Test of artificial neural networks in likelihood-free cosmological constraints: A comparison of information maximizing neural networks and denoising autoencoder

Jie-Feng Chen

Institute for Frontiers in Astronomy and Astrophysics, Beijing Normal University, Beijing 102206, China and Department of Astronomy, Beijing Normal University, Beijing 100875, China

Yu-Chen Wang

Kavli Institute for Astronomy and Astrophysics, Peking University, Beijing 100871, China; Department of Astronomy, School of Physics, Peking University, Beijing 100871, China and Department of Physics, Beijing Normal University, Beijing 100875, China

Tingting Zhang[*]

College of Command and Control Engineering, Army Engineering University, Nanjing 210017, China

Tong-Jie Zhang[†]

Institute for Frontiers in Astronomy and Astrophysics, Department of Astronomy, Beijing Normal University, Beijing 102206, China; Department of Astronomy, Beijing Normal University, Beijing 100875, China and Institute for Astronomical Science, Dezhou University, Dezhou 253023, China

(Received 26 October 2022; accepted 9 January 2023; published 13 March 2023)

In the procedure of constraining the cosmological parameters with the observational Hubble data (OHD), the combination of masked autoregressive flow (MAF) and denoising autoencoder (DAE_ can perform with good result. The above combination extracts the features from OHD with DAE, and estimates the posterior distribution of H_0, Ω_m, Ω_Λ with MAF. We ask whether we can find a better tool to compress large data in order to gain better results while constraining the cosmological parameters. Information maximizing neural networks (IMNN), a kind of simulation-based machine learning technique, was proposed at an earlier time. In a series of numerical examples, the results show that IMNN can find optimal, nonlinear summaries robustly. In this work, we mainly compare the dimensionality reduction capabilities of IMNN and DAE. We use IMNN and DAE to compress the data into different dimensions and set different learning rates for MAF to calculate the posterior. Meanwhile, the training data and mock OHD are generated with a simple Gaussian likelihood, the spatially flat ΛCDM model and the real OHD data. To avoid the complex calculation in comparing the posterior directly, we set different criteria to compare IMNN and DAE.

DOI: 10.1103/PhysRevD.107.063517

I. INTRODUCTION

Constraining cosmological parameters is a basic task in cosmology. To evaluate the parameters, the common method is to calculate an intractable likelihood directly to perform Bayesian inference with the existing observational datasets, e.g., observational Hubble parameter data (OHD, [1]), type Ia supernovae (SNe Ia, [2]), cosmic microwave background [2], and large-scale structures [3,4]. Approximate Bayesian computation (ABC) has also shown good performance in many astronomical tasks, such as galaxy evolution [5] and SN Ia cosmology [6]. Nevertheless, according to [7], conventional ABC algorithms may suffer from noisy computations.

In the past few decades, the artificial neural networks (ANN) developed rapidly and were gradually used to constrain the cosmological parameters [8–11]. Recently, a likelihood-free inference procedure using a denoising autoencoder (DAE) and masked autoregressive flow (MAF) was proposed by us [12]. In our previous work, the combination of MAF and DAE was compared to MCMC, which is the Markov Chain Monte Carlo method, and behaved well in calculating the posterior distribution $[P(\boldsymbol{\theta}|\boldsymbol{H}_{\rm obs})]$ of Ω_Λ, Ω_m, and H_0. We proved that MAF could give similar results as MCMC, which means that at least

[*] 101101964@seu.edu.cn
[†] tjzhang@bnu.edu.cn

MAF could be the substitute while we estimate the cosmological parameters. MAF was proposed by [13], in their several experiments, MAF gave accurate estimations of distributions and did well in likelihood-free inference [14]. DAE [15] is an ANN that can encode data by extracting the data features. With the DAE, we can obtain low-dimensional representative features of the input data without an artificial choice of statistics.

The higher-dimensional data from simulation or computational resources is inevitable in likelihood-free inference. For this reason, we need to find a good tool to reduce the dimensionality of data. At an earlier time, [16] proposed a kind of ANN named "information maximizing neural networks" (IMNNs), which can transform data into summaries by maximizing the Fisher information at fiducial values. In the examples proposed by [16,17], IMNN performed well in finding the informative data summaries, which means maybe we can test whether IMNN can be a substitute for DAE.

In this work, we attempt to compare the dimensionality reduction capabilities of IMNN and DAE. Like the procedure of constraining cosmological parameters applied by [12], we use DAE and IMNN to reduce the higher-dimensional OHD data and then use MAF to estimate the distributions of cosmological parameters with the low-dimensional features. Besides, we also estimate the distribution, which will be treated as the standard distribution, with MAF and the original-dimensional OHD data. In the rest of this article, we use MAF-IMNN to represent the combination of MAF and IMNN, and MAF-DAE to represent the combination of MAF and DAE. In Sec. II, we review the procedure of cosmological constraints using MAF-DAE. In Sec. III, we discuss the theory of IMNN. In Sec. IV, we will show the results of constraint with OHD in different ways, and explore the possibility of evaluating parameters with MAF-IMNN. In Sec. V, we compare the DAE and IMNN with different criteria. Finally, in Sec. VI, we conclude and discuss.

II. MAF-DAE FOR PARAMETER CONSTRAINT

MAF, the combination of normalizing flow and masked autoencoder for distribution estimation (MADE [18], one kind of autoregressive model), was proposed by [13]. MADE and normalizing flows are two kinds of neural density estimators, which can estimate the density distribution of the parameters.

With the MADE, the conditional distribution $P(\boldsymbol{x}|\boldsymbol{y})$ can be written in the form:

$$P(\boldsymbol{x}|\boldsymbol{y}) = \prod_d P(x_d|\boldsymbol{x}_{1:d-1}, y), \tag{1}$$

where $\boldsymbol{x}_{1:d-1} = (x_1, x_2, \ldots, x_{d-1})$, which means $\boldsymbol{x}$ has $\boldsymbol{d}$ dimensions. And then Eq. (1) will be parametrized into Gaussian distribution, the mean and the log standard deviation will be calculated by the neural network. In other words, we can obtain the parameters of all of these conditional distributions. The concise structure of the MADE is shown in Fig. 1.

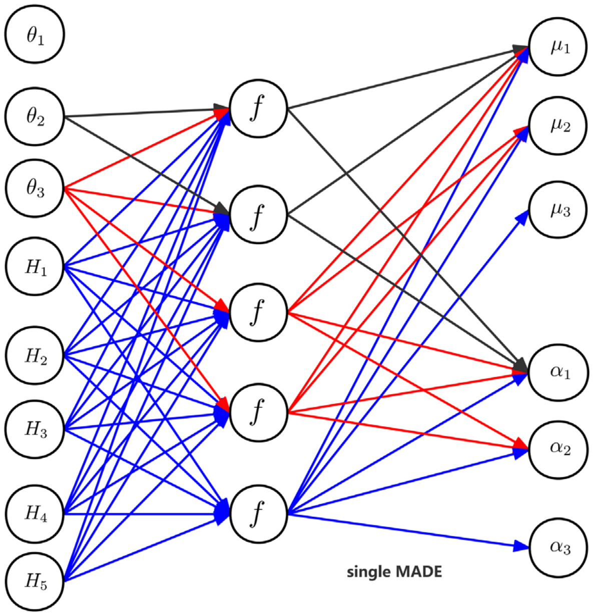

FIG. 1. The concise structure of the MADE. The blue lines mean full connection. The black lines and the red lines mean a set of connections is removed ensuring that MADE satisfies the autoregressive property. In autoregressive property, a unit can only connect to one more advanced unit. To explain the MAF, we only draw 5 H_is (the OHD) correlating to 3 θ_is (Ω_Λ, Ω_m, and H_0) in this figure, but in this work we actually applied 31 H_is.

According to the normalizing flows [19], the density $P(x)$ can be obtained from a base density $\pi_u(\boldsymbol{u})$ with an invertible differentiable transformation f:

$$P(x) = \pi_u(f^{-1}(x))\left|\det\left(\frac{\partial f^{-1}}{\partial x}\right)\right|, \tag{2}$$

where $\boldsymbol{u} = f^{-1}(x)$ and $\boldsymbol{u} \sim \pi_u(\boldsymbol{u})$ {usually $\pi_u(\boldsymbol{u})$ is a standard Gaussian distribution [$\boldsymbol{u} \sim \mathcal{N}(0, \mathrm{I})$]}. With the normalizing flows and autoregressive models, each of the conditionals $P(x_d|\boldsymbol{x}_{1:d-1}, \boldsymbol{y})$ can be parametrized as Gaussian distribution. In this case, the dth conditional is

$$P(x_d|x_{1:d-1}) = \mathcal{N}(x_d|\mu_d, (\exp\ \alpha_d)^2), \tag{3}$$

$$\pi(u_d) = \mathcal{N}(u_d; 0, 1), \tag{4}$$

and

$$x_d = f_d(u_d; \alpha_d, \mu_d) = \exp(\alpha_d)u_d + \mu_d. \tag{5}$$

The unconstrained scalar functions $\mu_i = f_{\mu i}(x_{1:i-1})$ and $\alpha_i = f_{\alpha i}(x_{1:i-1})$ compute the mean and log standard, respectively.

When doing the cosmological inference, we can represent $\boldsymbol{x}$ as the $\boldsymbol{\theta}$ in the Hubble parameters such as Ω_Λ, Ω_m, H_0, and represent $\boldsymbol{y}$ as $\boldsymbol{H}_{\mathrm{obs}}$. In this case $\boldsymbol{\theta}_i$ can be written in the form:

$$\theta_i = u_i \exp(\alpha_i) + \mu_i, \tag{6}$$

$$P(\theta) = \pi_u(f^{-1}(\theta)) \left| \det\left(\frac{\partial f^{-1}(\theta)}{\partial \theta}\right) \right|, \tag{7}$$

where $\mu_i = f_{\mu i}(\boldsymbol{\theta}_{1:i-1})$, $\alpha_i = f_{\alpha i}(\boldsymbol{\theta}_{1:i-1})$, and $u_i \sim \mathcal{N}(0,1)$, so with the MADE we can get the $\boldsymbol{\theta} = f(\boldsymbol{u};\boldsymbol{H})$ where $\boldsymbol{u} \sim \mathcal{N}(0, I)$. A single MADE may not fit the distribution well, which means that the corresponding random numbers $\boldsymbol{u} = f^{-1}(\boldsymbol{\theta};\boldsymbol{H})$ transformed from the training data $\boldsymbol{\theta}$ were not standard Gaussian (Also $\boldsymbol{\theta} = f(\boldsymbol{u};\boldsymbol{H})$). To improve the performance of MADE, we can stack several MADEs as a normalizing flow. According to [13], masked autoregressive flow (MAF) is the implementation of stacking MADEs into a flow. The loss function of MAF is defined by the negative log probability:

$$L = -\sum_n \ln P(\boldsymbol{\theta}_n|\boldsymbol{H}_n). \tag{8}$$

where n means the nth data in the training data.

The training set of the MAF should be the $\{\boldsymbol{\theta}_n, \boldsymbol{H}_n\}$, where the $\boldsymbol{\theta}$ means Ω_Λ, Ω_m, and H_0, and $\boldsymbol{H}$ means different dimensional mock OHD. In this work, we trained MAF to find the correlation between $\boldsymbol{\theta}$ and $\boldsymbol{H}$ (5-dimensional H, 10-dimensional H, 15-dimensional H, 20-dimensional H, 31-dimensional H). After training, MAF can be used to estimate the $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ with H_{obs}, which is in its own 31-dimension or being compressed into 20-dimension, 15-dimension, 10-dimension, and 5-dimension. The input of a trained MAF is a set of H_{obs}, while the output is 100000 (we can also set another number such as 10000, 50000.) sets of Ω_Λ, Ω_m and H_0, which can be used to calculated the $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ directly. Certainly, we can also input H_{fid} and calculate the $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{fid}})$.

A. Denoising autoencoders (DAE)

DAE is a special kind of autoencoder. A basic autoencoder is in a special neural network architecture, which is composed of an encoder and a decoder, can learn efficient, lower-dimensional codings of the input data. The autoencoder is trained with unsupervised learning to obtain lower-dimensional features of the input data by encoder. In the output part of the autoencoder (decoder), the lower-dimensional features can be reconstructed to original-dimensional data. Therefore, the input layer has the same number of neurons as the output layer. The training of the autoencoder is to minimize the error between the input and the output. The concise structure of the autoencoder is shown in Fig. 2.

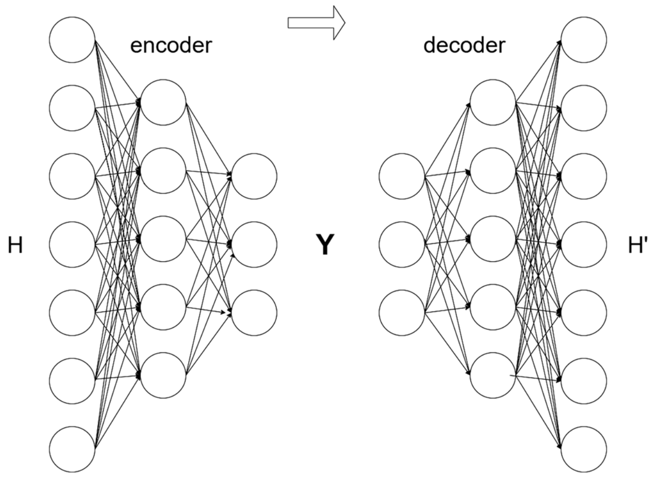

FIG. 2. The concise structure of the autoencoder and the procedure of compressing data. An autoencoder consists of an encoder and a decoder. With the encoder, the input $\boldsymbol{H}$ can be compressed into lower-dimensional $\boldsymbol{y}$, so the information in $\boldsymbol{H}$ can be represented by $\boldsymbol{y}$. With the decoder, $\boldsymbol{y}$ can be reconstructed to original-dimensional $\boldsymbol{H}'$. Usually, we train the autoencoder by minimizing the error between $\boldsymbol{H}$ and $\boldsymbol{H}'$.

DAE is trained with noise-free reconstruction criterion and noisy inputs, so that it can not only extract the robust features from the input data but also significantly reduce the noise. In this work, the DAE was trained with noise-free fiducial values $\boldsymbol{H}_{\mathrm{fid}}$ as labels and noisy simulated data $\boldsymbol{H}$ as the inputs in order to reduce the noise level of the $\boldsymbol{H}_{\mathrm{obs}}$ and preserve more information. After training, DAE can compress $\boldsymbol{H}_{\mathrm{obs}}$ to low-dimensional $\boldsymbol{y}$ [$\boldsymbol{y} = f_e(\boldsymbol{H})$]. Usually, an autoencoder is trained by minimizing the reconstruction error, i.e., the mean squared error (MSE) between reconstructed data $\boldsymbol{H}'$ and the label $\boldsymbol{H}_{\mathrm{fid}}$. However, to make sure $\boldsymbol{y}$ may contain more information about $\boldsymbol{\theta}$ and avoid giving too big variance of $P(\boldsymbol{y}|\boldsymbol{\theta})$, our previous work [12] proposed a complete batch loss function to require the mean of the conditional $P(\boldsymbol{y}|\boldsymbol{\theta})$ relies linearly on $\boldsymbol{\theta}$. The loss function consists of reconstruction MSE and encoding variance:

$$\begin{aligned} L_{AE} &= \mathrm{mean}\{(\boldsymbol{X}' - \boldsymbol{X}_{\mathrm{fid}}) \circ (\boldsymbol{X}' - \boldsymbol{X}_{\mathrm{fid}})\} \\ &\quad + \mathrm{var}\{\boldsymbol{Y} - \boldsymbol{\Theta}\boldsymbol{\Theta}^{+}\boldsymbol{Y}\}, \end{aligned} \tag{9}$$

where

$$\boldsymbol{X}_{\mathrm{fid}} = \begin{pmatrix} \boldsymbol{H}^T_{\mathrm{fid},1} \\ \boldsymbol{H}^T_{\mathrm{fid},2} \\ . \\ . \\ . \end{pmatrix}, \qquad \boldsymbol{X}_{\mathrm{fid}} = \begin{pmatrix} \boldsymbol{H}'^T_1 \\ \boldsymbol{H}'^T_2 \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} \tag{10}$$

and

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}_1^T \\ \boldsymbol{y}_2^T \\ \vdots \\ \vdots \\ \vdots \end{pmatrix}, \qquad \boldsymbol{\Theta} = \begin{pmatrix} 1\boldsymbol{\theta}_1^T \\ 1\boldsymbol{\theta}_2^T \\ \vdots \\ \vdots \\ \vdots \end{pmatrix}. \tag{11}$$

The $\boldsymbol{\Theta}^+$ in Eq. (9) is the pseudoinverse (Moore-Penrose inverse) of $\boldsymbol{\Theta}$. In this way, the loss function can be easily evaluated on the training set. In this work, we stick with this training method.

B. The simulated data

The real OHD is composed of $z_i, H(z_i)$ and σ_i, where z_i is the redshift, and $H(z_i)$ is the corresponding Hubble parameter and σ_i is the corresponding uncertainty. The 31 OHD data we use in this work are evaluated with the cosmic chronometer method, which are given in [20–23,23,24], [25,26], and are shown in Fig. 3. Based on the real data, we can generate training data and constrain parameters with ANNs.

According to the flat ΛCDM model, the Hubble parameter is expressed by redshift z with the simple formula:

$$H(z) = H_0\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}, \tag{12}$$

where the H_0 is the Hubble constant, or the nonflat ΛCDM model:

$$H(z) = H_0\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda + \Omega_k(1+z)^2}. \tag{13}$$

The parameters $H_0, \Omega_m, \Omega_\Lambda$ in Eqs. (12) and (13) are randomly sampled from the range [0, 100], [0, 1], and [0, 1]. As illustrated in [12], when hard boundaries are added to the prior, the new posterior is almost the same as the original one, provided that the boundaries encloses the likely region of the posterior. Therefore, there is no special requirement for the sampling interval. With the random sampled parameters, as well as the $\mathbf{z} = z_i$ from the 31 OHD data, the $\boldsymbol{H}_{\mathrm{fid}}(z_i)$ can be easily obtained by Eqs. (12) and (13). Finally, by sampling the $\Delta\boldsymbol{H}_i$ in $\mathcal{N}(0, \sigma_i^2)$ [12,27,28], we can obtain the with the formula:

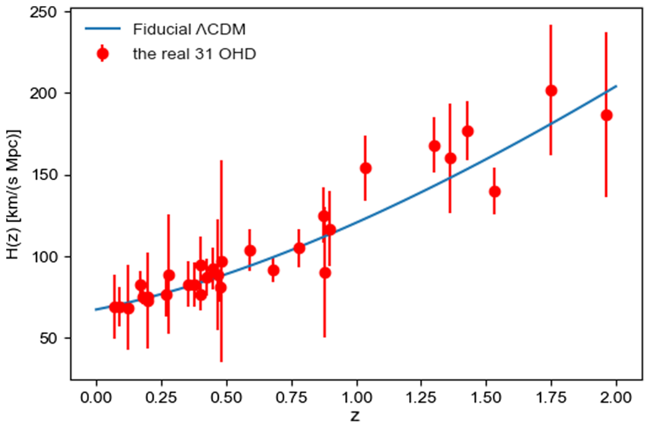

FIG. 3. The 31 real OHD datapoints and spatially flat ΛCDM model.

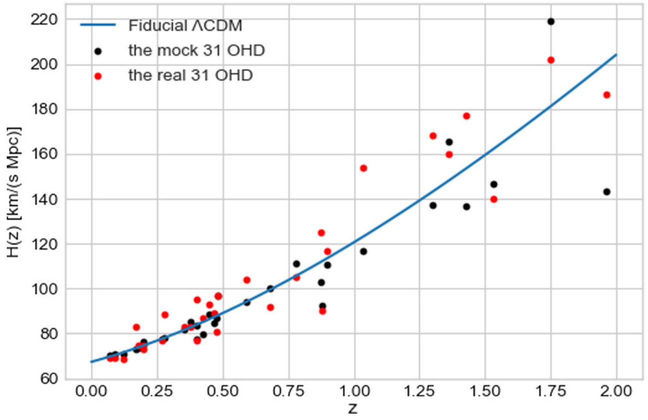

FIG. 4. 31 mock datapoints are made with spatially flat ΛCDM model and the Gaussian sample. The mock OHD is based on the real OHD.

$$H_{\mathrm{moc},i} = H_{\mathrm{fid}}(z_i) + \Delta H_i. \tag{14}$$

The training data, which consists of the simulated $\boldsymbol{H}_{\mathrm{sim},i}$ and the corresponding $\theta = (H_0, \Omega_m, \Omega_\Lambda)$, should be large enough to train the ANNs, so we also set 8000 training data like [12]. We show one set of the training data in Fig. 4. Furthermore, in order to make a comprehensive comparison, we also simulate a new kind of mock OHD and training data by narrowing the range of the corresponding uncertainty in the Gaussian sample. One set of the mock OHD is shown in Fig. 5.

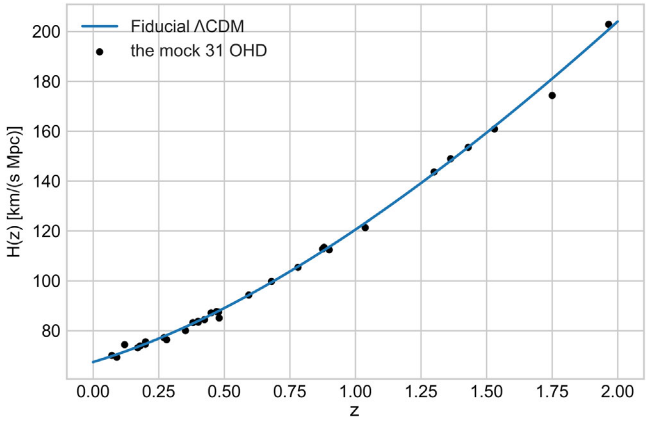

FIG. 5. The new mock OHD is also based on the real OHD. Because of the small range of the corresponding uncertainty in the Gaussian sample, the new mock 31 data points basically fit the flat ΛCDM model.

C. The procedure of constraining parameters

The procedure of constraining Ω_Λ, Ω_m, H_0 with MAF-DAE is summarized as below: (1) Generating 8000 training data $\{\boldsymbol{\theta}, \boldsymbol{H}\}$ and training a DAE with the training data; (2) Generating another set of training data and encoding the $\boldsymbol{H}_{\text{sim}}$ with the trained DAE to get lower-dimensional $\boldsymbol{H}_{\text{sim}}$; (3) Training a MAF with the lower-dimensional $\boldsymbol{H}_{\text{sim}}$ and corresponding parameters θ; (4) Encoding the real 31 OHD with the DAE and inputting the lower-dimensional OHD to the MAF to estimate the posterior distribution $P(\theta|\boldsymbol{H}_{\text{obs}})$.

III. MAF-IMNN FOR PARAMETER CONSTRAINT

According to the method evaluating the parameters with MAF-DAE mentioned above, we apply a similar procedure to constrain the cosmological parameters with MAF-IMNN in this paper. The procedure is summarized as below: (1) Generating training data $\{\boldsymbol{\theta}, \boldsymbol{H}\}$ with the same model and training a IMNN; (2) Generating 8000 training data and encoding the $\boldsymbol{H}_{\text{sim}}$ with the trained IMNN to get lower-dimensional $\boldsymbol{H}_{\text{sim}}$; (3) Training a MAF with the lower-dimensional $\boldsymbol{H}_{\text{sim}}$ and corresponding parameters θ; (4) Encoding the real 31 OHD with the IMNN and inputting the lower-dimensional OHD to the MAF to estimate the posterior distribution $P(\theta|\boldsymbol{H}_{\text{obs}})$. As the substitution of DAE, IMNN can find the most informative nonlinear data summaries by setting fiducial parameters and calculating the Fisher information matrix on the simulated data. Although IMNN is simulation-based, the examples proposed by [16] showed the training of the network seems fairly insensitive to the choice of fiducial parameter. In the rest of this section, we introduce the theory of IMNN briefly.

A. Fisher information and compression

The Fisher information [29–31] can measure how much information that an observable variable **d** contains about parameter $\boldsymbol{\theta}$. For this reason, the larger the Fisher information is, the more informative the data is. It can be obtained by calculating the variance of the partial derivative of the natural logarithm of the likelihood $\mathcal{L}(\mathbf{d}|\boldsymbol{\theta})$ with respect to the fiducial parameter value, $\boldsymbol{\theta}^{\text{fid}}$:

$$\mathbf{F}_{\alpha\beta}(\boldsymbol{\theta}) = -\left\langle \frac{\partial \ln \mathcal{L}(\mathbf{d}|\boldsymbol{\theta})}{\partial \theta_\alpha} \frac{\partial \ln \mathcal{L}(\mathbf{d}|\boldsymbol{\theta})}{\partial \theta_\beta} \right\rangle \bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{fid}}}, \quad (15)$$

where $\alpha, \beta \in [1, n_\theta]$ (where $\alpha \neq \beta$). In our work, we used ΛCDM model, therefore α and β represent Ω_Λ, Ω_m and H_0. If we use another model where theta has a higher dimension, the formula is still kept valid. If the likelihood is twice continuously differentiable, the expression of the Fisher information can be [30–32]:

$$\mathbf{F}_{\alpha\beta}(\boldsymbol{\theta}) = -\left\langle \frac{\partial^2 \ln \mathcal{L}(\mathbf{d}|\boldsymbol{\theta})}{\partial \theta_\alpha \partial \theta_\beta} \right\rangle \bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{fid}}}, \quad (16)$$

where the $\mathcal{L}(\mathbf{d}|\boldsymbol{\theta})$ is the likelihood function of the data **d** with the with $n_{\mathbf{d}}$ data points, and a set of n_θ parameters $\boldsymbol{\theta}$. We can constrain $\boldsymbol{\theta}$ in a smaller range if the $\mathcal{L}(\mathbf{d}|\boldsymbol{\theta})$ is sharp at a particular value. According to the Cramér-Rao bound [33,34], under certain conditions, we can calculate the maximum Fisher information to find the minimum variance of θ:

$$\langle(\theta_\alpha - \langle\theta_\alpha\rangle)(\theta_\beta - \langle\theta_\beta\rangle)\rangle \geq (\mathbf{F}^{-1})_{\alpha\beta}. \quad (17)$$

In particular, if the model of likelihood of the data **d** is Gaussian approximation, we can use massively optimized parameter estimation and data (MOPED) compression algorithm [35] to map the data to compressed summaries. While using the MOPED, the logarithm of the likelihood should be written as

$$-2 \ln \mathcal{L}(\mathbf{d}|\boldsymbol{\theta}) = (\mathbf{d} - \boldsymbol{\mu}(\boldsymbol{\theta}))^T \mathbf{C}^{-1} (\mathbf{d} - \boldsymbol{\mu}(\boldsymbol{\theta})) + \ln |2\pi \mathbf{C}|, \quad (18)$$

where $\boldsymbol{\mu}(\boldsymbol{\theta})$ is the mean of the parameters $\boldsymbol{\theta}$ and **C** is the covariance of the data **d**. Compared with MOPED, IMNN can map the data to compressed summaries without the limitation of the likelihood. f is the function that transforms $n_{\mathbf{d}}$ data **d** to $n_{\mathbf{s}}$ summary **x**, which means that $f: \mathbf{d} \to \mathbf{x}$. With the function f, the logarithm of the likelihood can be written as

$$-2 \ln \mathcal{L}(\mathbf{x}|\boldsymbol{\theta}) = (\mathbf{x} - \boldsymbol{\mu}_f(\boldsymbol{\theta}))^T \mathbf{C}_f^{-1} (\mathbf{x} - \boldsymbol{\mu}_f(\boldsymbol{\theta})), \quad (19)$$

where

$$\boldsymbol{\mu}_f(\boldsymbol{\theta}) = \frac{1}{n_s} \sum_{i=1}^{n_s} x_i^s, \quad (20)$$

is the mean value of $n_{\mathbf{s}}$ summaries $\{x_i^s | i \in [1, n_{\mathbf{s}}]\}$, and $\mathbf{C}_f^{-1}$ is the inverse of the covariance matrix:

$$(\mathbf{C}_f)_{\alpha\beta} = \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (\mathbf{x}_i^s - \boldsymbol{\mu}_f)_\alpha (\mathbf{x}_i^s - \boldsymbol{\mu}_f)_\beta. \quad (21)$$

While training, each summary $\mathbf{x}_i^s$ is obtained from f: $\mathbf{d}_i^s \to \mathbf{x}_i^s$, where $\mathbf{d}_i^s$ is from the simulation $\mathbf{d}_i^s = \mathbf{d}^s(\boldsymbol{\theta}, i)$ at the fiducial values $\boldsymbol{\theta}$. With Eqs. (16) and (19), the Fisher information matrix can be expressed in the form:

$$\mathbf{F}_{\alpha\beta} = \mathrm{Tr}[\boldsymbol{\mu}_{f,a}^T \mathbf{C}_f^{-1} \boldsymbol{\mu}_{f,\beta}]. \quad (22)$$

The $\boldsymbol{\mu}_{f,a}$ can be calculated by

$$\boldsymbol{\mu}_{f,\alpha} = \frac{\partial}{\partial \theta_\alpha} \frac{1}{n_s} \sum_{i=1}^{n_s} x_i^{s\text{fid}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \left(\frac{\partial x}{\partial \theta_\alpha}\right)_i^{s\text{ fid}}. \quad (23)$$

Note that the fiducial parameters are only used in the simulations, so we need to do some additional numerical differentiation to calculate $(\frac{\partial x}{\partial \theta_\alpha})_i^{s\mathrm{fid}}$ with these three copies of the simulation, $\mathbf{d}_i^{s\,\mathrm{fid}} = \mathbf{d}^s(\boldsymbol{\theta}^{\mathrm{fid}}, i)$, $\mathbf{d}_i^{s\,\mathrm{fid}-} = \mathbf{d}^s(\boldsymbol{\theta}^{\mathrm{fid}} - \Delta\boldsymbol{\theta}^-, i)$ and $\mathbf{d}_i^{s\,\mathrm{fid}+} = \mathbf{d}^s(\boldsymbol{\theta}^{\mathrm{fid}} + \Delta\boldsymbol{\theta}^+, i)$, where the $\Delta\boldsymbol{\theta}^\pm$ is the small deviation from the fiducial parameter values. With the above conditions, the $(\frac{\partial x}{\partial \theta_\alpha})_i^{s\,\mathrm{fid}}$ is therefore given by

$$\left(\frac{\partial x}{\partial \theta_\alpha}\right)_i^{s\,\mathrm{fid}} \approx \frac{x_i^{s\,\mathrm{fid}+} - x_i^{s\,\mathrm{fid}-}}{\Delta\theta_\alpha^+ - \Delta\theta_\alpha^-}. \tag{24}$$

Also we can calculate the $(\frac{\partial x}{\partial \theta_\alpha})_i^{s\,\mathrm{fid}}$ with the formula

$$\boldsymbol{\mu}_{f,\alpha} = \frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{k=1}^{n_d}\frac{\partial x_{ik}^{s\,\mathrm{fid}}}{\partial d_k}\frac{\partial d_{ik}^{s\,\mathrm{fid}}}{\partial \theta_\alpha}, \tag{25}$$

where i represents the random initialization of the simulation, and k represents the data point in the simulation.

Here, both the values of $\boldsymbol{\mu}_{f,\alpha}$ and $\mathbf{C}_f^{-1}$ are calculated with fixed, fiducial parameter values, $\boldsymbol{\theta}^{\mathrm{fid}}$. In IMNN, the function f is a neural network, which will be described in the next subsection.

B. Implementing f with artificial neural networks

A basic neuron unit is in the form:

$$a_j^l = \phi\left(\sum_j w_{ji}^l a_i^{l-1} + b_j^l\right). \tag{26}$$

The loss function in IMNN is defined using the Fisher information matrix $|\mathbf{F}|$:

$$\Lambda = -\frac{1}{2}|\mathbf{F}|^2 \tag{27}$$

or

$$\frac{\partial \Lambda}{\partial \mathbf{a}^L} = -|\mathbf{F}| + |\mathbf{C}_f|. \tag{28}$$

With the loss function, the weights and biases will be updated by gradient descent [36] in the updating procedure:

$$w_{ji}^l \to w_{ji}^l - \eta\frac{\partial \Lambda}{\partial w_{ji}^l} \tag{29}$$

and

$$b_i^l \to b_i^l - \eta\frac{\partial \Lambda}{\partial b_i^l}, \tag{30}$$

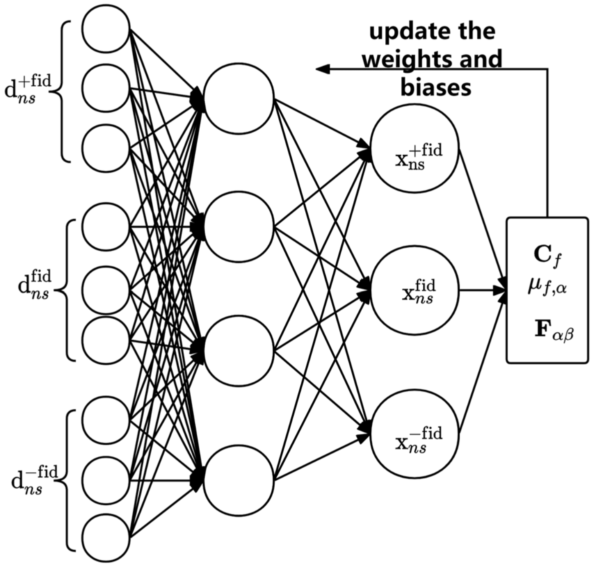
update the weights and biases
$\mathrm{d}_{ns}^{+\mathrm{fid}}$
$\mathrm{d}_{ns}^{\mathrm{fid}}$
$\mathrm{d}_{ns}^{-\mathrm{fid}}$
$\mathbf{x}_{\mathrm{ns}}^{+\mathrm{fid}}$
$\mathbf{x}_{ns}^{\mathrm{fid}}$
$\mathbf{x}_{ns}^{-\mathrm{fid}}$
$\mathbf{C}_f$
$\mu_{f,\alpha}$
$\mathbf{F}_{\alpha\beta}$

FIG. 6. The concise structure of the IMNN. The ANN can compress the input data **d** to the **x**. The loss function is calculated with $\mathbf{C}_f, \mu_{f,\alpha}$ and $\mathbf{F}_{\alpha\beta}$ with $\mathbf{x}^{+\mathrm{fid}}$. Normally **x** would be considered as the network output, but we can also choose the Fisher information matrix as the network output, which means that **x** will be the intermediate output of the neural network before calculating the loss function. In our work, obviously we wanted to obtain **x**.

where η is the learning rate, which controls the size of the steps in the procedure of updating the weights and biases [37]. The i means the ith element of the output vector of a collections of neurons in the $(l-1)$th layer, while the j means the jth neuron in the lth layer. The mean $\boldsymbol{\mu}_f$, covariance $\mathbf{C}^f$, which can be calculated with the Eqs. (20) and (21), are part of the loss function and therefore are functions of the weights and biases. The concise structure of the IMNN is shown in Fig. 6.

IV. CONSTRAINTS WITH REAL OHD

In this work, we use 3 types of methods (shown in Fig. 13) for ΛCDMs with and without curvature to constrain the cosmological parameters, which are: (1) using MAF and $\boldsymbol{H}_{\mathrm{obs}}$ to estimate the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ directly. (2) using the MAF-DAE to estimate the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ with $\boldsymbol{H}_{\mathrm{obs}}$. (3) using MAF-IMNN to estimate the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ with $\boldsymbol{H}_{\mathrm{obs}}$. We used the results from MAF as reference.

With $\boldsymbol{H}_{\mathrm{obs}}$ and the nonflat ΛCDM model, the posterior distribution estimated by MAF gives $H_0 = 68.34^{+1.06}_{-1.05}\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$, $\Omega_m = 0.30^{+0.14}_{-0.14}$, $\Omega_\Lambda = 0.65^{+0.17}_{-0.17}$, the posterior distribution estimated by MAF-IMNN gives $H_0 = 71.13^{+7.77}_{-7.87}\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$, $\Omega_m = 0.31^{+0.19}_{-0.19}$, $\Omega_\Lambda = 0.65^{+0.17}_{-0.17}$, the posterior distribution estimated by MAF-DAE

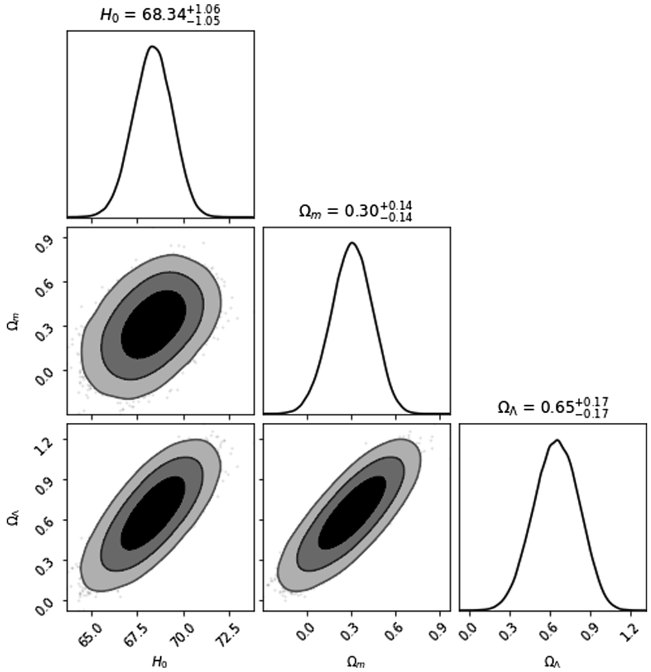

FIG. 7. The posterior distribution estimated by MAF, the real 31 OHD and nonflat ΛCDM model.

gives $H_0 = 66.17^{+2.45}_{-2.44}$ km s^{-1} Mpc^{-1}, $\Omega_m = 0.35^{+0.17}_{-0.18}$, $\Omega_\Lambda = 0.55^{+0.29}_{-0.29}$.

Meanwhile, with $\boldsymbol{H}_{\rm obs}$ and the flat ΛCDM model, the posterior distribution estimated by MAF gives $H_0 = 68.44^{+0.76}_{-0.75}$ km s^{-1} Mpc^{-1}, $\Omega_m = 0.31^{+0.03}_{-0.03}$, $\Omega_\Lambda = 0.69^{+0.03}_{-0.03}$, the posterior distribution estimated by MAF-IMNN gives H_0=$67.85^{+10.83}_{-10.82}$kms^{-1}Mpc^{-1}, Ω_m=$0.50^{+0.22}_{-0.22}$,

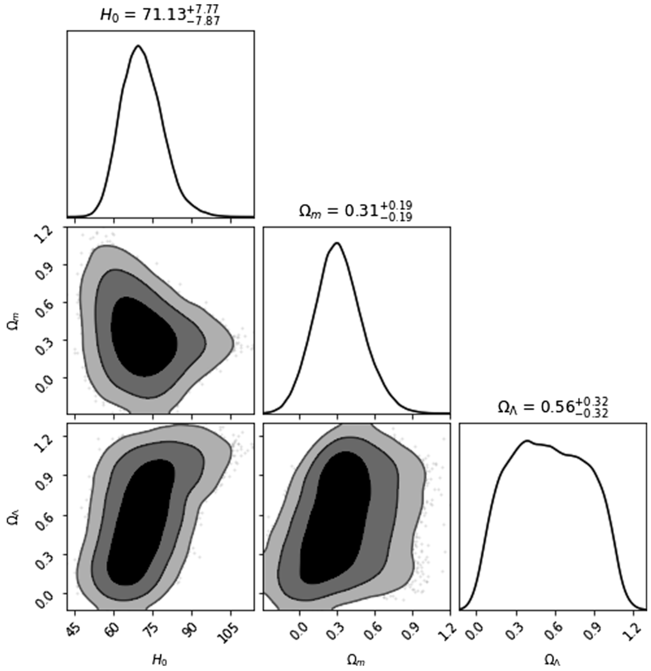

FIG. 8. The posterior distribution estimated by MAF-IMNN, the real OHD and nonflat ΛCDM model. The OHD is compressed into 10 dimensions.

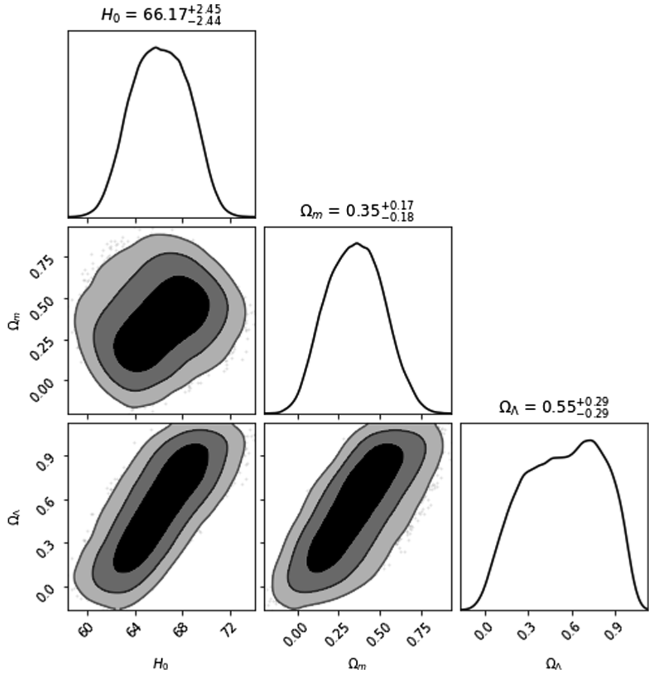

FIG. 9. The posterior distribution estimated by MAF-DAE, the real OHD and nonflat ΛCDM model. The OHD is compressed into 10 dimensions.

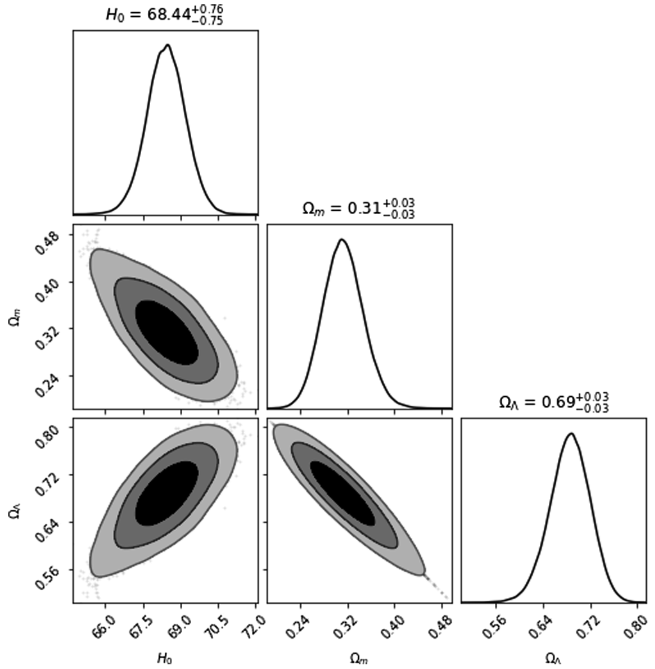

FIG. 10. The posterior distribution estimated by MAF, the real 31 OHD and flat ΛCDM model.

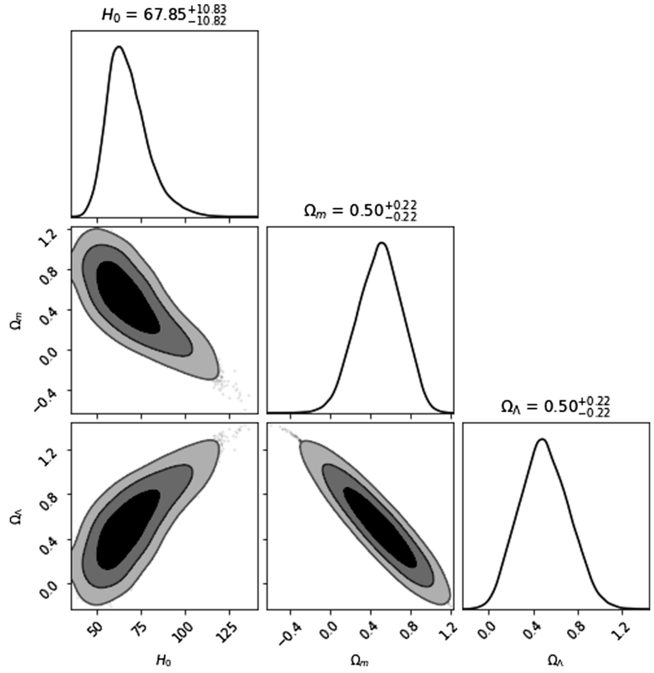

FIG. 11. The posterior distribution estimated by MAF-IMNN, the real OHD and flat ΛCDM model. The OHD is compressed into 10 dimensions.

$\Omega_\Lambda = 0.50^{+0.22}_{-0.22}$, the posterior distribution estimated by MAF-DAE gives $H_0 = 66.27^{+2.01}_{-2.00}\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$, $\Omega_m = 0.38^{+0.08}_{-0.09}$, $\Omega_\Lambda = 0.62^{+0.09}_{-0.08}$. We showed the table and figures of these posterior distributions in Figs. 7–12 and Table I.

TABLE I. The posterior distribution.

	H_0	Ω_m	Ω_λ
Nonflat ΛCDM			
MAF	$68.34^{+1.06}_{-1.05}$	$0.30^{+0.14}_{-0.14}$	$0.65^{+0.17}_{-0.17}$
MAF-IMNN	$71.13^{+7.77}_{-7.87}$	$0.31^{+0.19}_{-0.19}$	$0.65^{+0.17}_{-0.17}$
MAF-DAE	$66.17^{+2.45}_{-2.44}$	$0.35^{+0.17}_{-0.18}$	$0.55^{+0.29}_{-0.29}$
flat ΛCDM			
MAF	$68.44^{+0.76}_{-0.75}$	$0.31^{+0.03}_{-0.03}$	$0.69^{+0.03}_{-0.03}$
MAF-IMNN	$67.85^{+10.83}_{-10.82}$	$0.50^{+0.22}_{-0.22}$	$0.50^{+0.22}_{-0.22}$
MAF-DAE	$66.27^{+2.01}_{-2.00}$	$0.38^{+0.08}_{-0.09}$	$0.62^{+0.09}_{-0.08}$

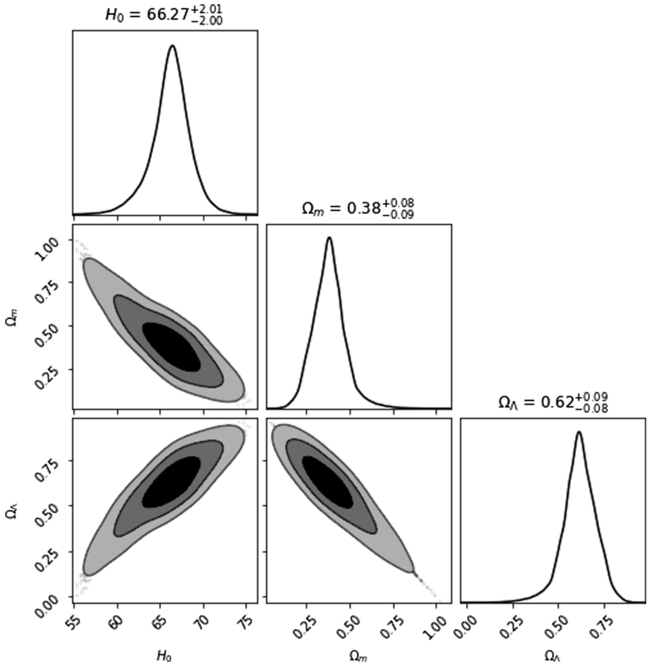

FIG. 12. The posterior distribution estimated by MAF-DAE, the real OHD and flat ΛCDM model. The OHD is compressed into 10 dimensions.

V. THE COMPARISON OF IMNN AND DAE

`To avoid computationally expensive calculation in comparing posterior directly, we apply some criteria, which can be calculated by posterior distributions. We try to train both DAE and IMNN to compress the 31 dimensional $\boldsymbol{H}_{\mathrm{obs}}$ into different dimensions and estimate the posterior $P(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ in different learning rates, so that we can compare the results under different learning rates and dimensionality reduction processes. In addition, we take the posterior $P_1(\boldsymbol{\theta}|\boldsymbol{H}_{\mathrm{obs}})$ obtained from only MAF as the standard posterior in an effort to investigate the impact of the addition of DAE and IMNN on the standard posterior. In the following subsection, we introduce the criteria we apply in this work and do the comparison of DAE and IMNN.

A. Comparison criteria

In this paper, we apply two criteria, KL divergence and figure of merit(FoM).

Kullback-Leibler divergence (KL divergence). Kullback-Leibler divergence is a statistical distance which can measure how one probability distribution is different from a second one. That is, Kullback-Leibler divergence can be used to calculate how much information is lost when we approximate one distribution with another. Generally, while processing probability and statistics, we can replace the observed data or complex distribution with a simpler approximate distribution. Suppose that there are two probability density distributions red $p_1(\boldsymbol{\theta})$ and $p_2(\boldsymbol{\theta})$, where $p_2(\boldsymbol{\theta})$ is the simulation of the $p_1(\boldsymbol{\theta})$. Then we can use the KL divergence to calculate the information loss of approximating $p_1(\boldsymbol{\theta})$ using $p_2(\boldsymbol{\theta})$. In this case, The KL divergence from $p_1(\boldsymbol{\theta})$ to $p_2(\boldsymbol{\theta})$ is defined as

$$D_{\mathrm{KL}} = (p_1(\boldsymbol{\theta})||p_2(\boldsymbol{\theta})) = \mathbb{E}_{p_1(\boldsymbol{\theta})}(\log p_1(\boldsymbol{\theta}) - \log p_2(\boldsymbol{\theta})). \tag{31}$$

In this paper, we sample M samples $\{\boldsymbol{\theta}_i\}$ from the posterior, so the KL divergence is estimated with:

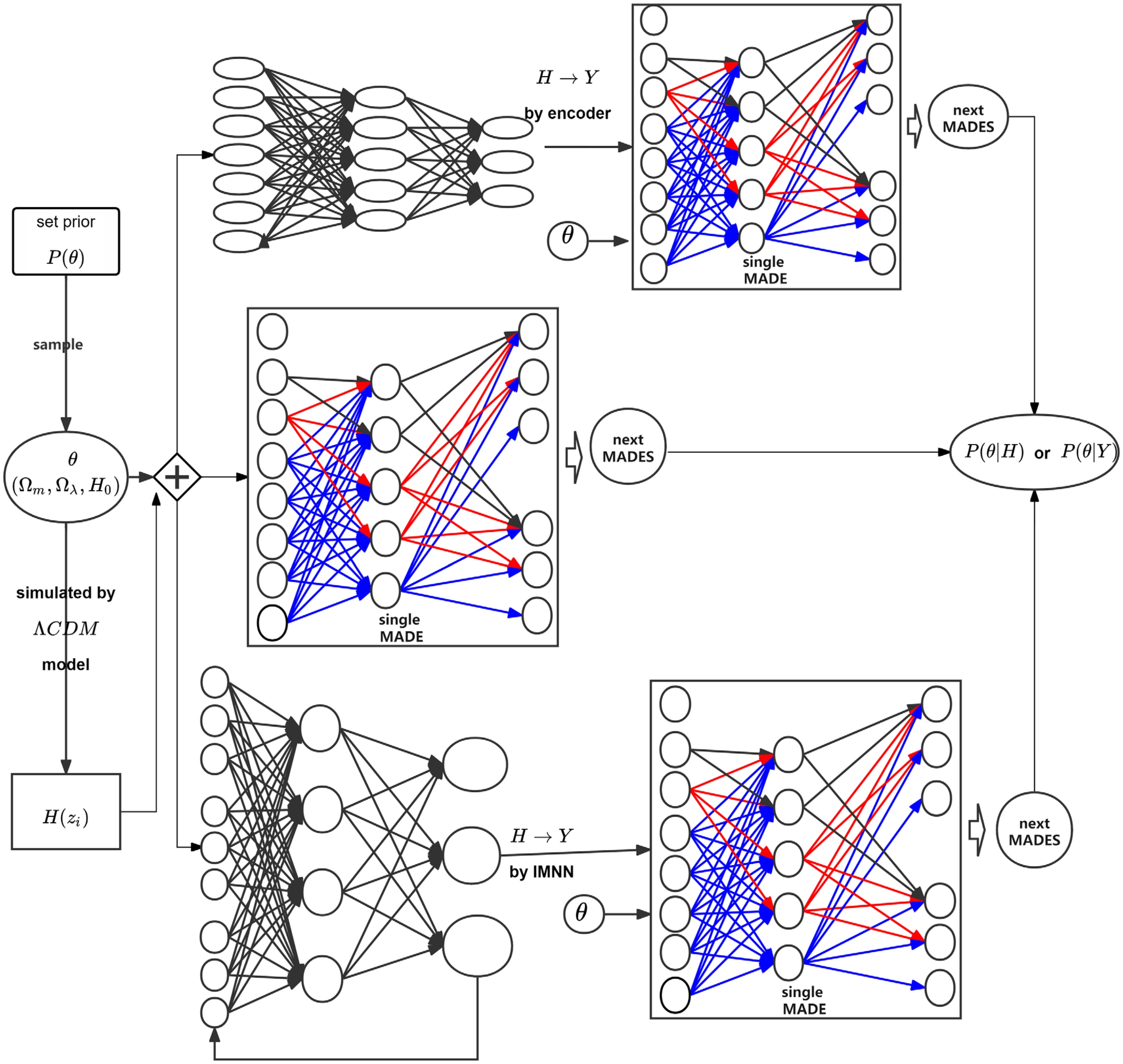

FIG. 13. The procedure in this work. We first produce training data. The upper part is MAF-DAE. The training data is compressed into low-dimensional $\boldsymbol{y}$ by IMNN, then $\boldsymbol{y}$ and the corresponding $\boldsymbol{\theta}$ are transmitted to the MAF. The middle part is MAF, which is trained with original-dimensional training data. The lower part is MAF-IMNN. In this method, the training data is compressed into low-dimensional $\boldsymbol{y}$ by DAE. Then $\boldsymbol{y}$ and the corresponding $\boldsymbol{\theta}$ are transmitted to the MAF. After training, when evaluating cosmological parameters, MAFs can give posterior $P(\boldsymbol{\theta}|\boldsymbol{H}_{\rm obs})$ and $P(\boldsymbol{\theta}|\boldsymbol{y}_{\rm obs})$ [or $P(\boldsymbol{\theta}|\boldsymbol{H}_{\rm moc})$ and $P(\boldsymbol{\theta}|\boldsymbol{y}_{\rm moc})$].

$$D_{\rm KL}(p_1||p_2) = \frac{1}{M}\sum_{i=1}^{M}(\ln P_1(\boldsymbol{\theta}_i|\boldsymbol{H}_{\rm obs}) - \ln P_2(\boldsymbol{\theta}_i|\boldsymbol{H}_{\rm obs})), \tag{32}$$

where $P_2(\boldsymbol{\theta}|\boldsymbol{H}_{\rm obs})$ is the posterior calculated from MAF-DAE or MAF-IMNN and $P_1(\boldsymbol{\theta}|\boldsymbol{H}_{\rm obs})$ is the posterior calculated with only MAF. From Eq. (32), it is obvious that the smaller the KL divergence, the closer the $P_1(\boldsymbol{\theta}|\boldsymbol{H}_{\rm moc})$ and $P_2(\boldsymbol{\theta}|\boldsymbol{H}_{\rm moc})$. When $D_{\rm KL}(p_1||p_2) = 0$, it means that the two posterior are almost identical.

Figure of merit (FoM). When constraining the parameters, we want to get an accurate range of the parameters and tighten the constraints. The FoM used in this work is similar to the one adopted by [38,39] in their work. The FoM is defined as:

$$P(\boldsymbol{\theta}|\boldsymbol{H}_{\rm obs}) = {\rm const.} = \exp(-\Delta\mathcal{X}^2/2)P_{\rm max}, \tag{33}$$

where $P_{\max}$ is the maximum probability density of the posterior, and $\exp(-\Delta\mathcal{X}^2/2)$ is a constant which ensures that $\exp(-\Delta\mathcal{X}^2/2)P_{\max}$ is equal to the probability density at the boundary of the 95.44% confidence region of the Gaussian distribution. According to [12], $\exp(-\Delta\mathcal{X}^2/2)$ here takes the same value of 8.02. The FoM represents the reciprocal volume of the confidence region of the posterior, so the larger the FoM, the tighter the constraint of the parameters are.

B. Experiments and results

1. Comparison using KL divergence

We show the different FoM in Figs. 14–16. The results show signs that DAE could make a better performance than IMNN. Besides, MAF-IMNN and MAF-DAE have better results in the nonflat ΛCDM, as is showed in Fig. 15, the KL divergence increases with the dimension reduction.

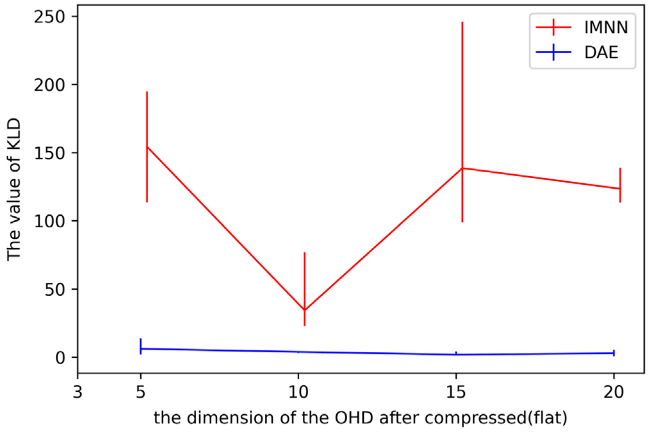

FIG. 14. KL divergence calculated by MAF-IMNN or MAF-DAE in the flat ΛCDM model.

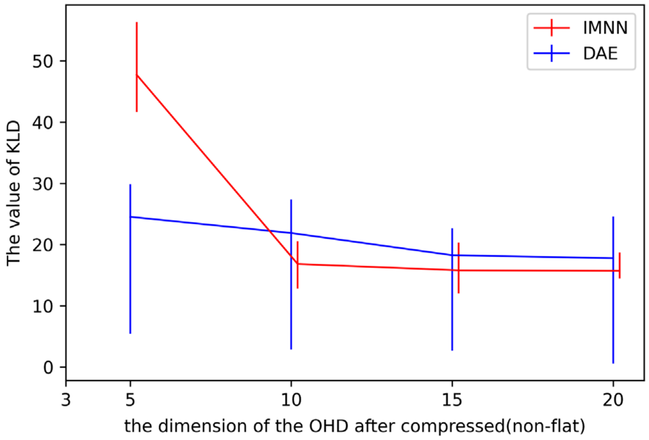

FIG. 15. KL divergence calculated by MAF-IMNN or MAF-DAE in the nonflat ΛCDM model.

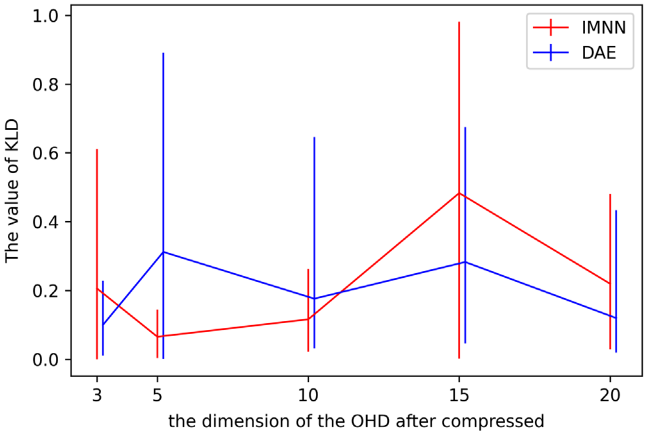

FIG. 16. KL divergence calculated by MAF-IMNN or MAF-DAE with low uncertainty.

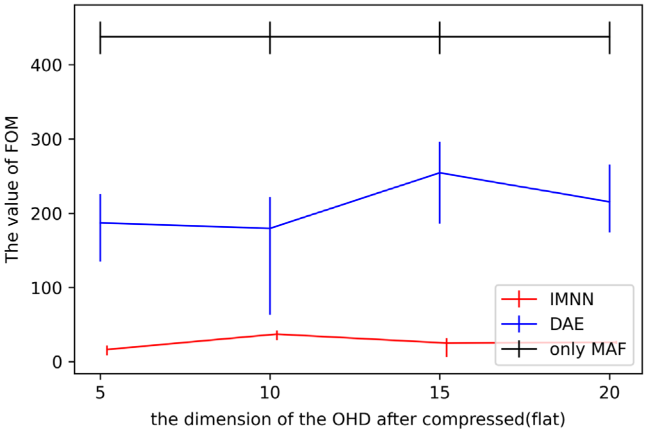

FIG. 17. FoM calculated by MAF-IMNN and MAF-DAE in the flat ΛCDM model. The black line is the FoM of the posterior calculated by only MAF with the uncompressed mock OHD in different learning rates.

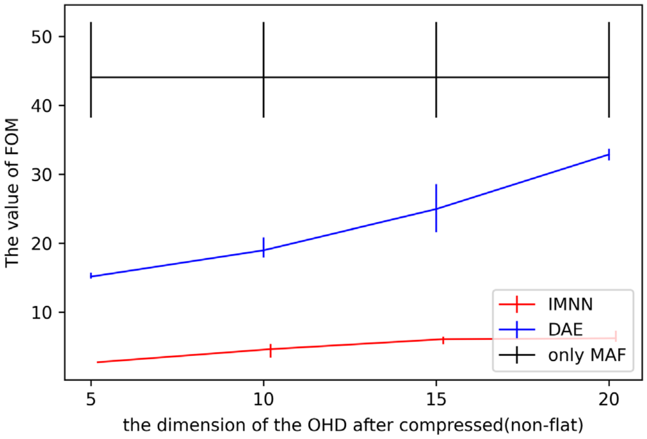

FIG. 18. FoM calculated by MAF-IMNN and MAF-DAE in the nonflat ΛCDM model. The black line is the FoM of the posterior calculated by only MAF with the uncompressed mock OHD in different learning rates.

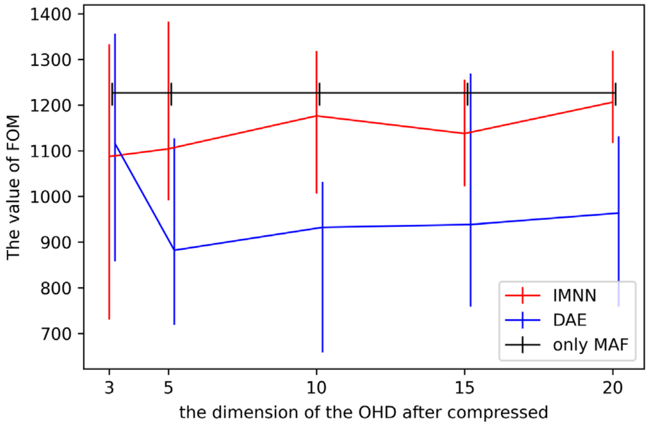

FIG. 19. FoM calculated by MAF-IMNN and MAF-DAE in the low uncertainty. The black line is the FoM of the posterior calculated by only MAF with the uncompressed mock OHD in different learning rates.

2. Comparison using FoM

We show the different FoM in Figs. 17–19. We can see that the FoM calculated from the posterior from MAF-DAE is generally larger, meaning that the data processed by DAE can give a tighter posterior. While using the small error training data, MAF-IMNN gives a bit smaller distributions.

VI. CONCLUSIONS AND DISCUSSION

In this paper, we validate the feasibility of MAF-IMNN, and compare IMNN and DAE in the procedure of constraining the cosmological parameters. Since we have already demonstrated that the confidence regions estimated with MAF are very close to those of MCMC [12], and the purpose of this work is to compare IMNN and DAE, we therefore used the results of MAF as the standard and did not calculate the KL divergence and FoM of the MCMC results.

We also used different model to simulate the training data to do a comprehensive comparison between DAE and IMNN. With the small error training data, The performance of those two methods is very similar. With the normal training data, the overall performance of DAE is better than that of IMNN. Nevertheless, there is always an apparent influence from IMNN or DAE, no matter which kind of training data was used. We can also estimate another cosmological model as long as we generate the training data according to the cosmological model.

Admittedly, our work is not perfect in some aspects. First, the simulation model in this paper is not complex enough to simulate the generation process and uncertainty, though we used Gaussian sample in this work and Gaussian process in our previous work [12] to generate training set. Because the main task in this work is to compare DAE and IMNN, we did not focus on the simulation model, but our next ongoing work is to build a better model to simulate OHD with deep learning. Second, there are other types of autoencoders, such as the denoising variational autoencoder (the combination of variational autoencoder [40] and denoising autoencoder). We chose DAE in this work and our previous work [12] because it can not only learn the robust features but also significantly reduce the noise level. However, it is hard to tell if DAE is the best choice without experiments, so one of our future works is to use the method in this work to compare DAE with other autoencoders.

In the future, we will probably be able to do a better constraint if we can extend our dataset. However, we do not recommend mixing datasets, because it means mixing different errors which are calculated by different methods, we will not necessarily obtain an accurate estimation.

ACKNOWLEDGMENTS

We thank the anonymous referee for the comments that helped us greatly improve this paper. The referee reviewed our paper carefully and put forward many useful suggestions. We thank Changzhi Lu, Jin Qin, Jing Niu, Kang Jiao and Tom Charnock for useful discussions and their kind help. This work was supported by the National Science Foundation of China (Grants No. 61802428, No. 11929301), and National Key R&D Program of China (2017YFA0402600).

[1] J. F. Jesus, T. Gregório, F. Andrade-Oliveira, R. Valentim, and C. Matos, Mon. Not. R. Astron. Soc. **477**, 3 (2017).

[2] D. M. Scolnic, D. O. Jones, A. Rest, Y. C. Pan, R. Chornock, R. J. Foley, M. E. Huber, R. Kessler, G. Narayan, and A. G. Riess, Astrophys. J. **859**, 101 (2017).

[3] C. Chia-Hsun and Y. Wang, Mon. Not. R. Astron. Soc. 226 (2012).

[4] S. Pan, M. Liu, J. Forero-Romero, C. G. Sabiu, Z. Li, H. Miao, and X.-D. Li, Sci. China Phys. Mech. Astron. **63**, 1 (2020).

[5] E. Cameron and A. N. Pettitt, Mon. Not. R. Astron. Soc. **425**, 44 (2012).

[6] A. Weyant, C. Schafer, and W. M. Wood-Vasey, Astrophys. J. **764**, 116 (2013).

[7] G. Papamakarios and I. Murray, in *Proceedings of the 30th Conference on Neural Information Processing Systems* (2016), arXiv:1605.06376.

[8] M. Reza, Y. Zhang, B. Nord, J. Poh, A. Ciprijanovic, and L. Strigari (2022), arXiv:2208.00134.

[9] L. A. Perez, S. Genel, F. Villaescusa-Navarro, R. S. Somerville, A. Gabrielpillai, D. Anglés-Alcázar, B. D. Wandelt, and L. Yung (2022), arXiv:2201.04142.

[10] H. J. Hortua, R. Volpi, D. Marinelli, and L. Malagò, Phys. Rev. D **102**, 103509 (2020).

[11] S. Hassan, S. Andrianomena, and C. Doughty, Mon. Not. R. Astron. Soc. **494**, 5761 (2020).

[12] Y. C. Wang, Y. B. Xie, T. J. Zhang, H. C. Huang, T. Zhang, and K. Liu, Astrophys. J. Suppl. Ser. **254**, 16 (2021).

[13] G. Papamakarios, T. Pavlakou, and I. Murray, arXiv:1705.07057.

[14] G. Papamakarios, D. C. Sterratt, and I. Murray, Sequential neural likelihood: Fast likelihood-free inference with autoregessive flows, arXiv:1805.07226.

[15] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, in *Proceedings of the 25th International Conference on Machine Learning, ICML '08* (Association for Computing Machinery, New York, NY, 2008), pp. 1096–1103.

[16] T. Charnock, G. Lavaux, and B. D. Wandelt, Phys. Rev. D **97**, 083004 (2018).

[17] J. Alsing, T. Charnock, S. Feeney, and B. Wandelt, Mon. Not. R. Astron. Soc. **488**, 4440 (2019).

[18] M. Germain, K. Gregor, I. Murray, and H. Larochelle, https://JMLR.org (2015).

[19] D. J. Rezende and S. Mohamed, Comput. Theor. Polym. Sci. 1530 (2015), arXiv:1505.05770.

[20] R. Jimenez, L. Verde, T. Treu, and D. Stern, Astrophys. J. **593**, 622 (2003).

[21] J. Simon, L. Verde, and R. Jimenez, Phys. Rev. D **71**, 123001 (2005).

[22] D. Stern, R. Jimenez, L. Verde, M. Kamionkowski, and S. A. Stanford, J. Cosmol. Astropart. Phys. 02 (2009) 008.

[23] M. Moresco, L. Verde, L. Pozzetti, R. Jimenez, and A. Cimatti, J. Cosmol. Astropart. Phys. 07 (2012) 053.

[24] Z. Cong, Z. Han, Y. Shuo, L. Siqi, Z. Tong-Jie, S. Yan-Chun *et al.*, Res. Astron. Astrophys. **14** 1211 (2014).

[25] M. Moresco, L. Pozzetti, A. Cimatti, R. Jimenez, C. Maraston, L. Verde, D. Thomas, A. Citro, R. Tojeiro, and D. Wilkinson, J. Cosmol. Astropart. Phys. 05 (2016) 014.

[26] A. L. Ratsimbazafy, S. I. Loubser, S. M. Crawford, C. M. Cress, B. A. Bassett, R. C. Nichol, and P. Väisänen, Mon. Not. R. Astron. Soc. 3 (2017).

[27] H.-R. Yu, S. Yuan, and T.-J. Zhang, Phys. Rev. D **88**, 103528 (2013).

[28] C. Zhang, H. Zhang, S. Yuan, S. Liu, T. J. Zhang, Y. C. Sun (D. O. Astronomy and B. N. University Collaboration), Res. Astron. Astrophys. (2014).

[29] R. Fisher, *Statistical Methods for Research Workers. Biological monographs and manuals* (The University of California, USA, 1925).

[30] M. G. Kendall, Technometrics **5**, 525 (1963).

[31] J. F. Kenney and E. S. Keeping, *Mathematics of statistics. Number Part II* (Van Nostrand, New York, USA, 1951).

[32] E. Lehmann and G. Casella, *Theory of Point Estimation. Springer Texts in Statistics* (Springer New York, 2003).

[33] H. Cramér, *Mathematical Methods of Statistics* (Princeton University Press, Princeton, USA, 1946).

[34] C. R. Rao, *Information and the Accuracy Attainable in the Estimation of Statistical Parameters* (Springer New York, 1945).

[35] A. F. Heavens, R. Jimenez, and O. Lahav, Mon. Not. R. Astron. Soc. **317**, 965 (2000).

[36] K. C. Kiwiel, Math. Program. **90**, 1 (2001).

[37] S. Theodoridis, Mach. Learn. 875 (2015).

[38] C. Ma and T.-J. Zhang, Astrophys. J. **730**, 74 (2011).

[39] H. Wang and T. J. Zhang, Astrophys. J. **748**, 315 (2011).

[40] D. P. Kingma and M. Welling (2014), arXiv:1312.6114.