG21 includes 40 different nuisance parameters: linear biases for all clustering samples (11 parameters), multiplicative biases for all comic shear samples (9 parameters), redshift distribution shifts for all photometric samples (18 parameters), and intrinsic alignment amplitude and evolution parameters (2 parameters). Of these, the multiplicative bias parameters and the redshift distribution shifts are marginalized over analytically using the methods described in Hadzhiyska et al. (2020) and Ruiz-Zapatero et al. (2023), leaving 13 nuisance parameters to marginalize over at the level of the likelihood.

As in CGG21, we compute the linear matter power spectrum $P_{\rm mm}$ with the Boltzmann code CLASS (Blas, Lesgourgues & Tram 2011) and the non-linear correction with HALOFIT (Takahashi et al. 2012). The kernels and angular power spectra are computed with the Core Cosmology Library (CCL) (Chisari et al. 2019).

Finally, we use CMB data from Planck 2018. We use the public likelihoods, as implemented in COBAYA. In particular, we use the auto and cross-correlations of the temperature (T) and polarization (E) fields. We use the data from the COMMANDER component separation algorithm for the low-ℓ data ($2 \leq \ell \leq 29$) and the PLIK likelihood for the high-ℓ data in the range $30 \leq \ell \leq 2508$ for TT and $30 \leq \ell \leq 1996$ for TE and EE Planck Collaboration V (2020a). We marginalize over the nuisance parameters with priors as recommended in (Planck Collaboration V 2020a). Finally, we include the CMB lensing autocorrelation power spectrum, considering the range of scales $8 \leq L \leq 400$ (Planck Collaboration VI 2020b). We will denote the CGG21 data set as $\boldsymbol{x}_{\rm cgg21}$ and the Planck data as $\boldsymbol{x}_{\rm p18}$.

4.2 Parameter inference

We run MCMC chains to sample the individual likelihoods of the CGG21 and P18 data sets. For the CGG21 analysis, we have a set of 13 nuisance parameters, which we denote as $\boldsymbol{\beta}$. In addition to these, we consider five ΛCDM cosmological parameters, which we denote $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \{\sigma_8,\ \Omega_c,\ \Omega_b,\ h,\ n_s\}.$$

For the P18 analysis, there are $\sim$20 nuisance parameters, and the cosmological parameters also include the optical depth to reionisation τ. We use top-hat, uninformative flat priors on all cosmological parameters.

Having generated chains for each individual experiment, we train two normalizing flows to recover the marginalized posterior distribution of cosmological parameters in each data set. With both

flows at hand, we can now consider three different set-ups to sample the joint likelihood of both experiments:

(i) The exact joint cosmological constraints, obtained from the product of the P18 and CGG21 likelihoods marginalized over nuisance parameters:

$$p(\boldsymbol{\theta}, \tau|\boldsymbol{x}_{\rm p18}, \boldsymbol{x}_{\rm cgg21}) \propto \int d\boldsymbol{\beta}_{\rm p18}\, d\boldsymbol{\beta}_{\rm cgg21} \\ p(\boldsymbol{x}_{\rm p18}|\boldsymbol{\theta}, \tau, \boldsymbol{\beta}_{\rm p18}) p(\boldsymbol{x}_{\rm cgg21}|\boldsymbol{\theta}, \boldsymbol{\beta}_{\rm cgg21}) \\ p(\boldsymbol{\beta}_{\rm p18}) p(\boldsymbol{\beta}_{\rm cgg21}) p(\boldsymbol{\theta}, \tau). \quad (34)$$

(ii) The product of the true CGG21 likelihood with the normalizing flow for the marginalized P18 distribution, using it as an effective prior:

$$p(\boldsymbol{\theta}, \tau|\boldsymbol{x}_{\rm p18}, \boldsymbol{x}_{\rm cgg21}) \propto p_{\rm nf}(\boldsymbol{\theta}, \tau|\boldsymbol{x}_{\rm p18}) \\ \int d\boldsymbol{\beta}_{\rm cgg21}\, p(\boldsymbol{x}_{\rm cgg21}|\boldsymbol{\theta}, \boldsymbol{\beta}_{\rm cgg21}) p(\boldsymbol{\beta}_{\rm cgg21}). \quad (35)$$

This set-up has the advantage that we do not need to evaluate the relatively expensive theoretical model and likelihood of the P18 data, and that we do not increase the dimensionality of the parameter space by combining it with CGG21.

(iii) The product of normalizing flows for both CGG21 and P18:

$$p(\boldsymbol{\theta}, \tau|\boldsymbol{x}_{\rm cgg21}, \boldsymbol{x}_{\rm p18}) \propto \frac{p_{\rm nf}(\boldsymbol{\theta}|\boldsymbol{x}_{\rm cgg21})\, p_{\rm nf}(\boldsymbol{\theta}, \tau|\boldsymbol{x}_{\rm p18})}{p(\boldsymbol{\theta}, \tau)}. \quad (36)$$

As described in CGG21, analysed within the model described above, the CGG21 data set exhibits tension with P18 in the value of $S_8 \equiv \sigma_8 \sqrt{\Omega_m/0.5}$ at the level of 3.5σ. The normalizing flow models must therefore capture the tails of both distributions with sufficient accuracy for them to recover the correct joint posterior distribution. Quantifying this is the goal of the next section.

4.3 Validation results

Let us start by studying the performance of the normalizing flows generated for each experiment individually. The 1D and 2D marginalized constraints obtained from the P18 chain (black contours) and from its normalizing flow (green contours) are shown in Fig. 4. The flow is able to recover the marginalized posterior with excellent accuracy. A similarly accurate flow density is recovered for the CGG21 data. We used 2×10^4 training points, randomly selected from the MCMC chain, to train the normalizing flow model. Since the models depend only on five or six parameters training them is very quick, taking $\sim$2 min on a desktop computer. After training, generating samples from the trained flow is nearly instantaneous.

Having access to the flow-based emulators for the individual marginalized posterior distributions, we can now ask ourselves whether these are accurate enough to be used in the joint analysis of different experiments. In a typical scenario, we want to combine the cosmological constraints obtained from a previous legacy experiment with data from a new experiment, for which the model depends on a given set of nuisance parameters, in addition to the common cosmological parameters. In this case, the emulated marginalized posterior for the legacy experiment may be used as an effective prior in the posterior distribution of the new data, avoiding the need to extend the full model parameter space to include the nuisance parameters of the legacy experiment. The combination of CGG21 and P18 allows us test this approach in a particularly challenging scenario. As discussed in CGG21, the CGG21 data display a $\sim 3.5\sigma$ tension in the value of S_8 with respect to P18. Thus, a combination

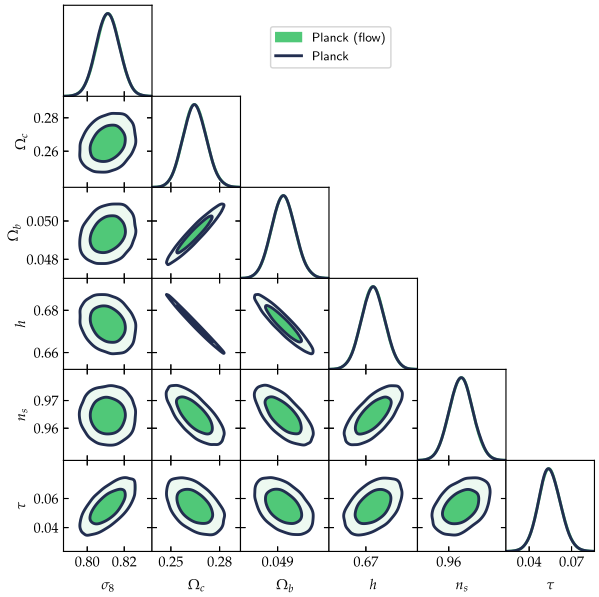
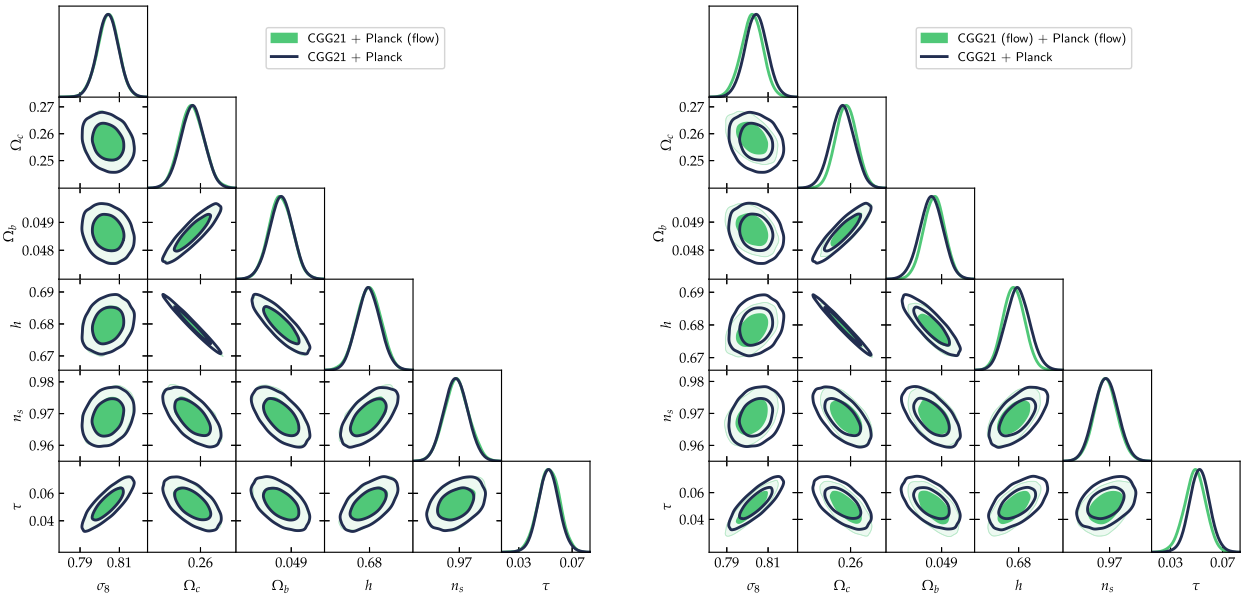


Figure 4. Figure showing the joint posterior of the cosmological parameters only (marginalized over the nuisance parameters) from the P18 data set. The green contours correspond to the samples obtained using the normalizing flow model and the black contours are the original samples.

of both data sets making use of the P18 emulated posterior is only possible if the flow is able to describe the tails of the distribution accurately. If the method is able to succeed in this case, its application to joint analyses of experiments that are in better agreement with one another would only be more reliable. The result is summarized in the left panel of Fig. 5. The figure shows, in black, the exact marginal posterior distribution on the ΛCDM cosmological parameters, found by sampling the product of the CGG21 and P18 likelihoods, including all their nuisance parameters. In turn, the constraints obtained by sampling the CGG21 likelihood using the P18 trained flow as an effective prior are shown in green. The latter approach is able to recover the exact posterior at very high accuracy. This is further quantified in Table 1, which lists the distance metrics $(\delta_\mu, \delta_\sigma, \delta_q)$. Posterior means are recovered with sub-per cent accuracy, while posterior widths are determined with a precision of 2–3 per cent. The energy distance metric $\tilde{D}$ is very close to zero, signifying an excellent agreement between both distributions.

The right panel of Fig. 5 shows the result of using the product of both trained flow models in order to obtain joint cosmological constraints. In this case we see that, although the product of emulated posteriors yields contours that are very similar to the true posterior (shown again in black), the differences between them are clearly visible. As quantified in Table 1, these correspond to shifts in the posterior means of up to $\sim 0.3\sigma$, and the energy distance metric $\tilde{D}$ grows by a factor $\sim$10 with respect with the previous case (while still staying relatively low). As discussed above, this is not entirely surprising, since the level of tension between these data sets requires both normalizing flow models to describe the distribution outskirts accurately. Given the two results presented in this work, we recommend using the emulated posterior as a prior in a likelihood analysis to achieve improved precision.

The ability to use trained flow models to describe the marginal posterior of legacy data sets leads to significant gains in computational speed for joint analyses. The exact joint posterior found by sampling the product of the P18 and CGG21 likelihoods – which



(a) CGG21 and P18 flow as prior

(b) CGG21 flow and P18 flow

Figure 5. Panel (a) shows the joint posterior of the cosmological parameters where the P18 normalizing flow is used as a prior in the analysis. Panel (b) shows the posterior in the case where the posterior is sampled using the local posterior due to the CGG21 and P18 data sets, where each density is learnt by the normalizing flow model. In both plots, the green contours correspond to the posterior due to the normalizing flow models, whereas the black contours are the known posterior as obtained by García-García et al. (2021).

includes the union of their full parameter spaces – using COBAYA takes ∼24 days using two HPC nodes until convergence is reached at the level of $R - 1 \leq 0.02$. If instead we use the P18 flow model as a prior to the CGG21 likelihood, the joint posterior is recovered after only ∼6 days, corresponding to a factor ∼4 speed-up. On the other hand, if we simply use the two flow models, joint constraints can be obtained in under ∼15 min by generating 120 000 MCMC samples using EMCEE on a desktop computer.

In addition to this study, we also perform additional tests, exploring other experiment combinations. These are described in Appendix A. In particular, we investigate the joint analysis of KiDS-1000 and DES-Y3 (Section A1) and the combination of P18 and DES-Y1 (Section A2). These cases correspond to posterior distributions exhibiting lower levels of tension than the CGG21 + P18 case described here, but displaying markedly less 'Gaussian' marginalized constraints. The qualitative results obtained above are confirmed and remain valid in these cases.

Although we have only considered pairwise combinations of experiments, normalizing flow models may be used to facilitate the combination of arbitrary numbers of independent data sets. Care should be taken, however, that no individual-data set posterior incorporates informative priors on the cosmological parameters, and that any non-informative but non-flat priors are corrected for. For instance, using Bayes' theorem:

$$p(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \propto p(\boldsymbol{x}_1|\boldsymbol{\theta}, \boldsymbol{\beta}) p(\boldsymbol{\theta}, \boldsymbol{\beta}) \prod_{i=2}^{N} \frac{p_{\rm nf}(\boldsymbol{\theta}|\boldsymbol{x}_i)}{p(\boldsymbol{\theta})}, \tag{37}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{\beta}$ are the data and nuisance parameters of the main experiment, respectively, $\boldsymbol{\theta}$ are the shared cosmological parameters, and $p(\theta)$ is the uninformative prior on θ assumed in all cases.

5 PUBLIC LIKELIHOODS

In addition to the above analyses, we have also created a 'Cosmological Zoo of Normalizing Flow Models', where we have taken public MCMC samples, processed them in such a way that we retain only the cosmological parameters, effectively marginalizing over the nuisance parameters. We then train and store the respective normalizing flow, which can then be used for two purposes as investigated in Section 4, (1) to perform joint analysis using the flow models only and (2) to use them as a prior in a cosmological analysis (see equation 2) using independent data sets. We will briefly describe the emulated chains below. The software used to generate these emulated chains is made publicly available, and users can easily expand on this set of public emulated likelihoods.

5.1 Planck 2018

A variety of MCMC chains were made publicly available by the *Planck* collaboration. For concreteness and simplicity, we use the `base_plikHM_TTTEEE_lowl_lowE` MCMC samples (Planck Collaboration VI 2020b).

As outlined in section 2 of Planck Collaboration VI (2020b), the samples were produced using a base ΛCDM cosmological model evaluated with CAMB (Lewis, Challinor & Lasenby 2000). This particular experiment focused on the TT, TE, and EE power spectra for $\ell > 30$, along with the low-ℓ likelihoods. To estimate the TT, TE, and EE power spectra, the PLIK high multipole likelihood was applied, employing a Gaussian approximation. For modelling the small-scale non-linear matter power spectrum, HMCODE (Mead et al. 2015, 2016) was used. Six cosmological parameters and numerous nuisance parameters were sampled, with derived parameters such as σ_8 and H_0 also being recorded.

5.2 DES Y3

We build a normalizing flow for the marginalized cosmological posterior of the DES Y3 '3 × 2-point analysis' using the full 5000 deg^2 of imaging data (Abbott et al. 2022). The data vector includes the two-point functions for galaxy clustering, galaxy–galaxy lensing and cosmic shear:

$$\boldsymbol{d} \equiv \{\hat{w}^i(\theta),\ \hat{\gamma}_t^{ij}(\theta),\ \hat{\zeta}_{\pm}^{ij}(\theta)\}$$

where only autocorrelations are considered for galaxy clustering. After applying the necessary scale cuts, we are left with 462 elements in the data vector. Abbott et al. (2022) employed two cosmological models, namely, ΛCDM and wCDM, which have a total of 31 and 32 parameters, respectively, including 25 nuisance parameters. We emulate the MCMC chains with fixed neutrino mass.

5.3 KiDS-1000

For KiDS-1000, we generate normalizing flow emulators for the ΛCDM chains obtained from different analysis choices.

Asgari et al. (2021) made use of different two-point statistics for modelling the cosmic shear data. In particular, correlation functions, band power spectra, and the Complete Orthogonal Sets of E-/B-Integrals (COSEBIs) were used, marginalizing over seven nuisance and astrophysical parameters. The posterior is sampled using nested sampling (Feroz et al. 2019). The forward cosmological model entails the calculation of the linear matter power spectrum with CAMB (Lewis et al. 2000) and the non-linear contribution using HMCODE (Mead et al. 2015, 2016). The different statistics yield different constraints on the cosmological parameters (see Fig. 6 from Asgari et al. 2021). We provide normalizing flow models, for each of these three types of two-point statistics.

On the other hand, Heymans et al. (2021) performed a joint cosmological analysis using KiDS-1000, BOSS (Alam et al. 2015), and the 2-degree Field Lensing Survey (2dFLenS) (Blake et al. 2016) data. The flat ΛCDM forward model has 20 parameters, 5 cosmological parameters and 15 nuisance and astrophysical parameters, and the posterior was sampled using nested sampling. The posterior distribution displays a $\sim 3\sigma$ with *Planck* in the value of S_8. The pre-trained normalizing flow based on the MCMC samples of this experiment is also made available.

Finally, Dark Energy Survey and Kilo-Degree Survey Collaboration (2023) performed a hybrid analysis using DES Y3 and KiDS-1000 cosmic shear data only. The goal was to investigate and compare different modelling strategies employed by the DES and KiDS teams separately. When the DES data only are used, the model has 17 parameters (6 cosmological and 11 nuisance and astrophysical parameters). On the other hand, when modelling the KiDS-1000 data, there are 14 parameters (6 cosmological and 8 nuisance and astrophysical parameters). We also train normalizing flow models for each individual experiment in this set-up.

5.4 ACT DR4

We also used the public MCMC chains (DR4 TT+TE+EE) for the Atacama Cosmology Telescope (ACT), to build a normalizing flow models for the joint cosmological parameters. Aiola et al. (2020) performed different analyses which include joint analyses with Wilkinson Microwave Anisotropy Probe (WMAP) and Planck separately. In this work, we use the MCMC samples generated using ACT data alone including TT, TE, and EE binned CMB bandpowers. The ΛCDM model, with six cosmological parameters, is used in the parameter estimation task and two derived parameters (H_0 and σ_8) are also recorded.

5.5 SDSS

Recently, Alam et al. (2021) conducted extensive cosmological analyses using data from the completed Sloan Digital Sky Survey (SDSS), encompassing SDSS, SDSS II, the Baryon Oscillation Spectroscopic Survey (BOSS), and the extended BOSS (eBOSS). These data sets allow for the extraction of various cosmological measurements, including baryon acoustic oscillations (BAO). Alam et al. (2021) explored several joint analyses involving cosmic shear, CMB temperature and polarization, supernovae, BAO, and other data sources to gain deeper insights of different cosmological models. We use the CMB+BAO and CMB+BAO+SN chains to train two separate normalizing flow models. Training each flow model with 20 000 MCMC samples took only about 2 min. The original analyses involved 6 cosmological parameters and 20 nuisance parameters, which were marginalized over using COBAYA.

6 SOFTWARE

The software used to generated normalizing flow emulators of marginalized cosmological posteriors is publicly available. We describe this software briefly here.

In the first step, the user processes the data in such a way that only the samples for a reduced set of cosmological parameters are retained. Specifically, we use the five ΛCDM parameters

$$\boldsymbol{\theta} = \{\sigma_8,\ \Omega_c,\ \Omega_b,\ h,\ n_s\}. \tag{38}$$

A configuration file is then created for this specific experiment, containing the experiment name, the learning rate, the number of optimization steps and the number of training points to be used for training the normalizing flow model. If we do not specify the number of training points, all the samples will be used for training. However, we recommend using around 2×10^4, if available. The model can be trained both for a single configuration or a combination of configurations (different learning rates, number of optimization steps, number of training points). After the training procedure, the code stores the trained model and plot the loss curve and the projected 1D and 2D distributions of the original samples and the samples generated by the normalizing flow model.

Once the models are trained, users may use them in joint cosmological analyses. We provide example code that describes how to use the trained normalizing flows as part of an MCMC run using EMCEE (Foreman-Mackey et al. 2013) (although nothing in the software limits its use to this particular sampler).

While many MCMC chains are available from past experiments, we strongly encourage current and future collaborations to make their MCMC chains publicly accessible. This would enable the rapid joint analysis of cosmological data sets using the method presented here. Additionally, it is feasible to extend this approach to higher dimensional posteriors, beyond the 5D or 6D cases explored in this work, by employing the normalizing flow method presented in this work. However, achieving reliable posterior reconstruction in higher dimensions may require a larger training set, potentially exceeding 2×10^4 samples. The optimal number of training points depends on both the dimensionality of the problem and the precision constraints on the parameters set by the data. The code outputs a comparison plot of the 1D and 2D marginal posterior distributions

from samples generated by the pre-trained flow model and the training set, providing a useful tool for assessing the robustness of the flow model.

7 CONCLUSION

In this work, we have explored how normalizing flow models can be used to learn the cosmological posterior distribution marginalized over nuisance parameters. Once trained and stored, these models can be used for various purposes, such as generating large numbers of samples of cosmological parameters, or calculating the log-density at any point in parameter space, which can lead to a significant simplification in the way data from independent experiments are combined.

We have performed and assessed different experiments of how the pre-trained models can be used. Using the PoE approach explained in Section 2.2, any N normalizing flow models can be combined by multiplying them together, effectively summing the log-posterior and hence enabling fast sampling of the joint posterior of N experiments. We have investigated this approach using two methods: (1) applying the flow model as a prior in a likelihood analysis, and (2) using two or more normalizing flow models to sample the joint posterior. These techniques have been thoroughly tested with a combination of large-scale structure data sets and Planck, as well as with other experiments like DES Y3 and KiDS-1000 in Appendix A.

Even in the case where we have a significant degree of tension between two sets of parameters in the joint analysis of P18 and CGG21, we have shown that we can recover the marginalized posterior distribution of the cosmological parameters, with good precision ($\delta_\sigma \lesssim 0.11$) and accuracy ($\delta_\mu \lesssim 0.07$). When the flow model is used as a prior, the joint posterior is closer (lower δ_μ, δ_σ, δ_q, and $\tilde{D}$) to the posterior of the full analysis. This is expected since we are coupling only one approximate density (compared to two or more) with the likelihood.

To test the method further, we have also used other data sets to sample the joint posterior. For example a joint analysis using the DES Y3 and KiDS-1000 data using their flow models results in a comparable posterior with the known joint posterior distribution of the cosmological parameters. If we extend this further and add the contribution due to the P18 normalizing flow model, the parameters shift in the expected directions (see Appendix A).

In addition to using our own MCMC samples for the above experiments, we have also processed, trained, and stored normalizing flow models for public MCMC chains. A few highlighted here are *Planck*, DES Y3, KiDS-1000, ACT, and SDSS. The software used to train and exploit these normalizing flows is written in a simple way and it should be straightforward for users to implement and train new models. Importantly, training new models, or extending the cosmological model under study (e.g. to include neutrino masses or dynamical dark energy), is both straightforward and computationally inexpensive, as long as sufficient training data exists in the form of MCMC samples.

SOFTWARES

The following PYTHON libraries have been used as part of this project: FLOWTORCH (Webb 2022), GETDIST (Lewis 2019), SCIPY (Virtanen et al. 2020), NUMPY (Harris et al. 2020), PANDAS (Reback et al. 2020), COBAYA (Lewis 2013), JAX-COSMO (Campagne et al. 2023), PYTORCH (Paszke et al. 2019), and HYDRA (Yadan 2019).

ACKNOWLEDGEMENTS

We thank Dr Zafiirah Hosenie for reviewing this manuscript and providing useful feedback. AM was supported through the LSST Discovery Alliance (LSST-DA) Catalyst Fellowship project; this publication was thus made possible through the support of Grant 62192 from the John Templeton Foundation to LSST-DA. DA and CGG acknowledge support from the Beecroft Trust. JRZ was supported by UK Space Agency grant ST/W001721/1. We made extensive use of computational resources at the University of Oxford Department of Physics, funded by the John Fell Oxford University Press Research Fund.

DATA AVAILABILITY

The code and part of the data products underlying this article can be found at: https://github.com/Harry45/emuflow/. The processed data and pre-trained normalizing flows are also made public in the folders `samples/` and `flows/`, respectively.

REFERENCES

Abbott T. M. C. et al., 2022, Phys. Rev. D, 105, 023520
Ade P. et al., 2019, J. Cosmol. Astropart. Phys., 2019, 056
Aiola S. et al., 2020, J. Cosmol. Astropart. Phys., 2020, 047
Alam S. et al., 2015, ApJS, 219, 12
Alam S. et al., 2021, Phys. Rev. D, 103, 083533
Alonso D., Sanchez J., Slosar A., LSST Dark Energy Science Collaboration, 2019, MNRAS, 484, 4127
Alsing J., Handley W., 2021, MNRAS, 505, L95
Alsing J., Wandelt B., Feeney S., 2018, MNRAS, 477, 2874
Amendola L. et al., 2018, Living Rev. Relativ., 21, 2
Aricò G., Angulo R. E., Contreras S., Ondaro-Mallea L., Pellejero-Ibañez M., Zennaro M., 2021, MNRAS, 506, 4070
Asgari M. et al., 2021, A&A, 645, A104
Bartlett D. J., Wandelt B. D., Zennaro M., Ferreira P. G., Desmond H., 2024a, A&A, 686, A150
Bartlett D. J. et al., 2024b, A&A, 686, A209
Bevins H., Handley W., Lemos P., Sims P., de Lera Acedo E., Fialkov A., 2022, preprint (arXiv:2207.11457)
Bevins H. T. J., Handley W. J., Lemos P., Sims P. H., de Lera Acedo E., Fialkov A., Alsing J., 2023, MNRAS, 526, 4613
Bevins H. T. J., Heimersheim S., Abril-Cabezas I., Fialkov A., de Lera Acedo E., Handley W., Singh S., Barkana R., 2024, MNRAS, 527, 813
Bińkowski M., Sutherland D. J., Arbel M., Gretton A., 2018, preprint (arXiv:1801.01401)
Blake C. et al., 2016, MNRAS, 462, 4240
Blas D., Lesgourgues J., Tram T., 2011, J. Cosmol. Astropart. Phys., 2011, 034
Campagne J.-E. et al., 2023, The Open Journal of Astrophysics, 6, 15
Chisari N. E. et al., 2019, ApJS, 242, 2
Dark Energy Survey and Kilo-Degree Survey Collaboration, 2023, The Open Journal of Astrophysics, 6, 36
DESI Collaboration, 2024, preprint (arXiv:2404.03002)
Feroz F., Hobson M. P., Cameron E., Pettitt A. N., 2019, The Open Journal of Astrophysics, 2, 10
Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, PASP, 125, 306
García-García C., Ruiz-Zapatero J., Alonso D., Bellini E., Ferreira P. G., Mueller E.-M., Nicola A., Ruiz-Lapuente P., 2021, J. Cosmol. Astropart. Phys., 2021, 030CGG21
García-García C., Zennaro M., Aricò G., Alonso D., Angulo R. E., 2024, J. Cosmol. Astropart. Phys., 2024, 024
Hadzhiyska B., Alonso D., Nicola A., Slosar A., 2020, J. Cosmol. Astropart. Phys., 2020, 056

Hang Q., Alam S., Peacock J. A., Cai Y.-C., 2021, MNRAS, 501, 1481
Harris C. R. et al., 2020, Nature, 585, 357
Heavens A., Fantaye Y., Mootoovaloo A., Eggers H., Hosenie Z., Kroon S., Sellentin E., 2017a, preprint (arXiv:1704.03472)
Heavens A., Fantaye Y., Sellentin E., Eggers H., Hosenie Z., Kroon S., Mootoovaloo A., 2017b, Phys. Rev. Lett., 119, 101301
Heymans C. et al., 2021, A&A, 646, A140
Hou J. et al., 2021, MNRAS, 500, 1201
Ivezić Ž. et al., 2019, ApJ, 873, 111
Jamieson D., Li Y., Villaescusa-Navarro F., Ho S., Spergel D. N., 2024, preprint (arXiv:2408.07699)
Kodi Ramanah D., Charnock T., Villaescusa-Navarro F., Wandelt B. D., 2020, MNRAS, 495, 4227
Leclercq F., 2018, Phys. Rev. D, 98, 063511
Lemos P. et al., 2021, MNRAS, 505, 6179
Lewis A., 2013, Phys. Rev. D, 87, 103529
Lewis A., 2019, preprint (arXiv:1910.13970)
Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473
McEwen J. D., Wallis C. G. R., Price M. A., Spurio Mancini A., 2021, preprint (arXiv:2111.12720)
Mead A. J., Peacock J. A., Heymans C., Joudaki S., Heavens A. F., 2015, MNRAS, 454, 1958
Mead A. J., Heymans C., Lombriser L., Peacock J. A., Steele O. I., Winther H. A., 2016, MNRAS, 459, 1468
Mootoovaloo A., Heavens A. F., Jaffe A. H., Leclercq F., 2020, MNRAS, 497, 2213
Mootoovaloo A., Jaffe A. H., Heavens A. F., Leclercq F., 2022, Astron. Comput., 38, 100508
Mootoovaloo A., Ruiz-Zapatero J., García-García C., Alonso D., 2024, MNRAS, 534, 1668
Paszke A. et al., 2019, preprint (arXiv:1912.01703)
Planck Collaboration V, 2020a, A&A, 641, A5
Planck Collaboration VI, 2020b, A&A, 641, A6
Planck Collaboration VIII, 2020c, A&A, 641, A8
Polanska A., Price M. A., Piras D., Spurio Mancini A., McEwen J. D., 2024, preprint (arXiv:2405.05969)
Reback J. et al., 2020, pandas-dev/pandas: Pandas 1.0.0, Zenodo, doi:10.5281/zenodo.3630805
Rizzo M. L., Székely G. J., 2016, Wiley Interdiscip. Rev. Comput. Stat., 8, 27
Rozo E. et al., 2016, MNRAS, 461, 1431
Ruiz-Zapatero J., Hadzhiyska B., Alonso D., Ferreira P. G., García-García C., Mootoovaloo A., 2023, MNRAS, 522, 5037
Spurio Mancini A., Piras D., Alsing J., Joachimi B., Hobson M. P., 2022, MNRAS, 511, 1771
Srinivasan R., Crisostomi M., Trotta R., Barausse E., Breschi M., 2024, preprint (arXiv:2404.12294)
Takahashi R., Sato M., Nishimichi T., Taruya A., Oguri M., 2012, ApJ, 761, 152
Taylor P. L., Cuceu A., To C.-H., Zaborowski E. A., 2024, The Open Journal of Astrophysics, 7, 86
Torrado J., Lewis A., 2021, J. Cosmol. Astropart. Phys., 2021, 057
Tresp V., 2000, Neural Comput., 12, 2719
Tröster T., Ferguson C., Harnois-Déraps J., McCarthy I. G., 2019, MNRAS, 487, L24
Virtanen P. et al., 2020, Nat. Methods, 17, 261
Webb S., 2022, FlowTorch, (Last accessed in July 2024), Github, https://github.com/facebookincubator/flowtorch
Wu P., Imbiriba T., Elvira V., Closas P., 2024, IEEE Trans. Signal Process., 72, 275
Yadan O., 2019, Hydra—A framework for elegantly configuring complex applications, (Last accessed in July 2024), Github, https://github.com/facebookresearch/hydra
Zuntz J. et al., 2018, MNRAS, 481, 1149

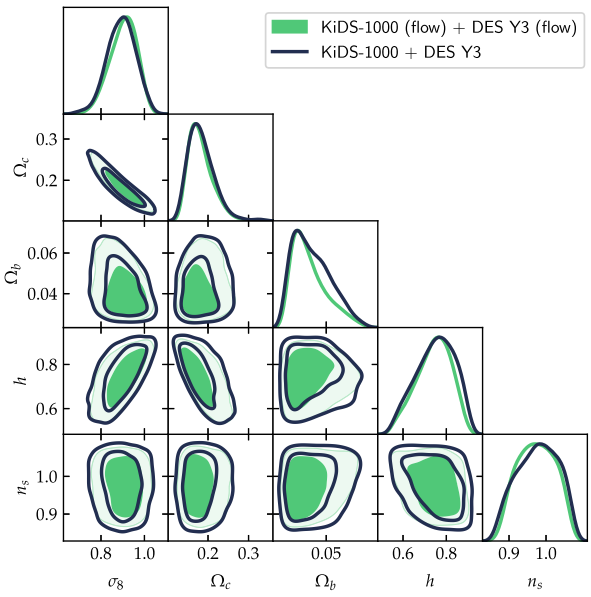


Figure A1. The marginalized posterior distribution of the cosmological parameters using KiDS-1000 and DES Y3 data. The contours in black shows the result due to the exact evaluation of the likelihoods while the contours in green correspond to the case where we jointly sample the individual normalizing flow models for DES Y3 and KiDS-1000.

APPENDIX A: ADDITIONAL TESTS

To further test the method described in this work, we perform additional tests using other data sets. First, in Section A1, we consider the KiDS-1000 and DES Y3 cosmic shear analysis, where the posteriors are broad, with strong constraints on only relatively few cosmological parameters. In Section A2, we use the normalizing flow model for the *Planck* public MCMC samples as a prior in a joint cosmological analysis and compare the results with the case where two normalizing flow models are used jointly. Using these data sets, we reach conclusions consistent with those from the main cosmological examples (CGG21 and Planck 2018) discussed in this work. This consistency highlights the robustness and reliability of the method presented in this paper.

A1 KiDS-1000 and DES Y3

To test the performance of the pre-trained normalizing flows, we also look into inferring cosmological parameters only from large-scale structure data sets, which currently yields broad posteriors and a non-Gaussian joint posterior in the σ_8 and Ω_m plane. García-García et al. (2024) carried out a joint analysis of the KiDS-1000, DES Y3, and HSC-DR1 cosmic shear data sets. To model the non-linear matter power spectrum, García-García et al. (2024) use the BACCOEMU emulator whilst taking advantage of the baryonification procedure implemented in the algorithm. The forward model for either data set consists of different nuisance parameters (shifts in the redshift distributions and multiplicative biases).

We build two normalizing flows based on the cosmological samples, $\boldsymbol{\theta}$ and we compute the joint posterior using EMCEE. This posterior is compared to the MCMC samples obtained from the full run, where the nuisance parameters in the forward modelling both data sets are marginalized over. Results are shown in Fig. A1. The two flows when sampled together, are able to capture the non-Gaussian

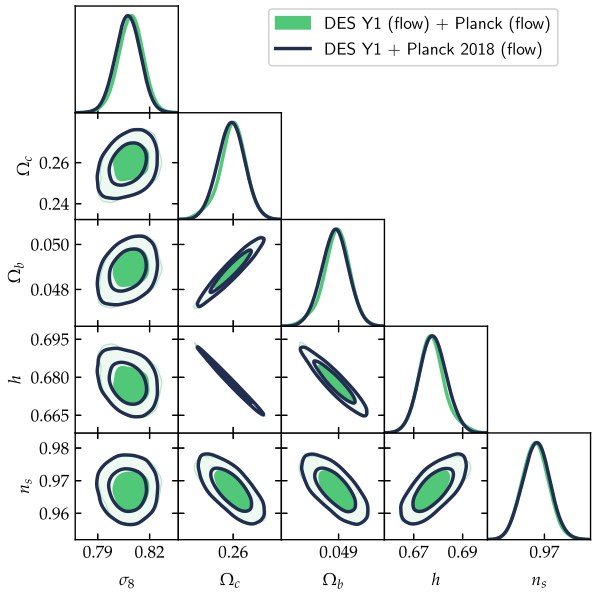


Figure A2. The marginalized posterior distribution of the cosmological parameters in the joint analysis of the DES Y1 and Planck 2018. For the latter, we use the public MCMC chains for Planck 2018 to train a normalizing flow model, which is then used as a prior in conjunction with the DES Y1 likelihood. This result is shown in black. In a separate analysis, we also simply use two normalizing flows (DES Y1 and Planck 2018) to sample the joint posterior, which is shown in green. There are mild differences between the two distributions (black and green), thus demonstrating the robustness of the method.

banana-shape posterior in the σ_8 and Ω_c plane and there are only very mild differences in the posterior distributions. This difference might also arise due to the overlapping area of the DES Y3 and KiDS-1000 surveys, which is not fully accounted for in the original joint analysis.

A2 DES Y1 and Planck

Similar to the analysis done in Section 4, in this section we would like to investigate if we can use a pre-trained model as part of a cosmological analysis, that is, we want to simulate a scenario in which we aim to explore the constraints obtained with a new experiment in combination with a previous independent data set (for which we have built a normalizing flow model). As an example, we will use band-powers data for DES Y1 galaxy clustering and cosmic shear data set, as well as the *Planck* public constraints. The forward model is described in Mootoovaloo et al. (2024). It has five cosmological parameters, similar to the ones used for building the normalizing flow model discussed in the main text. The forward model also has 20 nuisance parameters, which we would like to marginalize over. We will also use the pre-trained normalizing flow for the Planck 2018 (`base_plikHM_TTTEEE_lowl_lowE`). As discussed in Section 2.1, we only have to marginalize over the nuisance parameters, $\boldsymbol{\beta}$, for DES Y1, where we have introduced the approximate posterior built using the normalizing flow model for Planck 2018. Note that the normalizing flow model should be weighted by the prior in the analysis.

To sample the joint posterior, we have used the Metropolis–Hastings sampler implemented in COBAYA (Lewis 2013). We have set the specifications as follows: the number of samples is 5×10^5 and the Gelman–Rubin convergence criterion is $R - 1 = 0.01$. The sampler will stop once either of these criteria is met. A total of 215×10^3 MCMC samples were generated and the Gelman–Rubin convergence criterion was met. Sampling takes around 5 hours in this set-up, roughly similar to what it would take if we were to sample the cosmological and nuisance parameters in DES Y1.

On the other hand, we also use each individual flow (DES Y1 flow and Planck flow) to find the joint posterior distribution of the cosmological parameters (see Fig. A2). Sampling the joint in this case is quick and takes ∼15 min on a desktop computer.

This paper has been typeset from a TEX/LATEX file prepared by the author.