Constraining the Reionization History using Bayesian Normalizing Flows

IOP Publishing Mach. Learn.: Sci. Technol. 1 (2020) 035014 https://doi.org/10.1088/2632-2153/aba6f1

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
16 May 2020

REVISED
30 June 2020

ACCEPTED FOR PUBLICATION
17 July 2020

PUBLISHED
21 August 2020

PAPER

Constraining the Reionization History using Bayesian Normalizing Flows

Héctor J. Hortúa, Luigi Malagò and Riccardo Volpi

Machine Learning and Optimization Group, Romanian Institute of Science and Technology (RIST), Cluj-Napoca, Romania

E-mail: hortua.orjuela@rist.ro, malago@rist.ro **and** volpi@rist.ro

Keywords: Deep Learning, Reionization, Intergalactic Medium, Cosmology

Abstract

Upcoming experiments such as Hydrogen Epoch of Reionization Array(HERA) and the Square Kilometre Array (SKA) are intended to measure the 21 cm signal over a wide range of redshifts, representing an incredible opportunity in advancing our understanding about the nature of cosmic reionization. At the same time these kind of experiments will present new challenges in processing the extensive amount of data generated, calling for the development of automated methods capable of precisely estimating physical parameters and their uncertainties. In this deliverable we employ Variational Inference, and in particular Bayesian Neural Networks, as an alternative to MCMC in 21 cm observations to report credible estimations for cosmological and astrophysical parameters and assess the correlations among them. Finally, we have implemented the use of bijectors to improve the diagonal Gaussian approximate posteriors and be able to extract significant information from Non-Gaussian signal in the 21 cm dataset.

1. Introduction

In the past decade, Cosmology has entered into a new precision era due to the considerable number of experiments performed to obtain information both from early stages of the Universe through the Cosmic Microwave Background (CMB) and late times via deep redshift surveys of large-scale structures. These measurements have yielded precise estimates for the parameters in the standard cosmological model, establishing the current understanding of the Universe. However, the intermediate time known as Epoch of Reionization (EoR), when the first stars and galaxies ionized the InterGalactic Medium (IGM), remains vastly unexplored. This period is relevant to understand the properties of the first structures of our Universe and provide complementary information related to fundamental Cosmology, inflationary models, and neutrino constraints, among others, e.g. [1]. EoR observations combined with CMB can improve existing constraints on the cosmological parameters for particular low-redshift reionization scenarios. For instance, McQuinn *et al* [2] reported uncertainties 1.4 or 3 times smaller than those constrained by CMB alone, and these results can be further improved by including Weak Lensing in the analysis [3]. Furthermore, authors in [4] have reported that EoR observations are effective at breaking the CMB degeneration between the optical depth and the amplitude of the primordial fluctuation spectrum reducing significantly the errors on $\ln(10^{10}A_s)$. It is known that the EoR can be studied indirectly through its imprint in the IGM, using the redshifted 21 cm line [5]. This line results from the hyperfine splitting of the ground state of the hydrogen atom due to the coupled magnetic moments between the proton and the electron, emitting radiation with a 21 cm wavelength, then redshifted by the expansion of the Universe [1]. Future experiments such as Hydrogen Epoch of Reionization Array (HERA)[1] and the Square Kilometre Array (SKA)[2] are intended to measure this 21 cm signal over a wide range of redshifts providing 3D maps of the first hundreds millions years of the Universe. These instruments are expected to generate a huge amount of spectra, encouraging the development of automated methods capable of reliably estimating physical parameters with great accuracy.

[1]https://reionization.org/.
[2]https://www.skatelescope.org/.

Recently, Deep Neural Networks (DNNs) have been applied in several fields of Astronomy because of their ability to extract complex information from data, which makes it benefit for analysing non-Gaussian signatures. In particular, the application of DNNs to the 21 cm signal has received considerable attention due to the success of classifying reionization models [6] or estimating physical parameters [7]. For example, in [7] 2D images corresponding to slices along the line-of-sight axis of the light-cones were used for training Convolutional Neural Networks (CNN) in order to estimate some astrophysical parameters. More recently [8] and [9] generalized the previous findings by incorporating contamination from simulated SKA-like noise. However, DNNs are prone to overfitting due to the high number of parameters to be adjusted, and do not provide a measure of the uncertainty for the estimated parameters, see for instance [10–12]. These limitations can be addressed by following a Bayesian approach, both intrinsically providing an effective regularization during training and allowing to quantify the uncertainty in the predicted parameters at inference time [13].

In this paper, we generalize these preliminary works related to the application of DNNs on the 21 cm data by implementing Bayesian Neural Network (BNNs), in order to obtain the posterior probability estimates of the physical parameters and their correlations. We discuss methods for calibrating uncertainties in Bayesian Networks and we propose approaches toward the efficient extraction of information in non-Gaussian signals via diffeomorphic transformations of the output distribution.

This work is organized as follows. In section 2 we introduce the variational inference formalism which allows to compute the aleatoric and epistemic uncertainty of BNNs, and we also comment on the use of bijectors for improving inference tasks. In section 3 we describe the Reionization model used in this work and the generation of the synthetic dataset. In section 4 we describe the network architecture, and in section 5 we show the results related to the potential application of BNNs to obtain approximate posteriors over the parameter space. Moreover, we also present the most relevant findings about implementing Normalizing Flows in inference task and their advantages in estimating the parameters relevant for the EoR. Finally, conclusions and future works are shown in section 6.

2. Variational Inference

BNNs provide the adequate groundwork to output reliable estimations for many machine learning tasks. Let us consider a training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{D}$ formed by D couples of images $\boldsymbol{x}_i \in \mathbb{R}^M$ and their respective targets $\boldsymbol{y}_i \in \mathbb{R}^N$. By setting a prior distribution $p(\boldsymbol{w})$ on the model parameters $\boldsymbol{w}$, the posterior distribution can be obtained from Bayes' law as $p(\boldsymbol{w}|\mathcal{D}) \sim p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$. Unfortunately, the posterior usually cannot be obtained analytically and thus approximate methods are commonly used to perform the inference task. The Variational Inference (VI) approach approximates the exact posterior $p(\boldsymbol{w}|\mathcal{D})$ by a parametric distribution $q(\boldsymbol{w}|\theta)$ depending on a set of variational parameters $\boldsymbol{\theta}$ [10]. These parameters are adjusted to minimize a certain loss function, usually given by the KullBack-Leibler divergence $\mathrm{KL}(q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w}|\mathcal{D}))$. It has been shown that minimizing the KL divergence is equivalent to minimizing the following objective function [10]

$$\mathcal{F}_{VI}(\mathcal{D}, \boldsymbol{\theta}) = \mathrm{KL}(q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w})) - \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \int_{\Omega} q(\boldsymbol{w}|\boldsymbol{\theta}) \ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) d\boldsymbol{w} \,. \tag{1}$$

To infer the correlations between the parameters uncertainties [11, 14, 15], we need to predict the full covariance matrix. This requires to produce in output of the last layer of the network a mean vector $\boldsymbol{\mu} \in \mathbb{R}^N$ and a covariance matrix $\Sigma \in \mathbb{R}^{N\times N}$ representing the aleatoric uncertainty, for instance parameterized through its Cholesky decomposition $\Sigma = LL^\top$. These outputs determine the Negative Log-Likelihood (NLL) when $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})$ is a Multivariate Gaussian distribution [12, 14, 16]

$$\ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) \sim \frac{1}{2}\log|\Sigma| + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu}) \,. \tag{2}$$

Let $\hat{\boldsymbol{\theta}}$ be the value of $\boldsymbol{\theta}$ after training, corresponding to a minimum of $\mathcal{F}_{VI}(\mathcal{D}, \boldsymbol{\theta})$. The approximate predictive distribution $q_{\hat{\boldsymbol{\theta}}}$ of $\boldsymbol{y}^*$ for a new input $\boldsymbol{x}^*$ can be rewritten as [15]

$$q_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}^*|\boldsymbol{x}^*) = \int_{\Omega} p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{w}) q(\boldsymbol{w}|\hat{\boldsymbol{\theta}}) d\boldsymbol{w} \,. \tag{3}$$

Moreover [17] proposed an unbiased Monte-Carlo estimator for equation (3)

$$q_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}^*|\boldsymbol{x}^*) \approx \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{y}^*|\boldsymbol{x}^*, \hat{\boldsymbol{w}}_k), \quad \text{with } \hat{\boldsymbol{w}}_k \sim q(\boldsymbol{w}|\hat{\boldsymbol{\theta}}) \,, \tag{4}$$

where K is the number of samples. In Bayesian deep learning [18] two main uncertainties are of interest: the aleatoric, capturing the inherent noise in the input data, and the epistemic, capturing the uncertainty in the model, typically due to the lack of data points during training which are similar to the current observation. To obtain both uncertainties [11], we can invoke the total covariance law for a fixed $\boldsymbol{x}^*$

$$\mathrm{Cov}_{q_{\hat{\theta}}}(\boldsymbol{y}^*, \boldsymbol{y}^* | \boldsymbol{x}^*) = \mathbb{E}_{q_{\hat{\theta}}}[\boldsymbol{y}^* \boldsymbol{y}^{*\top} | \boldsymbol{x}^*] - \mathbb{E}_{q_{\hat{\theta}}}[\boldsymbol{y}^* | \boldsymbol{x}^*] \mathbb{E}_{q_{\hat{\theta}}}[\boldsymbol{y}^* | \boldsymbol{x}^*]^\top, \tag{5}$$

the images are forward passed through the network T times, obtaining a set of mean vectors $\boldsymbol{\mu}_t$ and a covariance matrices Σ_t. Then, an estimator for the total covariance of the trained model can be written as

$$\mathrm{Cov}_{q_{\hat{\theta}}}(\boldsymbol{y}^*, \boldsymbol{y}^* | \boldsymbol{x}^*) \approx \underbrace{\frac{1}{T} \sum_{t=1}^{T} \Sigma_t}_{\text{Aleatoric}} + \underbrace{\frac{1}{T} \sum_{t=1}^{T} (\boldsymbol{\mu}_t - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu}_t - \overline{\boldsymbol{\mu}})^\top}_{\text{Epistemic}}, \tag{6}$$

with $\overline{\boldsymbol{\mu}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\mu}_t$. In this setting, BNNs can be used to learn the correlations between the targets and to produce estimates of their uncertainties.

2.1. Parametrizing the approximate posterior

As explained in this section, the learning of a BNN is targeting the learning of an approximate posterior distribution over the weights. A simple way is to define $q(\boldsymbol{w}|\boldsymbol{\theta})$ through dropout, by dropping neurons from a network layer with probability dr (commonly known as Dropout rate). The authors of [17] have shown a connection between the Dropout technique and VI for Gaussian processes, allowing the neural network to be interpreted as an approximate Bayesian model. The other approach used in this work is called Flipout [19], which decompose the distribution into a mean plus a perturbation and assumes that this perturbations over different weights are independent; and the perturbation distribution is symmetric around zero, allowing to decorrelate the gradients within a mini-batch sampling with a pseudo-random noise flipping matrix, achieving lower variance for BNNs. In particular, in the present paper we will consider the case in which the distribution over the weights used by Flipout for sampling is a Gaussian.

2.2. Normalizing Flows

Transforming probability distributions has become a powerful tool in deep learning. The main idea of a Normalizing Flow is to use a diffeomorphism (a differentiable and bijective mapping) to transform the sample space of a distribution. It can be demonstrated that this is equivalent to transforming the probability distribution and thus allowing for a more expressive output of the model [20]. For a more comprehensive introduction about flows we remind the reader to [21, 22].

2.2.1. Basics concepts of Normalizing Flows

Let us consider $\boldsymbol{u} \in U \subset \mathbb{R}^D$ and $\boldsymbol{x} \in X \subset \mathbb{R}^D$ two D dimensional vectors in some subsets of $\mathbb{R}^D$. We can define probability distributions for $\boldsymbol{u}$ and $\boldsymbol{x}$, in the associated sample spaces U and X, respectively. Let f be a diffeomorphism between the two sample spaces $f: U \to X$. Knowing the probability distributions q_u, we can define q_x as

$$q_x(\boldsymbol{x}) = q_u(\boldsymbol{u}) \left| \det \frac{\partial f(\boldsymbol{u})}{\partial \boldsymbol{u}} \right|^{-1}, \tag{7}$$

where $\boldsymbol{x} = f(\boldsymbol{u})$. We can construct more complex densities by applying successively the bijector (7), thus transforming an initial random variable $\boldsymbol{u}$ with distribution q_u $(= q_0)$ through a series of transformations f_n as

$$\boldsymbol{x}_n = f_n \circ \ldots \circ f_2 \circ f_1(\boldsymbol{u}) \tag{8}$$

$$\ln q_n(\boldsymbol{x}_n) = \ln q_u(\boldsymbol{u}) - \sum_{i=1}^{n} \ln \left| \det \frac{\partial f_i(\boldsymbol{x}_{i-1})}{\partial \boldsymbol{x}_{i-1}} \right|, \tag{9}$$

where we defined $\boldsymbol{x}_0 = \boldsymbol{u}$ for convenience of notation. Any expectation $\mathbb{E}_{q_n}[h(\boldsymbol{x}_n)]$ can be written as an expectation under q_u as

$$\mathbb{E}_{q_n}[h(\boldsymbol{x}_n)] = \mathbb{E}_{q_u}[h(f_n \circ f_{n-1} \circ \ldots \circ f_1(\boldsymbol{u})] \,. \tag{10}$$

Through a suitable choice of the diffeomorphisms f_n, we can start from simple factorized distributions such as a mean-field Gaussian and apply normalizing flows to obtain complex and multi-modal distributions [21–23].

Ideally a flow is both expressive and requires a reduced additional cost. Several transformations have been proposed in the literature [21, 22] to compute efficiently both (8) and (9). An example are Masked Autoregressive Flows [24] (MAF) and its inverse, the Inverse Autoregressive Flows [25] (IAF) which allow to compute efficiently equation (9) and equation (8), respectively. Real valued non-volume preserving transformations [26] (NVP) are a special case of both MAF and IAF, in which d pass-through units are selected and the transformations on the other units are function of these pass-through units. A more recent improvement is represented by neural ODE [27, 28] in which the flow is represented by a continuous transformation specified through an ordinary differential equation. In this paper we will use: NVP, MAF and IAF as a proof of concept to show how to provide a more flexible and scalable distribution in the output of the network with the purpose of extracting complex features in the data such as a non-Gaussianities and provide well calibrated BNN model. When training (or calibrating) with a flow, we will use equation (9) to compute the log likelihood $\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$ in the Variational Inference objective 1.

3. Dataset generation

We generated 21 cm simulations through the semi-numerical code 21cmFast [29], producing realizations of halo distributions and ionization maps at high redshifts. Through approximate methods, the code generates full 3D realizations of the density, ionization, velocity, spin temperature, and 21-cm brightness temperature fields. The latter is computed as [30]

$$\delta T_b \approx 27(1+\delta_m)x_{HI}\Big(\frac{T_S - T_\gamma(z)}{T_S}\Big)\Big(\frac{\Omega_b h^2}{0.023}\Big)\Big(\frac{1+z}{10}\frac{0.15}{\Omega_m h^2}\Big)^{\frac{1}{2}}\ \text{mK}, \tag{11}$$

where T_S and $T_\gamma(z)$ are the gas spin and the CMB temperatures at redshift z, respectively, δ_m is the density contrast of baryons, and x_{HI} denotes the neutral fraction of hydrogen. At $z \approx 6-20$, the emission of photons from the first galaxies and black holes ionizes and warm up the IGM [31]. Once the gas is ionized at some percent level ($\sim 25\%$, see [32]), the spin temperature becomes greater than CMB temperature, $T_S >> T_\gamma$ (saturation assumption), and thus, the dependence on T_S may be neglected in equation (11). Combining the excursion-set formalism and perturbation theory [29], 21cmFast computes the contrast density and the HI ionized field parametrized by x_{HI}, relevant to derive the 21 cm brightness temperature, equation (11). In order to generate the synthetic dataset for training the network, we follow the ideas started in [13] where we have varied four parameters. Two parameters corresponding to the cosmological context: the matter density parameter $\Omega_m \in [0.2, 0.4]$, and the amplitude of mass fluctuations on 8h^{-1}Mpc, $\sigma_8 \in [0.6, 0.8]$. The latter parameter is related to the number of collapsed dark matter halos affecting the timing of reionization. The remaining cosmological parameter is fixed at $\Omega_b h^2 = 0.022$ with $h = 0.68$. The other two parameters corresponding to the astrophysical context: the ionizing efficiency of high-z galaxies $\zeta \in [10, 100]$ and the minimum virial temperature of star-forming haloes $T_{vir}^F \in [3.98, 39.80] \times 10^4$K (hereafter represented in log10 units) that imposes a threshold which suppresses the star formation. These astrophysical variables parameterize the reionization and primarily control the timing of the EoR. For each set of parameters we produce 20 images at different redshifts in the range $z \in [6, 12]$ allowing to work under the saturated regime ($T_S >> T_\gamma$), and we stack these redshift-images into a single multi-channel tensor. This scheme brings two main advantages, first the network can extract effectively the information encoded over images as it was reported in [14], and secondly, it represents adequately the signals for the next-generation interferometers and provides advantages when we need to include effects of foreground contamination [8]. As a final result we have obtained 6, 000 images with with size (128, 128, 20) and resolution of 1.5 Mpc generated from simulations with box size of 192Mpc, and number of cells N = 128. We used a 70-10-20 split for training, validation and test, respectively.

4. Architecture and network training

All the networks are implemented using TensorFlow [3] and TensorFlow-Probability [4]. We used a modified version of the VGG architecture with 5 VGG blocks [33] (each made by two Conv2D layers and one max pooling) and channels size [32, 32, 32, 32, 64]. Kernel size is fixed to 3×3 and activation function used is

[3] https://www.tensorflow.org/.

[4] https://www.tensorflow.org/probability.

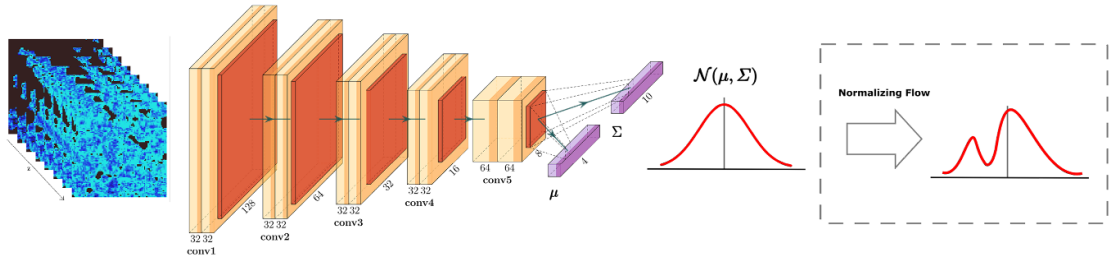


Figure 1. Scheme of the architecture used in the paper. Gray box represents the NF applied in the output of the network.

LeakyReLU ($\alpha = -0.3$). Each convolutional layer in the network is followed by a batch renormalization layer [34]. The last layer is dense with output corresponding to the mean of predictions μ and a lower triangular matrix L, cf [12, 14, 16], yielding a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\Sigma = LL^{\top}$ to guarantee positive definiteness. The network architecture is illustrated in figure 1, we train the network both with and without the Normalizing Flows. We trained the networks for 180 epochs with batches of 32 samples, using 10 samples from the approximate posterior for the estimation of equation (4) (experiments with 1 sample are also reported in the appendix for comparison). After training, to obtain the prediction distributions and the related uncertainties, we feed each input image from the test set 2,500 times to each network.

4.1. Calibration in BNNs

Predicting reliable uncertainties is crucial for classification and regression models in many applications. However, it is known that DNNs trained with NLL may produce poor uncertainty estimates [35, 36]. Weight decay, Batch Normalization and the choice of specific divergences in VI have been shown to be important factors influencing the calibration [35, 37]. One way to observe this miscalibration is computing the coverage probability in the test set. For doing this, we have binned the samples drawn from inference and computed their mode [38]. With this value, and assuming an unimodal posterior, we estimated the intervals that include the 68, 95, and 99% of the samples.

To deal with the miscalibration of the network, we will proceed as follows. First, we try to calibrate the network during training by tuning hyper-parameters such as dropout rate in Dropout [14, 38] or the regularization for the scale of the variational distribution in Flipout [14]. Then we calibrate the network using NF (figure 1), following two possible paths. On one side we can retrain the best model found so far by including NF in the output of the network to minimize equation (1). On the other hand, we can use the approach proposed in [14] to calibrate the network with a post-processing calibration approach, by fine-tuning the last layer of the network and minimizing again the NLL defined in equation (1) transformed by NF (see figure 1). At the end, we will compare the resulting networks and analyze the pros and cons for the different cases.

5. Results

We quantify the performance of the networks by the coefficient of determination and the accuracy of the uncertainties (well calibrated networks). The coefficient of determination is defined as

$$R^2 = 1 - \frac{\sum_i (\bar{\boldsymbol{\mu}}(\boldsymbol{x}_i) - \boldsymbol{y}_i)^2}{\sum_i (\boldsymbol{y}_i - \bar{\boldsymbol{y}})^2} \tag{12}$$

where $\bar{\boldsymbol{\mu}}(\boldsymbol{x}_i)$ (see equation (6)) is the prediction of the trained Bayesian network , $\bar{\boldsymbol{y}}$ is the average of the true parameters and the summations are performed over the entire test set. R^2 ranges from 0 to 1, where 1 represents perfect inference. Regarding uncertainties accuracy, for calibrated networks $\boldsymbol{y}_i$ should fall in a β% confidence interval of the conditional density estimation (4) approximately β% of the time, where $\beta = \{68.3, 95.5, 99.7\}$ corresponding to 1, 2, and 3σ confidence levels of a normal distribution.

5.1. Comparison among BNNs methods

For Dropout we tested several dropout rates in the range [0.01, 0.1] keeping L2 regularization fixed to $1e^{-5}$, while for Flipout we tested several L2 regularizations in the range [$1e^{-5}$, $1e^{-7}$]. A detailed summary of the experiments is reported in figure A1 and tables A1-A2 in the appendix.

Table 1. Metrics for the best experiments with Flipout and Dropout.

	Flipout (NLL = -2.94)				Dropout (NLL = -0.74)			
	σ_8	Ω_m	ζ	T^F_{vir}	σ_8	Ω_m	ζ	T^F_{vir}
R^2	0.94	0.98	0.87	0.97	0.87	0.94	0.65	0.92
C.L. 68.3%	69.6	73.6	72.8	76.1	70.4	67.3	58.5	76.1
C.L. 95.5%	96.0	97.1	97.4	96.7	95.7	96.3	91.7	98.5
C.L. 99.7%	99.6	99.9	99.7	99.7	99.6	99.8	99.8	99.9

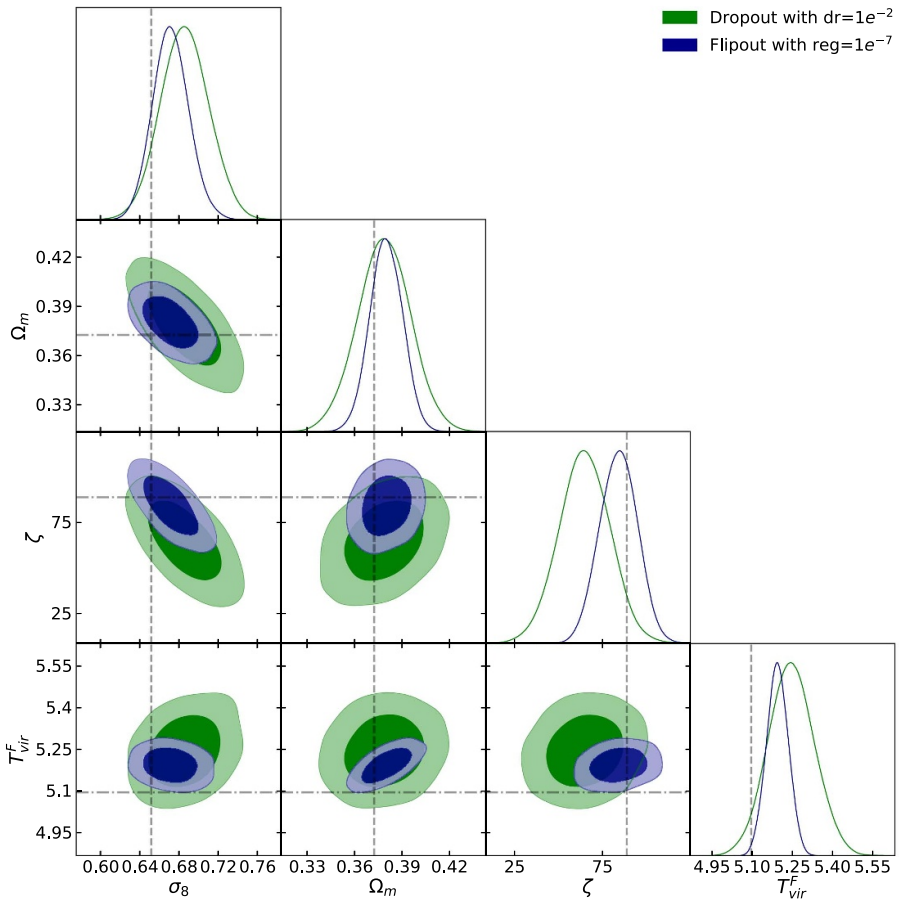


Figure 2. Posterior distributions of the parameters for one example in the test set. The dashed lines stand for the real values. The contour regions in the two-dimensional posteriors stand for 68 and 95% confidence levels.

Table 2. Limits at the 95% confidence level of the credible interval of predicted parameters.

	σ_8	Ω_m	ζ	T^F_{vir}
Flipout	$0.672^{+0.037}_{-0.036}$	$0.380^{+0.020}_{-0.020}$	$84.00^{+20.00}_{-20.00}$	$5.193^{+0.080}_{-0.079}$
Dropout	$0.686^{+0.048}_{-0.048}$	$0.379^{+0.033}_{-0.033}$	$65.00^{+30.00}_{-30.00}$	$5.250^{+0.170}_{-0.170}$
Example true value	0.652	0.372	88.847	5.096

Table 1 reports the best configuration of the network: Flipout with L2 regularizer $1e^{-7}$ and Dropout with dropout rate $1e^{-2}$. We report in the same table the coefficient of determination and average confidence intervals for Flipout and Dropout after calibration [14], and we observe that Flipout obtains the best estimations even though tends to overestimate its uncertainties.

In figure 2 we report the confidence intervals[5] for a single example in the test set and in table 2 we present the parameters predictions at the 95% confidence level. Notice that Flipout yields more accurate inferences and provides tighter constraints contours, see for example T^F_{vir}-ζ. Moreover, the correlations extracted from

[5]We use the getdist [39] package to produce the plots.

Table 3. Metrics for the best experiments with Normalizing Flows.

	IAF (NLL = -3.63)				MAF (NLL = -3.19)				NVP (NLL = -2.00)			
	σ_8	Ω_m	ζ	T^F_{vir}	σ_8	Ω_m	ζ	T^F_{vir}	σ_8	Ω_m	ζ	T^F_{vir}
R^2	0.93	0.97	0.86	0.97	0.93	0.98	0.86	0.97	0.93	0.96	0.86	0.97
C.L. 68.3%	65.6	69.8	64.0	66.8	65.6	66.0	60.3	67.7	56.6	71.1	67.1	68.9
C.L. 95.5%	94.0	95.2	93.0	94.0	95.2	94.0	90.0	94.3	86.1	96.4	94.6	95.2
C.L. 99.7%	99.1	99.7	98.7	99.3	99.4	99.2	97.8	99.3	96.4	99.6	99.5	99.4

Table 4. Limits at the 95% confidence level of the credible interval of predicted parameters using Normalizing Flows.

	σ_8	Ω_m	ζ	T^F_{vir}
IAF	$0.670^{+0.037}_{-0.033}$	$0.375^{+0.021}_{-0.021}$	$82.00^{+20.00}_{-10.00}$	$5.142^{+0.072}_{-0.070}$
MAF	$0.667^{+0.031}_{-0.029}$	$0.382^{+0.015}_{-0.016}$	$84.00^{+10.00}_{-10.00}$	$5.179^{+0.066}_{-0.068}$
Example true value	0.652	0.372	88.847	5.096

Table 5. Metrics for the best experiments with Normalizing Flows after calibration.

	IAF (NLL = -3.80)				MAF (NLL = -3.73)				NVP (NLL = -3.44)			
	σ_8	Ω_m	ζ	T^F_{vir}	σ_8	Ω_m	ζ	T^F_{vir}	σ_8	Ω_m	ζ	T^F_{vir}
R^2	0.94	0.98	0.87	0.98	0.94	0.98	0.87	0.98	0.94	0.98	0.87	0.98
C.L. 68.3%	66.0	64.0	69.2	65.4	64.7	63.7	69.1	65.0	65.9	64.8	68.8	66.0
C.L. 95.5%	94.0	94.0	95.0	94.0	93.3	94.2	95.1	94.0	93.0	94.0	94.0	93.0
C.L. 99.7%	99.2	99.2	99.5	99.6	99.0	99.3	99.3	99.4	99.0	99.2	99.0	99.0

EoR such as σ_8-Ω_m (see figure 2) provide significant information for breaking parameter degeneracies and thus, be able to improve the existing measurements on cosmological parameters [1, 2, 40]. In the following sections we will focus on the methods used for producing reliable uncertainties. Since Flipout achieves better performances than Dropout, from now on we will use Flipout to determine the performance of the subsequent calibration experiments.

5.2. Normalizing Flows during Training

A good predictive distribution depends on how well the parametric distribution matches the exact posterior. For the case presented above, the variational distribution provides a simple Gaussian approximation for the conditional density $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})$.

Normalizing Flows map an initial probability distribution through a series of transformations to produce a richer, and even a multi-modal distribution [20].

We consider different kinds of Normalizing Flows acting on the output distribution of a BNN: the inverse autoregressive flow (IAF), Masked Autoregressive Flow (MAF) and non-volume preserving flows (NVP). The results of these experiments are reported in table 3. We observe that the R^2 are comparable for all methods, but the NLL is higher for the IAF, which means that this method tends to recover better accuracy in the uncertainties. This is consistent with the findings showed in the last three rows in table 3, where the coverage probability in the test set is closer to the confidence intervals for IAF.

Normalizing Flows lead to more expressive output distributions that focus on obtaining better calibrated probabilities rather than enhancing the precision in the target values (quantified by R^2), cf with results in table 1. In figure 3 we compare the best models i.e. IAF and MAF. We can notice how smaller are the confidence regions predicted by MAF with respect to IAF, although both methods predict roughly the same orientations as expected. Finally, the values of the parameters with confidence levels at 95% are reported in table 4. Comparing these results with table 2 we can notice how the predicted parameters uncertainties are both tighter and showing some degree of asymmetry, the average skewness and kurtosis are 0.1, 0.04 for IAF, and 0.07, −0.02 for MAF.

5.3. Normalizing Flows in the post-process calibration

In this part we will focus on post-process methods for network calibration using Flows. We apply Normalizing Flows on the output distribution of the Flipout experiment reported in section 5.1, trained with a vanilla Multivariate Gaussian in output. We retrain the last layer as it was suggested in [14], plus we train the parameters of the flow. This method has the advantage that does not require to retrain the entire network, thus demonstrating to be very cost efficient.

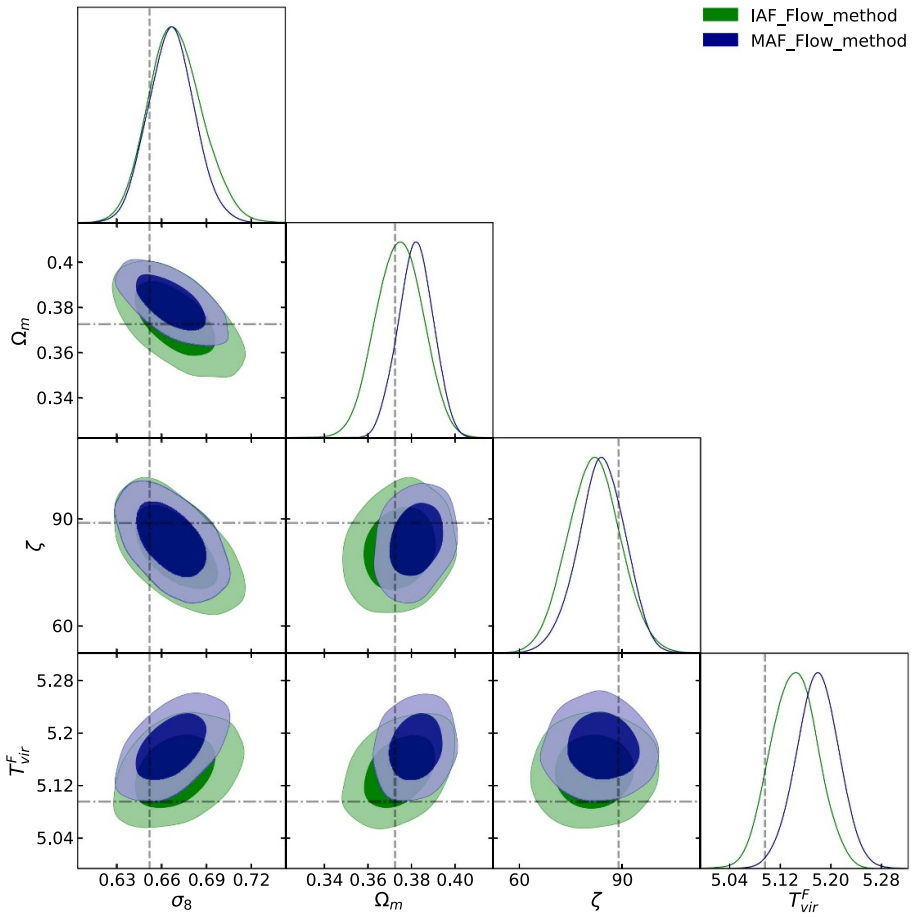


Figure 3. Posterior distributions of the parameters for one example in the test set using Normalizing flows. The dashed lines stand for the real values. The contour regions in the two-dimensional posteriors stand for 68 and 95% confidence levels.

Table 6. Limits at the 95% confidence level of the credible interval of predicted parameters using Normalizing Flows with and without calibration.

	σ_8	Ω_m	ζ	T^F_{vir}
IAF	$0.667^{+0.025}_{-0.026}$	$0.288^{+0.014}_{-0.013}$	$58.00^{+10.00}_{-10.00}$	$4.624^{+0.054}_{-0.053}$
MAF	$0.656^{+0.025}_{-0.024}$	$0.295^{+0.012}_{-0.011}$	$60.00^{+10.00}_{-10.00}$	$4.638^{+0.046}_{-0.047}$
NVP	$0.656^{+0.032}_{-0.030}$	$0.302^{+0.013}_{-0.013}$	$55.00^{+10.00}_{-10.00}$	$4.655^{+0.058}_{-0.065}$
IAF calibrated	$0.659^{+0.026}_{-0.024}$	$0.293^{+0.012}_{-0.013}$	$58.00^{+10.00}_{-10.00}$	$4.629^{+0.053}_{-0.052}$
MAF calibrated	$0.660^{+0.025}_{-0.024}$	$0.292^{+0.013}_{-0.014}$	$58.00^{+10.00}_{-10.00}$	$4.629^{+0.049}_{-0.051}$
NVP calibrated	$0.663^{+0.026}_{-0.024}$	$0.291^{+0.013}_{-0.013}$	$57.00^{+10.00}_{-10.00}$	$4.634^{+0.045}_{-0.045}$
Non-Flow	$0.662^{+0.031}_{-0.029}$	$0.294^{+0.015}_{-0.015}$	$57.00^{+12.00}_{-12.00}$	$4.644^{+0.069}_{-0.068}$
Example true value	0.664	0.285	60.750	4.629

Results after 120 epochs of the proposed recalibration are reported in table 5. Notably, the post-processing calibration outperforms all other experiments done so far, in terms of both the expected deviation from the target value, R^2, and the NLL, cf tables 1 and 3. Additionally, the improvement of the NLL leads to better calibrated networks, validated thought the coverage probabilities in table 5. What is perhaps even more interesting is that using any Normalizing Flow method in the post-process period, outperforms the performances obtained by that same flow during training, which leads to a powerful method for obtaining models with correct interpretation of its uncertainty estimates. Calibrating with a NF results to be an easier optimization, converging to better results. The results obtained are comparable to the state-of-the-art (R^2 for: $\zeta = 0.850$ and $T^F_{vir} = 0.980$ in [7], $\Omega_m = 0.997, \sigma_8 = 0.997$ in [9]), but the architecture used in the present paper is considerably smaller (250,185 trainable parameters) compared with the networks employed in [7, 9] with ~ 10 millions. Furthermore as discussed in this section our approach has the advantage to be able to provide uncertainty estimation for the predicted parameters.

In order to compare the methods used so far, we chose an example from the test set, and produced the posterior and marginalized distributions of the parameters, obtaining the results displayed in figure 4. We

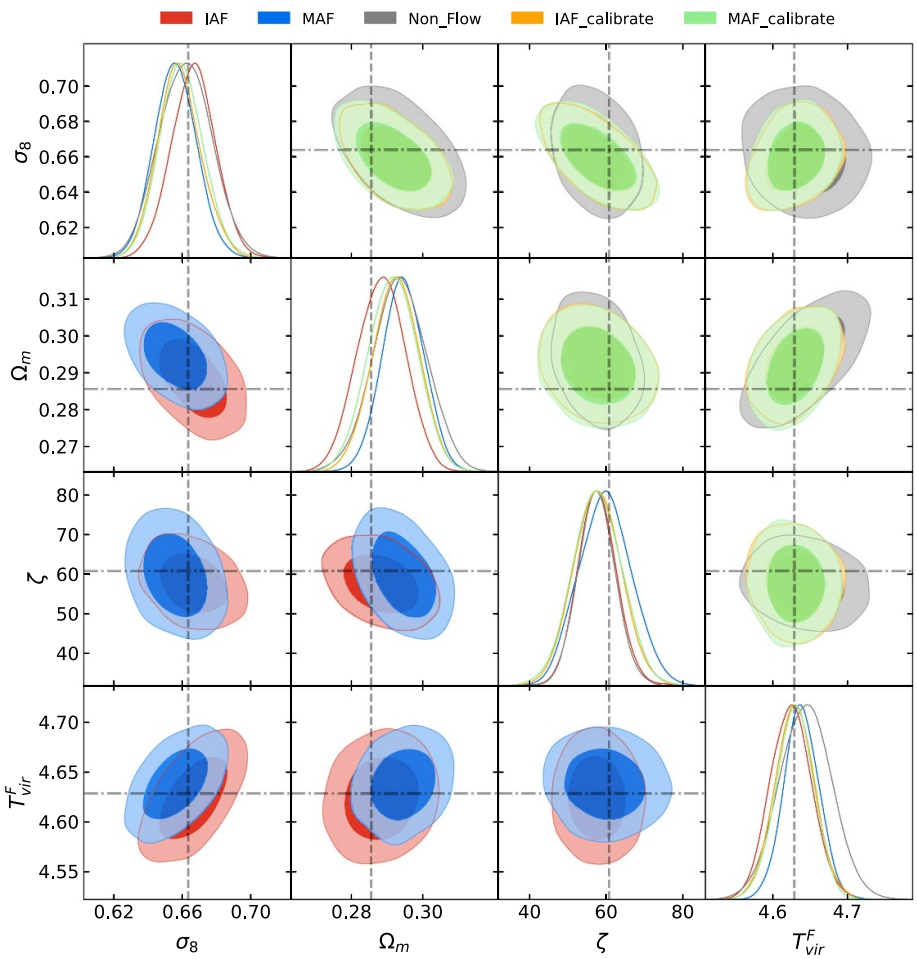


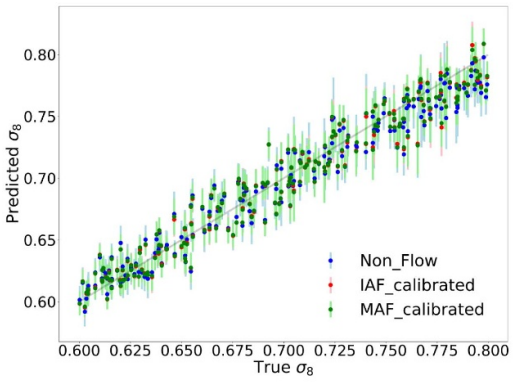
Figure 4. Posterior distributions of the parameters for one example in the test set using Normalizing flows after calibration. The dashed lines stand for the real values. The contour regions in the two-dimensional posteriors stand for 68 and 95% confidence levels.

can observe that after calibration, the contours produced by MAF becomes wider solving the underestimation found during training. Moreover, the contours of MAF and IAF applied in calibration overlap, and they are smaller compared with the base experiment, while Flows applied during training produce better results only for IAF. The credible intervals at 95% are shown in table 6. There we can see the effect of the flows on the performance of the network, allowing for a reduction of the uncertainty intervals. In appendix B we report the correlation matrix for IAF and MAF.

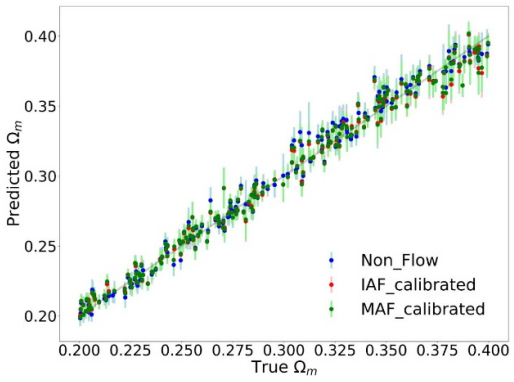
Finally, in figure 5 we plot the predicted and true values of the cosmological and astrophysical parameters using the model calibrated with IAF and MAF Flows. The error bars displayed in the plots correspond to both aleatoric and epistemic uncertainties. Here we observe that σ_8 parameter contains larger errors which means this parameter is the most difficult to predict accurately, this could be a consequence of the 21cm signal being less sensitive to the effects of the density field than the IGM properties [7, 9]. The ionizing efficiency, ζ, presents instead accurate predictions at low values, which are getting progressively less accurate and less precise at larger values. This fact could be explained due to the limited information at high redshifts (which can be reduced by assuming not spin temperature saturation) and also, the small variability of the brightness temperature maps at lower redshift with respect to large values of ζ.

5.4. Comparison with MCMC based frameworks

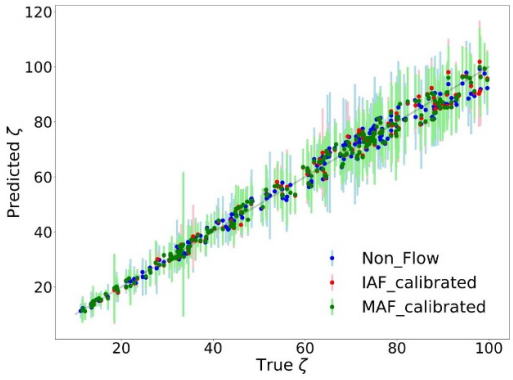
While a proper quantitative comparison with MCMC methods is outside the scope of the present work, it is important to notice that although MCMC methods converge asymptotically to the exact posterior, this can take a long time due to their high variance [41], and assessing their accuracy or evaluating the convergence can be very difficult. In [42, 43] the authors develop MCMC analysis tools to constrain the EoR astrophysical parameters; however, extensions such as the IGM heating, the estimation of Cosmological and additional instrumental parameters, might yield to a computationally intractable problem. An alternative to overcome this problem is the use of emulators [44] which can efficiently provide the summary statistics from



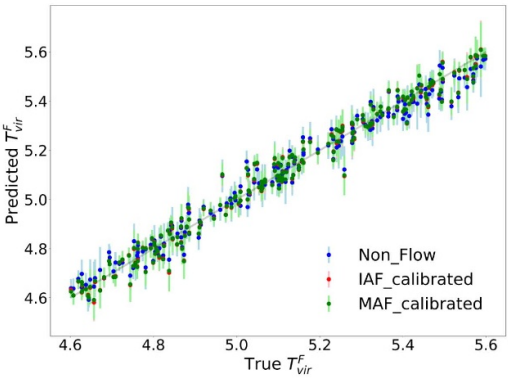
(a) Predicted σ_8 value vs true value.



(b) Predicted Ω_m value vs true value.



(c) Predicted ζ value vs true value.



(d) Predicted T_{vir}^F value vs true value.

Figure 5. Parameter inference for different Flow methods using the test set after calibration. The images in each panel contain the true values vs the predicted parameter values. The shadow lines represent the total uncertainty, epistemic plus aleatoric.

simulations enhancing the speed of MCMC sampling processes, but without the ability to extract non-Gaussian information if the power spectrum is only used during inference. Another alternative presented in the present paper is making use of VI which has the advantage of cheaply computing the approximate posterior distribution and makes it easier to assess convergence. Inference for a 21cm map is performed in approximately 5 seconds, using 3500 samples to estimate the approximate posterior on a GeForce GTX 1080 TI. Notice that VI is prone to underestimating their errors, nonetheless calibration methods like the ones applied in this paper aim to solve this issue and pave the way towards providing reliable uncertainty estimates in low-cost computation. Additionally, we can even combine VI and MCMC and leveraging the advantages of both inference techniques [45].

6. Conclusions

We presented the first study using a Bayesian Neural Network and Normalizing Flows to obtain credible estimates for astrophysical and cosmological parameters from 21cm signals. These methods offer alternative ways different from MCMC to make inference and recover the information in the 21cm observations. Firstly, we show that Flipout outperforms Dropout and is able both to better estimate parameters correlations and to obtain a better coefficient of determination. Comparing with existing literature, we obtain comparable performances, while using a relatively smaller network than [7, 9], furthermore by using a BNN, based on Variational Inference, we can estimate the confidence intervals for the predictions and the parameters uncertainties correlations. The 21 cm signal is highly non-Gaussian due to the complex physics involved during the EoR. Normalising Flows provide a flexible likelihood model capable to better capture complex information encoded in the dataset. This improves the performance of the network, training with flows achieves better NLL values than experiments without Flows (tables 1 and 3). Additionally we propose novel calibration methods employing flows after training, showing how this method provide accurate uncertainty estimates and high prediction of the parameters regardless of the flow used (table 6). Fine tuning the last layer, in combination with NFs leads to a simple, fast and effective technique for calibrating BNNs.

Calibration via hyperparameter tuning is expensive because needs a separate training for each hyperparameter combination. Hyperparameter tuning (like dropout rate for Dropout or L2 regularizer for Flipout) is modulating the variance of the distribution of the weights during training. Furthermore we found in practice that tuning hyperparameters to obtain a calibrated network reaches a sub-optimal configuration (in terms of coefficient of determination R^2) compared to a network trained with higher variance in the weights distribution, but calibrated afterwards (tables 3 and 4).

As future perspective, we plan on evaluating the performances of different network architectures (also in particular residual networks) and estimate the cosmological and astrophysical parameters in the presence of realistic noise from instruments of the future 21 cm surveys and in other astrophysical dataset.

Acknowledgment

H.J. Hortúa, R. Volpi, and L. Malagò are supported by the DeepRiemann project, co-funded by the European Regional Development Fund and the Romanian Government through the Competitiveness Operational Programme 2014-2020, Action 1.1.4, project ID P_37_714, contract no. 136/27.09.2016.

Appendix A

In tables A1-A2 we report different experiments, to determine the most adequate technique for estimating the parameters from the 21 cm dataset. First, we found that sampling more than once during training improves the results. Second, Flipout does a good job for extracting the information in the 21 cm images rather than other techniques such as Dropout.

Finally, we observe that Dropout underestimates its uncertainties while Flipout overestimates its uncertainties, therefore methods for calibration should be used before reporting the predictions. The

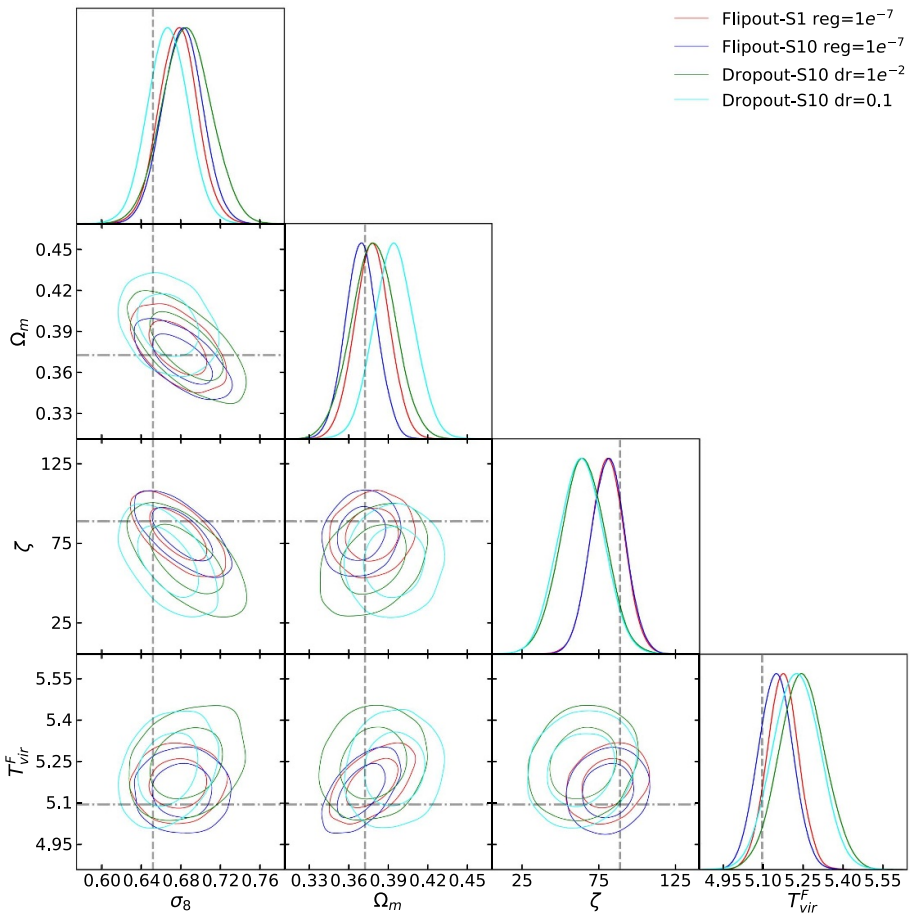


Figure A1. One and two-dimensional posterior distributions of the parameters for one example in the test set. Each color represents different experiments.

Table A1. Metrics for all Dropout experiments: dr= $(1e^{-2}, 0.1)$, reg = $1e^{-5}$. In each experiment, we sample once and ten times during training.

	Dropout dr= $1e^{-2}$		Dropout dr= 0.1	
	Sample = 1	Sample = 10	Sample = 1	Sample = 10
NLL	-0.18	-0.74	0.99	0.28
R^2	0.77	0.85	0.70	0.78
68% C.L.	65.3	68.1	66.7	65.0
95% C.L.	94.1	95.5	92.8	92.2
99% C.L.	99.3	99.7	98.7	98.7

Table A2. Metrics for all Flipout experiments: reg = $(1e^{-5}, 1e^{-7})$. In each experiment, we sample once and ten times during training.

	Flipout reg= 1e−7		Flipout reg= $1e^{-5}$	
	Sample = 1	Sample = 10	Sample = 1	Sample = 10
NLL	-2.30	-2.94	-1.81	-2.00
R^2	0.91	0.94	0.84	0.84
68% C.L.	75.5	73.0	76.2	76.4
95% C.L.	97.2	96.8	97.6	97.5
99% C.L.	99.5	99.8	99.8	99.8

confidence level reported in tables A1-A2, are computed with the method explained in section 4.1. The contour regions for the best results are also reported in figure A1. The R^2 for Flipout is reported in table A2.

Appendix B

In this appendix we show the correlation matrix for the example $\sigma_8 = 0.664$, $\Omega_m = 0.285$, $\zeta = 60.750$ and $T^F_{vir} = 4.620$, for IAF and MAF after calibration:

$$IAF = \begin{pmatrix} 1 & -0.483 & -0.600 & 0.284 \\ -0.483 & 1 & -0.191 & 0.360 \\ -0.600 & -0.191 & 1 & -0.103 \\ 0.284 & 0.360 & -0.103 & 1 \end{pmatrix} \tag{B1}$$

$$MAF = \begin{pmatrix} 1 & -0.493 & -0.588 & 0.222 \\ -0.493 & 1 & -0.191 & 0.365 \\ -0.588 & -0.191 & 1 & -0.09 \\ 0.222 & 0.365 & -0.09 & 1 \end{pmatrix}$$

The above values present a quantitative measure of the correlation displayed in the upper part of figure 5.

References

[1] Pritchard J R and Loeb A 2012 21 cm cosmology in the 21st century *Rep. Prog. Phys.* **75** 086901

[2] McQuinn M, Zahn O, Zaldarriaga M, Hernquist L and Furlanetto S R 2006 Cosmological parameter estimation using 21 cm radiation from the epoch of reionization *Astrophys.* J. **653** 815–34

[3] Kaisey S 2006 Mandel and Matias Zaldarriaga. Weak gravitational lensing of high-redshift 21 cm power spectra *Astrophys.* J. **647** 719–36

[4] Liu A, Pritchard J R, Allison R, Parsons A R, Seljak Uš and Sherwin B D 2016 Eliminating the optical depth nuisance from the CMB with 21 cm cosmology *Phys. Rev.* D **93** 043013

[5] Greig B and Mesinger A 2015 21CMMC: an MCMC analysis tool enabling astrophysical parameter studies of the cosmic 21 cm signal *Mon. Not. R. Astron. Soc.* **449** 4246–63

[6] Hassan S, Liu A, Kohn S and Plante P L 2018 Identifying reionization sources from 21cm maps using Convolutional Neural Networks *Mon. Not. R. Astron. Soc.* **483** 2524–37

[7] Gillet N, Mesinger A, Greig B, Liu A and Ucci G 2019 Deep learning from 21-cm tomography of the cosmic dawn and reionization *Mon. Not. R. Astron. Soc.* **484** 282–93

[8] La Plante P and Ntampaka M 2019 Machine learning applied to the reionization history of the universe in the 21 cm signal *Astrophys.* J. **880** 110

[9] Hassan S, Andrianomena S and Doughty C 2020 Constraining the astrophysics and cosmology from 21cm tomography using deep learning with the SKA *Mon. Not. R. Astron. Soc.* **494** 5761–74

[10] Graves A 2011 Practical variational inference for neural networks In *Advances in Neural Information Processing Systems 24* ed Shawe-Taylor J, Zemel R S, Bartlett P L, Pereira F and Weinberger K Q Curran Associates, Inc. pp 2348–56

[11] Kwon Y, Won J-H, Kim B J and Paik M C 2020 Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation *Comput. Stat. Data Anal.* **142** 106816
[12] Cobb A D *et al* 2019 An ensemble of Bayesian neural networks for exoplanetary atmospheric retrieval *Astrophys. J.* **158** 33
[13] Hortúa H J, Volpi R and Malagò L 2020 Parameters estimation from the 21 cm signal using variational inference, FSAI workshop, ICLR
[14] Hortua H J, Volpi R, Marinelli D and Malagò L 2019 Parameters Estimation for the Cosmic Microwave Background with Bayesian Neural Networks arXiv:1911.08508
[15] Gal Y and Ghahramani Z 2015 Bayesian convolutional neural networks with Bernoulli approximate variational inference
[16] Dorta G, Vicente S, Agapito L, Campbell N D F and Simpson I 2018 Structured uncertainty prediction networks *2018 IEEE/ Conf. on Computer Vision and Pattern Recognition*
[17] Gal Y and Ghahramani Z 2015 Dropout as a Bayesian approximation: Insights and applications *Deep Learning Workshop, Icml*
[18] Kendall A and Gal Y 2017 What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30* ed Guyon I, Luxburg U V, Bengio S, Wallach H, Fergus R, Vishwanathan S and Garnett R pp 5574–84 Curran Associates, Inc.
[19] Wen Y, Vicol P, Jimmy B, Tran D and Grosse R 2018 Flipout: Efficient pseudo-independent weight perturbations on mini-batches *Int. Conf. on Learning Representations*
[20] Trippe B L and Turner R E 2018 Conditional Density Estimation with Bayesian Normalising Flows arXiv:1802.04908
[21] Papamakarios G, Nalisnick E, Rezende D J, Mohamed S and Lakshminarayanan B 2019 Normalizing flows for probabilistic modeling and inference arXiv:1912.02762
[22] Kobyzev I, Prince S and Brubaker M A 2019 Normalizing flows: An introduction and review of current methods arXiv:1908.09257
[23] Huang C-W, Krueger D, Lacoste A and Courville A 2018 Neural autoregressive flows arXiv:1804.00779
[24] Papamakarios G, Pavlakou T and Murray I 2017 Masked autoregressive flow for density estimation *Advances in Neural Information Processing Systems* pp 2338–47
[25] Kingma D P, Salimans T, Jozefowicz R, Chen Xi, Sutskever I and Welling M 2016 Improved variational inference with inverse autoregressive flow *Advances in Neural Information Processing Systems* pp 4743–51
[26] Dinh L, Sohl-Dickstein J and Bengio S 2016 Density estimation using real NVP arXiv:1605.08803
[27] Chen T Q, Rubanova Y, Bettencourt J and Duvenaud D K 2018 Neural ordinary differential equations *Advances in Neural Information Processing Systems* pp 6571–83
[28] Grathwohl W, Chen R T Q, Bettencourt J, Sutskever I and Duvenaud D 2018 Ffjord: Free-form continuous dynamics for scalable reversible generative models arXiv:1810.01367
[29] Mesinger A, Furlanetto S and Cen R 02 2011 21CMFAST: a fast, seminumerical simulation of the high-redshift 21-cm signal *Mon. Not. R. Astron. Soc.* **411** 955–72
[30] Furlanetto S R, [Peng Oh] S and Briggs F H 2006 Cosmology at low frequencies: The 21cm transition and the high-redshift universe *Phys. Rep.* **433** 181–301
[31] Choudhury M, Datta A and Chakraborty A 2019 Extracting the 21 cm global signal using artificial neural networks *Mon. Not. R. Astron. Soc.* **491** 4031–44
[32] Santos Mario G, Amblard A, Pritchard J, Trac H, Cen R and Cooray A 2008 Cosmic reionization and the 21 cm signal: Comparison between an analytical model and a simulation *Astrophys. J.* **689** 1–16
[33] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv:1409.1556
[34] Ioffe S 2017 Batch renormalization: Towards reducing minibatch dependence in batch-normalized models *arXiv:1702.03275*
[35] Guo C, Pleiss G, Sun Y and Weinberger K Q 2017 On Calibration of Modern Neural Networks arXiv:1706.04599
[36] Levi D, Gispan L, Giladi N and Fetaya E 2019 Evaluating and calibrating uncertainty prediction in regression tasks arXiv:1905.11659
[37] Li Y and Gal Y p 2017 Dropout Inference in Bayesian Neural Networks with Alpha-divergences arXiv:1703.02914
[38] Levasseur L P, Hezaveh Y D and Wechsler R H 2017 Uncertainties in parameters estimated with neural networks: Application to strong gravitational lensing *Astrophys. J.* **850** L7
[39] Lewis A 2019 GetDist: a Python package for analysing Monte Carlo samples arXiv:1910.13970
[40] Zhang J-F, Gao Li-Y, Dong-Ze H and Zhang X 2019 Improving cosmological parameter estimation with the future 21 cm observation from ska *Phys. Lett.* B **799** 135064
[41] Ruiz F J R and Titsias M K 2019 A contrastive divergence for combining variational inference and MCMC arXiv:1905.04062 [stat.ML]
[42] Greig B and Mesinger A 2015 21CMMC: an MCMC analysis tool enabling astrophysical parameter studies of the cosmic 21 cm signal *Mon. Not. R. Astron. Soc.* **449** 4246–63
[43] Greig B and Mesinger A 2017 Simultaneously constraining the astrophysics of reionisation and the epoch of heating with 21CMMC *Proc. of the Int. Astronomical Union* vol 12 pp 18–21
[44] Kern N S, Liu A, Parsons A R, Mesinger A and Greig B 2017 Emulating simulations of cosmic dawn for 21 cm power spectrum constraints on cosmology, reionization and x-ray heating *Astrophys. J.* **848** 23
[45] Salimans T, Kingma D P and Welling M 2014 Markov chain Monte Carlo and variational inference: Bridging the gap arXiv:1410.6460