Seeking new physics in cosmology with Bayesian neural networks: Dark energy and modified gravity

M. Mancarella, J. Kennedy, B. Bose, and L. Lombriser
Département de Physique Théorique, Université de Genève, 24 quai Ernest Ansermet, 1211 Genève 4, Switzerland

(Received 28 July 2021; accepted 11 January 2022; published 28 January 2022)

We study the potential of Bayesian neural networks (BNNs) to detect new physics in the dark matter power spectrum, concentrating here on evolving dark energy and modifications to general relativity. After introducing a new technique to quantify classification uncertainty in BNNs, we train two BNNs on mock matter power spectra produced using the publicly available code ReACT in the k range $(0.01 - 2.5)$ $h\mathrm{Mpc}^{-1}$ and redshift bins (0.1, 0.478, 0.783, 1.5) with Euclid-like noise. The first network classifies spectra into five labels including ΛCDM, $f(R)$, wCDM, Dvali-Gabadadze-Porrati gravity and a "random" class, whereas the second is trained to distinguish ΛCDM from non-ΛCDM. Both networks achieve a comparable training, validation and test accuracy of ∼95%. Each network is also capable of correctly classifying spectra with deviations from ΛCDM that were not included in the training set, demonstrated with spectra generated using the growth index γ. To obtain an indication of the BNNs classification capability, we compute the smallest deviation from ΛCDM such that the noise-averaged non-ΛCDM classification probability is at least 95% according to our estimated error quantification, finding these bounds to be $f_{R0} \lesssim 10^{-7}$, $\Omega_{\mathrm{rc}} \lesssim 10^{-2}$, $-1.05 \lesssim w_0 \lesssim 0.95$, $-0.2 \lesssim w_a \lesssim 0.2$, and $0.52 \lesssim \gamma \lesssim 0.59$. The bounds on $f(R)$ can be improved by training a specialist network to distinguish solely between ΛCDM and $f(R)$ power spectra which can detect a nonzero f_{R0} at $\mathcal{O}(10^{-8})$. We expect that further developments, such as the inclusion of smaller length scales or additional extensions to ΛCDM, will only improve the potential of BNNs to indicate the presence of new physics in cosmological datasets, regardless of the underlying theory.

DOI: 10.1103/PhysRevD.105.023531

I. INTRODUCTION

The scientific method is based upon the meticulous comparison of theoretical hypotheses with observations. A hypothesis can be promoted to a foundational theory once it has rigorously satisfied a multitude of observational tests. Such is the case with the concordance cosmological model ΛCDM, named after the two dominant components that contribute to the current energy density of the Universe: the cosmological constant (Λ) and cold dark matter (CDM). Despite their dominant contribution to the stress-energy density of the Universe, the fundamental nature of both dark matter and dark energy remains a mystery. Determining the physical nature of these two components is a central challenge in modern physics. Combined with the task of furthering our understanding of dark energy and dark matter is the requirement to test Einstein's theory of general relativity (GR) in the hitherto unexplored cosmological regime. Potential modifications to gravitational dynamics at cosmological length scales may also play a part in providing an explanation for dark energy and dark matter. A principal pursuit of contemporary cosmology is therefore to stringently compare both GR and ΛCDM against a considerable collection of alternative models [1–7].

The large-scale structure (LSS) of the Universe provides an ideal testing ground for competing hypotheses. Assuming that CDM can be treated as a perfect fluid, it undergoes gravitational collapse into localized overdensities, generating gravitational wells which the galaxy distribution subsequently traces. By correlating galaxy positions over a large volume, a statistical description of how the underlying dark matter clusters can be obtained. This is largely characterized by the two-point correlation function or the power spectrum in Fourier space. Although the distribution of dark matter is not directly observable, modern cosmological surveys use observables such as galaxy clustering [8] or weak lensing [9] to probe the underlying dark matter distribution. The next generation of galaxy surveys such as Euclid [10] and Legacy Survey of Space and Time (LSST) [11] have the capability to measure the cosmological galaxy distribution with extremely high precision, especially at length scales where the cosmological background becomes subdominant to baryonic and nonlinear gravitational physics. Analytic methods are impractical in this regime as the evolution equations do not possess closed-form solutions. Cosmological N-body simulations can provide highly accurate numerical predictions, yet their computational cost renders them unsuitable

for constraining model parameters in Markov chain Monte Carlo (MCMC) analyses. Motivated by this issue, Refs. [12–14] constructed emulators and nonlinear models for the matter power spectrum, with extensions to include deviations from ΛCDM developed in Refs. [15–17]. These are fast and accurate methods which compute predictions for the shape of the matter power spectrum but are limited to an underlying hypothesis. Recently, Ref. [18] provided a method to predict the shape of the nonlinear matter power spectrum for a wide range of models which was subsequently implemented into a code called ReACT in Ref. [19]. Using this framework it is possible to generate a large dataset of mock matter power spectra for a broader class of extensions to ΛCDM with varying values of the model parameters.

Such tools enable one to extract information from a large range of length scales, substantially improving the constraining power. MCMC analyses are frequently employed to determine whether physics beyond ΛCDM is present in cosmological data. To consistently constrain beyond ΛCDM physics in such analyses, one must choose a finite set of parameters quantifying the new physics. It turns out that the number of parameters needed to do this including nonlinear scales while remaining agnostic to the underlying fundamental physics is immense (see Ref. [20] for example), making such an analysis currently unfeasible. Consequently, current analyses either restrict themselves to the linear regime of structure formation or perform a model-by-model analysis. It is worth noting that even if computational expense was not an issue, the simple inclusion of such a large additional parameter space would strongly penalize the extended modeling on the basis of the Bayesian evidence. It is therefore of interest to examine alternative approaches which do not rely on picking an effective set of parameters and are less computationally expensive.

With the ability to produce a large dataset consisting of power spectra for a variety of models, it is natural to consider the capability of deep neural networks (DNNs) to classify power spectra according to their underlying cosmological model. However, the prediction given by a trained DNN can be subject to several sources of uncertainty. Adding a slight perturbation to the input, passing the input to a network with a different architecture or training on a separate subset of the training set could all drastically alter the result [21–23]. Taking these issues into account is therefore crucial to obtaining statistically robust predictions with neural networks. Quantifying the potential variability of the prediction, and in turn the confidence, is extremely difficult with DNNs.

Bayesian neural networks (BNNs) try to model the uncertainty by replacing each weight in the network by a distribution initialized to a prior [23–28]. Rather than obtaining the same pointlike output for an example with every pass through the network, the BNN's prediction varies as each pass draws a different sample from the weight distribution. By repeatedly passing an example to the BNN, one obtains a distribution of predictions conditioned on the training data, the network architecture, and the noise in the data along with other potentially unknown sources of uncertainty. A quantitative estimate of the classification uncertainty can be obtained in minutes once the BNN has been trained. As an additional advantage, BNNs naturally provide a regularization procedure for preventing overfitting [29]. BNNs have recently been applied in many fields such as gravitational waves [30–32], the cosmic microwave background [33,34], autonomous driving [35], cellular image classification [36] and the detection and classification of supernovae [37,38].

In this paper we explore the potential of BNNs to classify non-ΛCDM models from the matter power spectrum. In particular, BNNs can be trained on as many deviations from ΛCDM as can be implemented in numerical codes such as ReACT. Even if none of these theories turn out to be the correct model, they are all representative of possible sources of new physics. We will therefore investigate whether BNNs can identify general deviations from concordance cosmology based on the observational features of known models.

The goal is, at the very least, to develop a promising tool to inform standard and more rigorous MCMC analyses by providing a refinement of the theoretical parameter space that needs to be explored. On the other hand, the possibility of constructing a well-defined probability distribution which accounts for all sources of uncertainty in the prediction from a BNN is an open research question [39–48]. Should this become possible, this method could be promoted to a statistical tool competitive to MCMC. Regardless of this possibility, DNNs can simply be used as a tool to compress the information from the power spectrum in a small set of numbers, that can be in turn combined with other machine-learning (ML)-based methods in rigorous statistical frameworks (such as approximate Bayesian computation [49] or likelihood-free inference [50]) to perform model selection in a fully ML-based fashion. In any case, it it worthwhile to assess their potential, and this work is a first step in this direction.

In Fig. 1 we show a schematic representation of the method. Using ReACT to generate a training set of thousands of example matter power spectra for both ΛCDM and selected extensions, we train two BNNs to classify the spectra according to the underlying model. A five-label BNN is trained to classify an example spectrum as either ΛCDM or one of four chosen extensions, while a two-label network is trained simply to classify between ΛCDM and non-ΛCDM. Following the introduction of a novel method to construct a well-defined probability distribution from the output of a BNN in order to take into account the effect of the uncertainty in the final classification (thus preventing the network from being overconfident), we evaluate the

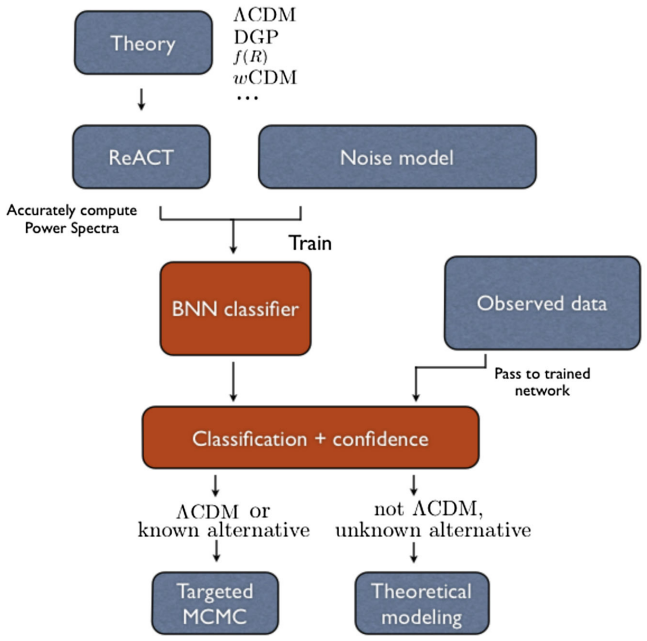

FIG. 1. Representation of the work flow presented in the paper to study the presence of deviations from ΛCDM in the matter power spectrum. The key elements of the method, namely the use of a Bayesian neural network and a novel way to quantify the confidence, are colored in red.

performance of each network on the training, validation and test sets.

In addition, we determine the minimal deviation in the model parameters for each chosen non-ΛCDM model such that the five-label BNN classifies them as non-ΛCDM with some specified probability before passing the same spectra through the two-label BNN to compare their performance. After studying how effective each BNN is at recognizing spectra which do not belong to any class in the training set, we compare their predicted classification probabilities averaged over noise realizations for different values of the model parameters. Finally, we examine the potential benefits of training specialist networks on selected subsets of the original classes in the training set. While we only consider three well-studied dark energy and modified gravity models in this work, this method can be extended to general extensions to ΛCDM such as massive neutrinos and Horndeski scalar-tensor theory as long as rapid and accurate predictions for the shape of the matter power spectrum can be computed. The effect of neutrino masses and baryonic feedback can now be taken into account and will be the subject of future work, given that it has been recently implemented in ReACT [51].

This paper is laid out as follows. Section II presents a concise theoretical background to both DNNs and BNNs. Section III then describes the generation and preparation of the training, validation and test data before they can be passed to the BNN, followed by a discussion of the BNN's architecture. Section IV discusses the overall performance of each BNN. We determine the values of the model parameters in each non-ΛCDM model such that the five- and two-label BNNs are confident that a spectrum deviates from ΛCDM. After analyzing how sensitive each classification was to the noise in the power spectrum we also discuss the notion of specialist networks. Section VI lays out potential avenues that should be explored in future studies before we conclude in Sec. VII.

II. BAYESIAN NEURAL NETWORKS

Neural networks are becoming ever more widely employed in physics. The interested reader can find a review of the core concepts surrounding the use of neural networks in supervised classification problems in Appendix A, to which we refer for the basic concepts used in the rest of the paper. For a thorough treatment see Refs. [52–54]. In this section we discuss the aspects of BNNs relevant to this work, in particular the quantification of the classification uncertainty.

A. Uncertainty in BNN classifiers

Two principal sources of uncertainty can be identified in the prediction given by a trained network for a new example, namely *aleatoric* and *epistemic* uncertainty [55]. The former encompasses any intrinsic, nonreducible, uncertainty due to the stochasticity of the observations, while the latter describes uncertainty in the model. High epistemic uncertainty quantifies limitations in modeling accuracy and could be reduced by, for example, choosing a more appropriate model architecture, adjusting the hyperparameters, or training with more data.

Differently from traditional NNs, BNNs can give an estimate of both sources of uncertainty. Technical details, including a discussion on the difference with traditional NNs, are given in Appendix B. The key concept is the replacement of the networks' weights w with distributions, so that an approximate posterior distribution of the weights, $q_\theta(w)$, can be learned by variational inference, instead of learning a single value for each weight as in traditional NNs. This means that after training the learned distribution can be used to obtain predictions marginalized over the weights, rather than pointlike prediction. This in turn allows one to take into account potential variability in the network's output and the relative uncertainty as we will now show.

In the case of a classification problem with N classes, the final layer of the network outputs an N-dimensional vector with components that sum to one and can therefore be interpreted as probabilities. We denote these components by $p(y_i^\star = 1|X^\star, w, \mathcal{D})$ $(i \in \{1, \ldots, N\})$ for a new example with features $X^\star$ and one-hot encoded label $y^\star$, for a given realization of the weights w and conditioned on the training data $\mathcal{D}$. Marginalization over the weights can be obtained

via Monte Carlo sampling from $q_\theta(w)$, giving for each component of the one-hot encoded label vector

$$\mu_i \equiv p(y_i^\star = 1|X^\star, \mathcal{D}) \approx \frac{1}{N_S}\sum_{\alpha=1}^{N_S} p_\alpha,$$
$$p_\alpha \equiv p(y_i^\star = 1|X^\star, w_\alpha, \mathcal{D}), \qquad w_\alpha \sim q_\theta(w|\mathcal{D}), \quad (1)$$

where N_S is the number of samples and throughout this paper we use Greek indices to denote MC samples and Latin indices to denote vector components. Equation (1) is the Monte Carlo approximation of the exact expression in Eq. (B7). A prediction for the label for a new example with features $X^\star$ is obtained by assigning the label to the maximum output probability $y^\star_{\text{pred}} = \arg\max_i \mu_i \ (i = 1 \ldots N)$, if this exceeds a chosen threshold probability p_{th}.

Defining μ to be the vector with components μ_i from Eq. (1), the full covariance of the classification is given by [56]

$$\begin{aligned}\Sigma_{q_\theta} &= \mathbb{E}_{q_\theta}[\mathrm{Cov}_{p(y_i^\star=1|X^\star,w,\mathcal{D})}(y^\star)] \\ &\quad + \mathrm{Cov}_{q_\theta}[\mathbb{E}_{p(y_i^\star=1|X^\star,w,\mathcal{D})}(y^\star)] \\ &= \frac{1}{N_S}\sum_{\alpha=1}^{N_S}(\mathrm{diag}(p_\alpha) - p_\alpha^{\otimes 2}) + \frac{1}{N_S}\sum_{\alpha=1}^{N_S}(p_\alpha - \mu)^{\otimes 2} \\ &= \mathrm{diag}(\mu) - \mu^{\otimes 2}, \end{aligned} \quad (2)$$

where the first line follows from the definition of the covariance and the second from the use of Eq. (1) with the following property of a multinomial distribution (which is used as the likelihood of the optimisation problem as customary in classification tasks; see Appendix B): $\mathbb{E}_{p(y_i^\star=1|X^\star,w,\mathcal{D})}(y^\star) = p(y_i^\star = 1|X^\star, w, \mathcal{D})$. This shows that the covariance is simply the standard multinomial covariance over the distribution of MC averages μ. The second term in the sum is the standard mean-squared error coming from the fact that the weights have a distribution $q_\theta(w|\mathcal{D})$; hence, it corresponds to the epistemic uncertainty. The first term encodes the contribution to the variance marginalizing over $q_\theta(w|\mathcal{D})$, and as such it describes the aleatoric uncertainty. In order not to yield overconfident estimates of whether a given power spectrum is classified as ΛCDM or not it is important to accommodate both sources of uncertainty into the analysis. When training on data coming from real-world observations, one has no means to reduce the aleatoric uncertainty (this is why this is sometimes referred to as "uncertainty in the data"). In this paper we train a network on simulated noisy data, as described in Sec. III A. In principle, the knowledge of the model from which the noise is drawn could be incorporated in the loss. Here we rather make the choice of treating noise as an effective aleatoric uncertainty and including its effect in the classification uncertainty. Of course, a dependence on the noise model will be inherited during training. We note however that any data analysis tool relies on a model of the noise.

In order to compute the uncertainty, it must be kept in mind that despite μ_i being a probability by construction, it still does not represent an inferred "true probability" for the resultant classification as occurs in a likelihood or MCMC analysis. The quantity μ_i should rather be interpreted as a parameter in itself used to classify a given spectrum if the magnitude exceeds the chosen threshold probability p_{th}. Constructing a confidence in the classification at test time requires a joint distribution on μ_i to compute the subvolume where $\mu_i > p_{\text{th}}$. We shall detail in the following subsection how we utilize the uncertainty in Eq. (2) to estimate the confidence in a particular classification. We stress here that while it is tempting to view BNNs as being able to provide a clear and statistically rigorous definition of probability in the classification, we should keep in mind that the model of the error is still subject to approximations, such as the variational approach described in this section and the choice of the parametric distribution $q_\theta(w)$. For these reasons, it is also important to point out that the use of the definition "Bayesian" neural networks in the formulation used in this work comes from the machine-learning literature, and the estimated classification probabilities should not be confused with the result of a truly Bayesian model selection as resulting, for example, from the computation of Bayesian evidences with a nested sampling algorithm. Rather, at the current state of the art BNNs should be viewed as tools that at least enable one to introduce a model of the uncertainty, preventing overly optimistic interpretations of the results as well as providing an effective regularization procedure.

B. Quantifying the classification confidence

Currently there is no well-established method of quantifying the confidence in a prediction from a BNN. In general, obtaining a classification confidence requires the definition of a probability distribution over the softmax output of the network. One possibility is the Dirichlet distribution which is both analytic and possesses a natural interpretation as a distribution over probabilities being defined on the N-simplex. Possible approaches include mapping the variance of the presoftmax network output to Dirichlet parameters [48], directly including a Dirichlet distribution in the loss function definition [46], or training "prior networks" that directly output the parameters of the Dirichlet distribution [57]. Another approach is to empirically define a "confidence score" using μ and the covariance in Eq. (2) [31,36].

In this work we introduce a novel approach which also directly utilizes the covariance in Eq. (2). We consider a random variable $x \in \mathbb{R}^N$ distributed as a multivariate Gaussian truncated to lie between 0 and 1 with mean μ and covariance Σ_{q_θ} and compute the volume where

$x_i > p_{\rm th}\ \forall\ i = 1 \ldots N$ to obtain the confidence. In practice, the definition of such a distribution is complicated by the fact that the components x_i are not independent as both they and the means μ_i must sum to one. This interdependency of the components x_i implies one cannot define a multivariate Gaussian directly with the covariance Eq. (2). The full derivation of the resulting probability distribution we denote as $\mathcal{F}(x;\mu,\Sigma_{q_\theta})$ is outlined in Appendix C with the final result being

$$\mathcal{F}(x;\mu,\Sigma_{q_\theta}) = \delta\left(1-\sum_{j=1}^{N} x_j\right)\times\sqrt{N} \times \prod_{i=1}^{N-1}\tilde{N}([B^{-1}(x-\mu)]_i;0,[B^{-1}\Sigma_{q_\theta}B]_{ii}), \quad (3)$$

where B is the matrix which diagonalizes Σ_{q_θ} and $\tilde{N}$ denotes a Gaussian truncated between 0 and 1. The Dirac delta function enforces the constraint that the components must sum to one with the remaining terms being the product of $N-1$ one-dimensional Gaussians each with a variance given by the non-null eigenvalues of Σ_{q_θ}. By using the threshold probability $p_{\rm th}$ and marginalizing over the remaining labels, the probability an example is assigned the label I can then be defined as

$$P_I \equiv \int_{p_{\rm th}}^{1} dx_I \int_0^1 dx_1 \ldots \widehat{dx_I} \ldots dx_N \mathcal{F}(x;\mu,\Sigma_{q_\theta}), \quad (4)$$

where $\hat{dx}_I$ denotes that the integration on the Ith variable is omitted. In practice, to compute the integrals in Eq. (4) we sample Eq. (3) as outlined in Appendix C and determine the fraction of samples which satisfy $x_I > p_{\rm th}$. If no components of a sample exceed $p_{\rm th}$, then it is not assigned a label and the total fraction of such samples gives the probability the example is unclassifiable.

The probability P_I encodes an estimate of the uncertainty in the classification and can be used to construct a first approximation of the confidence in the following manner. Denoting $P_{\rm gauss}(n\sigma)$ to be the usual volume of a Gaussian distribution in the interval centered on a mean value with width $n\times\sigma$ we define there to be a $n\sigma$ detection of a deviation from ΛCDM if $P_{\Lambda\rm CDM} = 1 - P_{\rm gauss}(n\sigma)$. For example, a 2σ detection corresponds to $P_{\Lambda\rm CDM} = 1 - P_{\rm gauss}(2\sigma) = 1 - 0.9545 = 0.0455$. Moreover, if an example is classified with the label I at less than 1σ confidence such that $P_I < 0.68$, we shall not consider this a detection even if $\mu_I > p_{\rm th}$. Note that $1 - P_{\Lambda\rm CDM}$ represents the probability of an example not being ΛCDM, including the probability of the example being unclassifiable. It therefore does not strictly represent the probability of a non-ΛCDM detection but also includes the probability that the BNN is not able to determine which class from the training set the example belongs to.

III. TRAINING THE NETWORK

In this section we discuss the procedure of preparing the training, validation and test data, designing the network architecture and the subsequent hyperparameter optimization.

A. Generating and preparing matter power spectra

We consider three well-studied modifications to ΛCDM: the $f(R)$ gravity model described in Ref. [58], the Dvali-Gabadadze-Porrati (DGP) brane-world model of Ref. [59] and an evolving dark energy model as parametrized in Refs. [60,61] (wCDM). We compute dark matter power spectra for these theories utilizing the recently developed code ReACT [19] which calculates modified power spectra using the halo-model reaction method developed in Ref. [18]. We sample the parameter space defining each model and pass the values to ReACT, which generates power spectra in four redshift bins $z \in \{1.5, 0.785, 0.478, 0.1\}$ and one hundred k bins in the range $0.01 \leq k \leq 2.5\ h/\mathrm{Mpc}$ at equal intervals in log space, according to that expected from a Euclid-like survey [10,62]. Details about the choices of the parameter space, redshift and k ranges are given in Appendix D.

In addition to the aforementioned well-studied extensions to ΛCDM we also include an additional class to represent potential "unknown" models. Such models would imprint various signatures in the power spectrum that would be correlated in both space and time. Since *a priori* we have no way of knowing what these signals are, we produce a dataset of filters with randomly generated features correlated in k and z before applying these to randomly selected spectra from the set of ΛCDM, wCDM, $f(R)$ and DGP model spectra. We describe the method to generate this dataset in Appendix E.

For each of the five models considered, ΛCDM, wCDM, $f(R)$, DGP and random, we use 18 475 examples resulting in a total training dataset size of 92 375 examples. Every example is a matrix of dimension 100×4 with each entry given by the value of the power spectrum in the particular k and z bin. Of the 92 375 generated power spectra we set aside 15% for the validation set with the remainder used in training the BNN. Furthermore we generate a test set composed of 2500 examples per class. Gaussian noise is then added to each spectrum in accordance to what one would expect from a Euclid-like survey [17,62–64]:

$$\sigma_p(k) = \sqrt{\frac{4\pi^2}{k^2\Delta k V(z)}\times\left(P(k)+\frac{1}{\bar{n}(z)}\right)^2+\sigma_{\rm sys}^2}. \quad (5)$$

The redshift-dependent survey volume $V(z)$ and the shot noise $\bar{n}(z)$ are presented in Table II. In addition, a constant systematic error of $\sigma_{\rm sys}^2 = 25\ \mathrm{Mpc}^6/h^6$ is included to represent potential modeling inaccuracies. This value of $\sigma_{\rm sys}$ is chosen such that we are able to recover the fiducial

TABLE I. Mean model parameter values (μ) and standard deviations (σ) used in ReACT for generating ΛCDM, wCDM, $f(R)$ and DGP matter power spectra for the training, validation and test data.

	ΛCDM					Extensions			
Parameter	H_0	n_s	Ω_m	Ω_b	$\sigma_8(z=0)$	w_0	w_a	$\lvert f_{R0}\rvert$	$\Omega_{\rm rc}$
Mean (μ)	67.3	0.966	0.316	0.0494	0.766	−1	0	0	0
Variance (σ)	0.4	0.007	0.009	0.032	0.004	0.097	0.32	$10^{-5.5}$	0.173

TABLE II. Chosen redshift bins, cosmological volume and number density parameters used to construct the Gaussian errors for each redshift bin in Eq. (5).

z	0.1	0.478	0.783	1.5
$V(z)$ [Gpc$^3/h^3$]	0.283	3.34	6.27	10.43
$\bar{n}(z)$ [h^3/Mpc^3]	0.0013	0.0010	8.3×10^{-4}	3.6×10^{-4}

Planck parameters with 2σ confidence when performing an MCMC analysis using the nonlinear halofit Planck spectrum as our data vector and Eq. (5) as our errors, with a χ^2 likelihood. We leave a thorough analysis of how much this choice affects the results to future work.

For each of the original set of 92 375 examples we generate ten spectra each with a different realization of the Gaussian noise, ensuring that on top of recognizing deviations from particular models, the network is more robust to different noise realizations for the same model. In total, the number of training and validation examples is given by 923 750.

Finally, to ensure the data passed to the BNN are of comparable orders of magnitude across all scales and redshift bins we normalize each training example to a reference ΛCDM power spectrum with a cosmology given by the mean values in Table I.

In Fig. 2 we display the process of how spectra generated with ReACT are transformed before being passed to the BNN, including the addition of Gaussian noise followed by normalization by a fiducial Planck spectrum. Therefore the network is trained to detect deviations from ΛCDM for different noise realizations and choice of standard cosmological parameters.

B. Training and optimization

In this work we are concerned with the capability BNNs possess in tackling two questions. The first is how effective BNNs can be in recognizing the distinct features in the power spectrum for a particular modification to ΛCDM, such as $f(R)$ or DGP. The second is the ability of BNNs to detect a deviation from ΛCDM in the power spectrum irrespective of the particular modification. In practice, we train two BNNs with the same architecture, the first for five labels divided between ΛCDM and the four extensions and the second trained to distinguish between the two labels ΛCDM and non-ΛCDM. Due to the fact that there are only four redshift bins it is beneficial to treat the data as four separate time series and use one-dimensional convolutional layers. Treated this way, the spectra are passed to the network with dimension $100\times1\times4$, or in analogy with image classification tasks, as 100×1 pixel images with four channels. The architecture of the network used to train

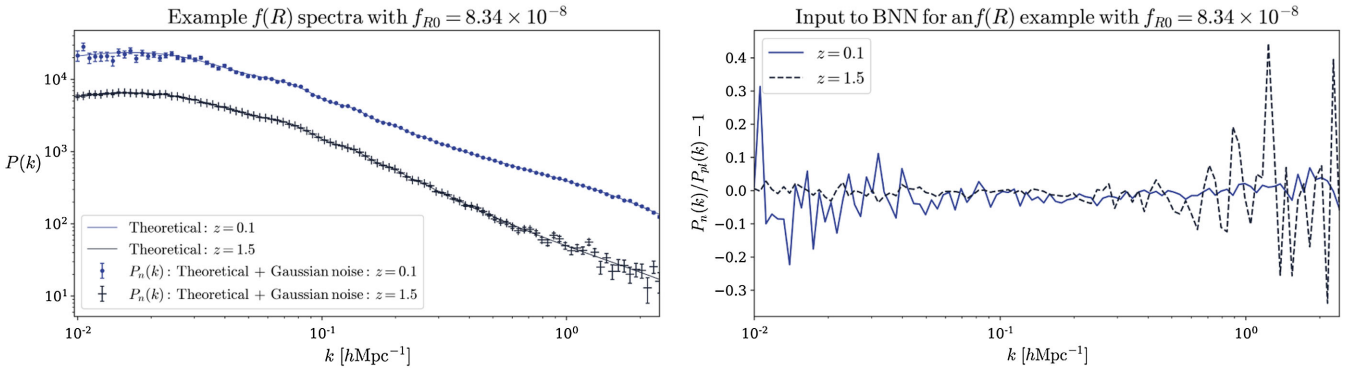

FIG. 2. Left: a pair of example $f(R)$ spectra with $f_{R0}=8.34\times10^{-8}$ at redshifts $z=1.5$ and $z=0.1$ generated using ReACT with the additional Gaussian noise. Note that at low redshift cosmic variance dominates at low k and at high redshift the shot noise dominates at high k. Right: After normalizing the noisy spectrum $P_n(k)$ by a fiducial Planck spectrum $P_{\rm pl}(k)$ and centering around zero the spectra are ready to be passed to the BNN. Due to this normalization choice the BNN is trained to detect deviations from this fiducial Planck spectrum. Note that in practice all four redshift bins are passed to the BNN. Despite the presence of such a small modification, the five-label BNN classifies this spectrum as ΛCDM with only 5% confidence, favoring the presence of a modification (see Fig. 7).

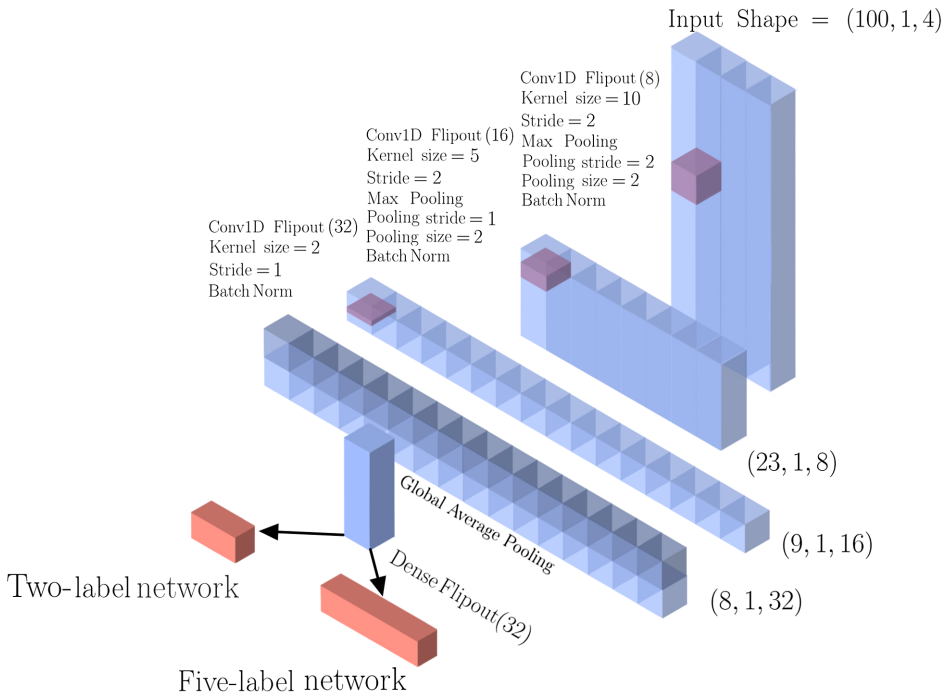

FIG. 3. Depiction of the BNN architecture employed for both the five-label and two-label classification tasks. The height of each block illustrates the dimension size for each layer, while the number of blocks per layer corresponds to the number of filters. Additionally the dense blocks embedded in the first three transparent layers indicate the kernels for the first three one-dimensional convolutional layers scaled by their respective size.

both the five-label and two-label networks is displayed in Fig. 3. Initially the structure consists of three 1D convolutional flip out layers with 8, 16 and 32 filters, kernel sizes of 10, 5 and 2 with strides of 2, 2 and 1, respectively. Each of the first two 1D convolutional layers are followed by a max pooling layer with a pool size of 2 and a pooling stride of 2 for the first max pooling layer and a pooling stride of 1 for the second max pooling layer. After both of these max pooling layers there is a batch normalization layer. Following the final convolutional layer there is a global average pooling layer to reduce the filter size to one in order to pass it to a dense layer with 32 nodes. Finally, after a further batch normalization there is a softmax layer consisting of five or two neurons for either the five- or two-label networks, respectively. The network's architecture is summarized in Table III. The five-label and two-label networks consist of 6605 and 6410 trainable parameters, respectively. We set the initial learning rate lr_0 to be 0.01 with a decay rate 0.95 such that with a training set size M and at each epoch e the learning rate is

$$\mathrm{lr}(e) = \mathrm{lr}_0 \times 0.95^{(e/M)}. \tag{6}$$

TABLE III. Description of the network's architecture.

Operation layer	Number of filters	Size	Stride	Output size	Number of parameters
Input	⋯	⋯	⋯	$100 \times 1 \times 4$	⋯
Convolution 1D flip out	8	10	2	$46 \times 1 \times 8$	648
Max pooling 1D	⋯	2	2	$23 \times 1 \times 8$	⋯
Batch normalization	⋯	⋯	⋯	$23 \times 1 \times 8$	32
Convolution 1D flip out	16	5	2	$10 \times 1 \times 16$	1296
Max pooling 1D	⋯	1	2	$9 \times 1 \times 16$	⋯
Batch normalization	⋯	⋯	⋯	$9 \times 1 \times 16$	64
Convolution 1D flip out	32	2	1	$8 \times 1 \times 32$	2080
Batch normalization	⋯	⋯	⋯	$8 \times 1 \times 32$	128
Global average pooling	⋯	⋯	⋯	32	⋯
Dense flip out	⋯	⋯	⋯	32	2080
Batch normalization	⋯	⋯	⋯	32	128
Dense flip out	⋯	⋯	⋯	5/2	325/130

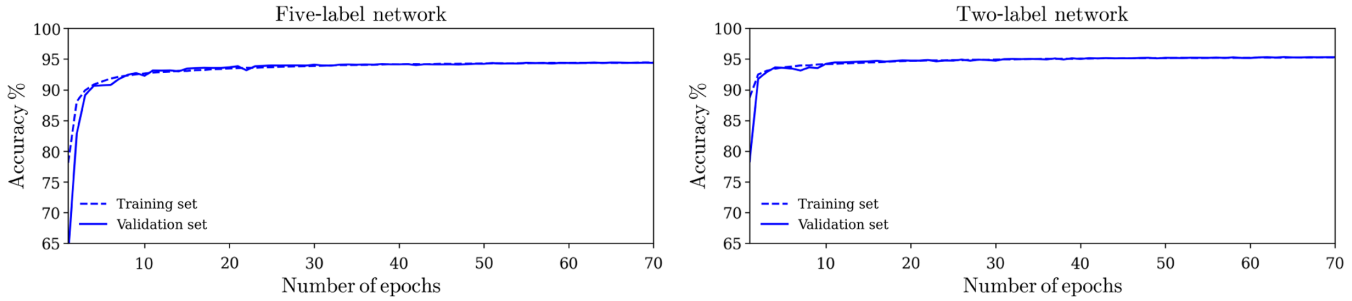

FIG. 4. Left: evolution of the training and validation accuracy for the five-label network. The accuracy on both datasets stabilizes at ∼94.4% indicating that the five-label network is relatively robust. Right: evolution of the training and validation accuracy for a network trained to classify spectra between ΛCDM or non-ΛCDM, reaching 95.3% accuracy for both datasets. Note that the accuracy for both networks is evaluated with the output from a single pass through the BNN for all the training and validation spectra and it therefore simply gives an estimate of the network's performance.

The batch size was set to 500×5. Each batch is composed of an equal number of power spectra for each of the labels in the training data. During training, we adjust the training set size by dropping a random subset of the data in order to have an integer number of batches of the same size. In Fig. 4 we show the evolution of the accuracy for a network trained to classify between ΛCDM, wCDM, $f(R)$, DGP and random spectra (top panel) and a network trained to distinguish simply between ΛCDM from non-ΛCDM (bottom panel). In each classification task for both the training and the validation sets the five-label network asymptotes to an accuracy of 94.4% and the two-label network asymptotes to a training accuracy of 95.3%. Bear in mind that the accuracy is evaluated by passing examples once through the BNN with a single draw from the weight distribution and therefore only approximates the BNN's overall performance. Note also that despite the fact the overall accuracy of the two-label network is slightly greater, it does not necessarily imply that it is generally better at detecting deviations from ΛCDM in the power spectrum (see Sec. IV).

IV. RESULTS

In this section we determine the ability of both the five- and two-label networks to classify previously unseen matter power spectra and perform tests to determine the robustness of the method. In Sec. IV A we study the overall performance of the network on the test set, followed by a calibration check in Sec. IV B as well as test the robustness of the five-label BNN against variations in the training set (Sec. IV C). We then evaluate the performance on individual spectra in Sec. IV D, including a study of the impact of noise on the classification in Sec. IV E and a comparison of the two- and five-label networks in Sec. IV F. We examine the ability of each network to recognize out-of-distribution examples which were not included in the training set in Sec. IV G before studying the constraints each network is capable of placing on the model parameters in Sec. IV H. We finally comment on the relevance for future experiments in Sec. IV I.

A. Performance on the test set

Now that the network has been trained, the next step is to evaluate its performance on the test set in order to determine how capable it is in classifying previously unseen examples. To this end, ten copies of every test example are made with different noise realizations added to the same underlying spectra. By computing the average of the output after 500 MC samples using Eq. (1), each example's label is assigned to be the maximum μ_i as long as it exceeds the threshold value $p_{\rm th} = 0.5$. If no μ_i is greater than the threshold, the example is assigned the label "not classified" (N.C.). The resulting overall test accuracies are 94.9% and 95.8% for the five- and two-label networks, respectively. As these results are comparable to the training and validation accuracies in Fig. 4 the network can be considered to be robust. In Fig. 5 we show the confusion matrices for each network which provide information on the percentage of examples from each class that are classified accurately and, if not, what class they were erroneously classified into. Theories that show a greater degree of degeneracy in their effects on the matter power spectrum are more likely to be classified incorrectly. For the five-label network, the strongest degeneracy exists between wCDM and ΛCDM, likely because the signatures of wCDM occur at length scales where the noise can dominate. Indeed, wCDM modifications appear at the level of the cosmological background. In contrast, the other theories considered here can affect the higher-order perturbations which leave a direct imprint on the power spectrum. Following ΛCDM, wCDM also possesses a slight degeneracy with DGP. By contrast, only 1% of $f(R)$ examples were misclassified which correspond to spectra with small values of f_{R0} that are noise dominated. The high efficacy the BNN has in detecting $f(R)$ models warrants a more detailed analysis which we discuss in Sec. V.

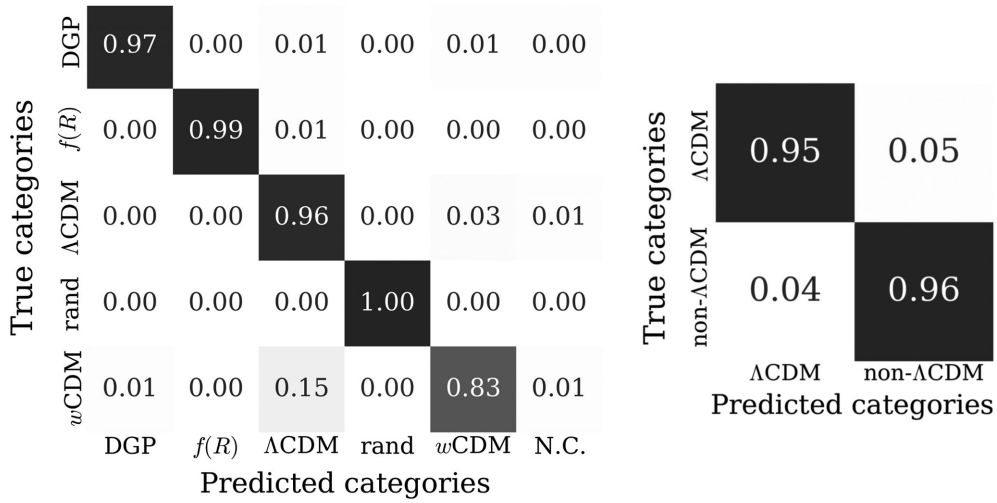

FIG. 5. The confusion matrices for the five-label network (left) and the two-label network (right) display the percentage of examples per class that are assigned to each class by the network. A classification is obtained by assigning the label to be the maximum multinomial mean μ_i as long as it exceeds the threshold $p_{\rm th}$. If no μ_i exceeds $p_{\rm th}$, the example is considered to be "not classified" (N.C.).

B. Calibration

Another important test is to ensure the probabilities given by the network represent the likelihood for a prediction to be correct. If this holds, the network is said to be well calibrated. By definition, a model is perfectly calibrated if the accuracy on all examples classified with probability p is $p \times 100\%$. Ensuring that DNNs are well calibrated is a key step in assessing their reliability [65–67].

In Fig. 6 we present reliability diagrams for both the two- and five-label networks which are constructed as follows. First, we divide predictions for μ into bins. The number of bins for each class, or component μ_i, is chosen such that each bin contains at least 0.5% of the total number of examples in the test set in order to avoid a large variance. We then compute the accuracy in every bin for each class. Let B_i denote the set of bins for the ith class, n_{bi} the number of predictions in bin b for class i and $\mathrm{acc}(b,i)$ and $\hat{\mu}(b,i)$ the corresponding accuracy and average probability for each b and i, respectively. Shown in Fig. 6 are the reliability diagrams displaying how $\mathrm{acc}(b,i)$ varies with $\hat{\mu}(b,i)$ for both the five-label network and the two-label network. For the $f(R)$ and random examples in the test set we find the probability is always either very close to 1 or 0, resulting in only two bins. By construction, the reliability diagram would result in a straight line for a perfectly calibrated network. We can quantify the deviation from perfect calibration by computing the static calibration error (SCE), defined as [67]

$$\mathrm{SCE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{b=1}^{B_i}\frac{n_{bi}}{N_{\rm tot}}|\mathrm{acc}(b,i)-\hat{\mu}(b,i)|, \qquad (7)$$

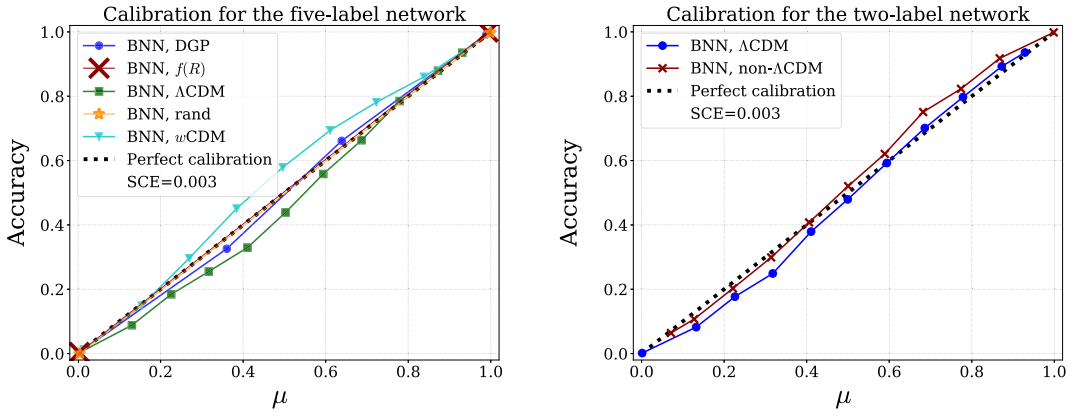

FIG. 6. The reliability diagrams for both the five-label (left) and the two-label (right) BNNs display the predictions of μ for examples in the test set divided into bins containing at least 0.5% of the total test examples plotting against the resultant test accuracy evaluated on each bin. A perfectly calibrated network corresponds to the bisector with the deviation from "perfect calibration" measured by the SCE defined in Eq. (7) also reported.

where $N_{\rm tot}$ is the total number of test examples and N is the number of labels. From their values displayed in Fig. 6 we find that both networks are well calibrated with an SCE of 0.3%. Furthermore, we verified this value remains stable under changes in the number of bins and never exceeds $\lesssim$0.5%.

C. Robustness against variations in the training set

In this subsection we evaluate the impact of using the confidence introduced in Sec. II B to detect extensions to ΛCDM in the matter power spectrum and its usefulness in taking into account uncertainty due to the presence of noise in the training set. Recall that, even when marginalized over the weights, the network's output is still conditioned on the training data [see Eq. (1)]. Despite the fact we include multiple realizations of the noise for each clean spectrum to ensure the BNN is more robust to variations in the training set, there is no guarantee this eliminates significant fluctuations in the result with slight variations in the training set. However, if the resulting classification of a particular example is highly dependent on the specific realization of the noise during training, the associated uncertainty will be large for a similar example at test time. As the covariance matrix in Eq. (2) contains an estimate of the aleatoric uncertainty, the classification confidence in Eq. (4) should be lower for such noise-dependent examples.

To explore this issue we train a second five-label network with an alternate partitioning of the data into training and validation sets before evaluating the probability μ for every example in the test set for both networks. Note that the same realization of the noise was added to each test example to ensure any variation in the result cannot be accounted for by the variation in the noise at test time. We find that for 243 test examples, or 2% of the test set, each network gives different predictions. When considering our estimated confidence, however, in 217 of these cases, or 89% of the discrepancies, both networks yield a classification confidence of $< 1\sigma$. Of the remaining 26 discrepancies, only three give inconsistent predictions and in 23 cases one of the two network predictions has a confidence of $< 1\sigma$. Since each discrepancy involves spectra with very small deviations from ΛCDM, we generate an additional dataset of 200 example spectra for each of the three extensions DGP, $f(R)$ and wCDM with narrower ranges for the model parameters in the regime where each network may give different predictions, namely $f_{R0} \in [3\times10^{-8}, 1\times10^{-7}]$, $\Omega_{\rm rc} \in [0.002, 0.06]$, $w_0 \in [-1.025, -0.975]$ and $w_a \in [-0.1, 0.1]$. In this case, we find a discrepancy in 10% of the dataset but in all these cases at least one network has a confidence of $< 1\sigma$. In 94% of these discrepant examples both networks yield a classification confidence of $< 1\sigma$ while only in a single case does one network incorrectly classify an example with a confidence $> 1\sigma$. This analysis suggests that the confidence in Eq. (4) is a more realistic indicator of a prediction's reliability with respect to μ.

D. Illustration on explicit examples

In this subsection we illustrate the new method to compute the confidence, as introduced in Sec. II B, on explicit examples.

We choose three underlying noiseless spectra belonging to the $f(R)$, DGP and wCDM classes and add a fixed realization of the noise to each of them, thus mimicking an actual observational situation where the network is given some noisy spectrum to classify. We choose the parameters and the noise realization so that the probability of being non-ΛCDM is around 95% for each example. We will investigate the role of the noise and dependence on the strength of the modifications more extensively in Secs. IV E and IV H, respectively. Following the procedure outlined in Sec. II B, we compute μ and Σ_{q_θ} with the five-label network for each spectrum. These are used to construct the distribution $\mathcal{F}$ using Eq. (3). This represents the distribution of possible outcomes of the network, taking into account the epistemic and aleatoric uncertainties. Then, according to the algorithm described in Appendix C to compute the probabilities in Eq. (4), samples are drawn from $\mathcal{F}$, each sample being a vector of dimensions equal to the number of classes (5) with values between 0 and 1, and where the dimensions $\{0, \ldots, 4\}$ correspond, respectively, to the classes DGP, $f(R)$, ΛCDM, random, and wCDM. The fraction of samples where the Ith component (with $I \in \{0, \ldots, 4\}$) lies above $p_{\rm th} = 0.5$ is determined $\forall\, I$, which gives the integral in Eq. (4). If a sample has no component above $p_{\rm th} = 0.5$, it is considered as unclassified. The fraction of samples for which this happens gives $P_{\rm Unclassified}$.

In Fig. 7 we display the results. Samples from $\mathcal{F}$ are shown in green.

The first spectrum we consider (top panel) is an $f(R)$ spectrum with $f_{R0} = 8.34\times10^{-8}$. We find it is correctly classified as $f(R)$ with a probability of 88%, with the remaining probability falling into ΛCDM with 5% and unclassified with 7%. This remains consistent with the evaluation of the confusion matrix on the test set which showed there were no $f(R)$ spectra classified as DGP or wCDM. In contrast, a DGP spectrum with $\Omega_{\rm rc} = 0.0072$ (mid panel) is classified as DGP with only 54% probability with 11% wCDM, 30% unclassified and 4% ΛCDM showing the stronger degeneracy between DGP and wCDM. Classifying wCDM spectra with small deviations in w_0 and w_a is particularly difficult for the BNN due to their high degree of degeneracy with ΛCDM and the fact its features appear in noise-dominated regions of the power spectrum. In this case, a spectrum with a deviation of $(w_0, w_a) = (-1.03, -0.04)$ (lower panel) is classified as wCDM with 78% probability, 17% unclassified and 4%

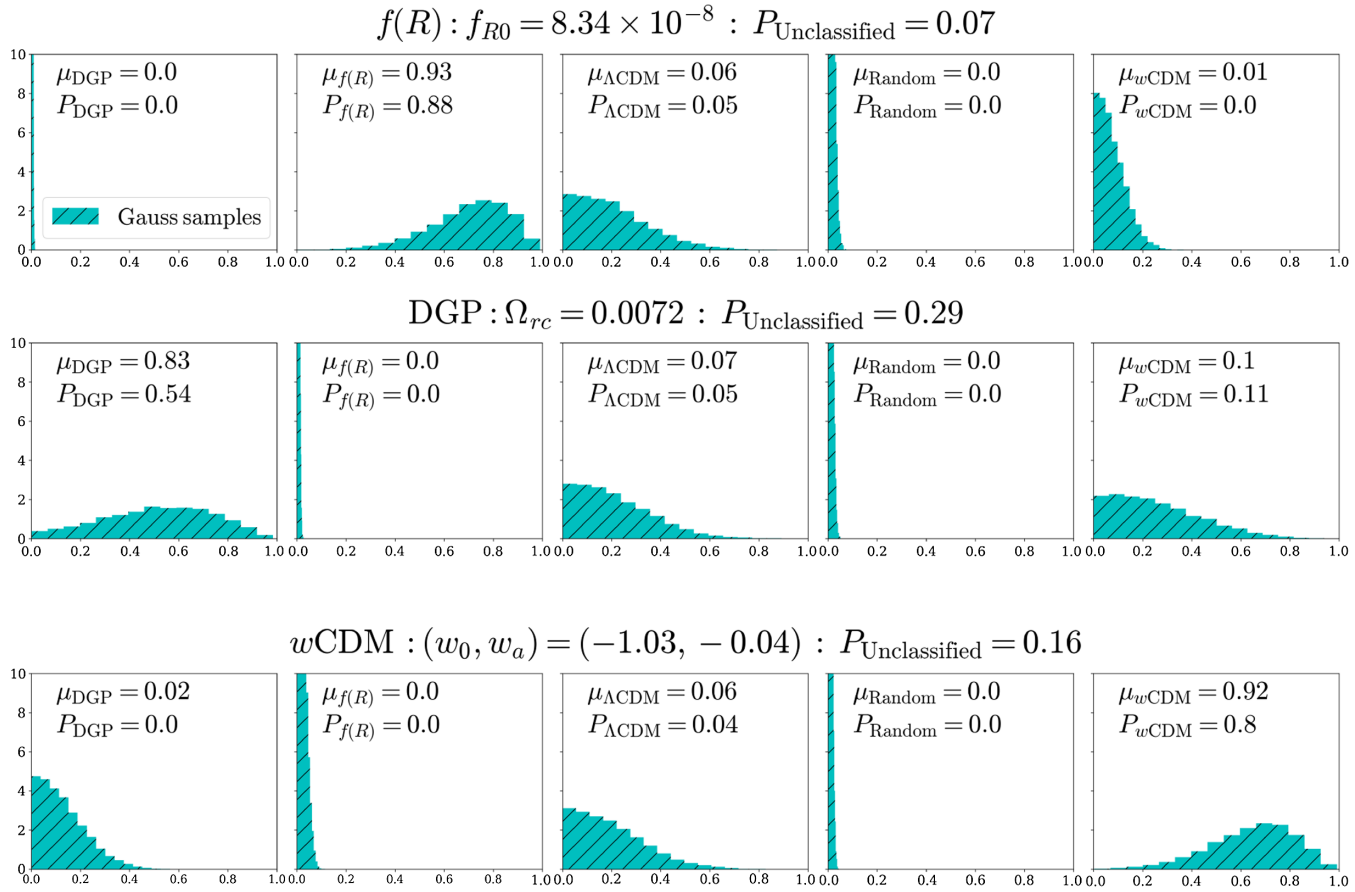

FIG. 7. Examples of a wCDM, $f(R)$ and DGP model which the BNN can correctly identify as non-ΛCDM at our chosen confidence level. We compute μ and Σ_{q_θ} as described in Sec. II B, use them to construct the probability $\mathcal{F}$ defined in Eq. (3), and then sample from this distribution. The corresponding samples are shown in green. The probabilities denoted by P_I (I = DGP, f_R, ΛCDM, random, wCDM) correspond to the fraction of samples where the Ith component of the sample lies above 0.5, i.e., Eq. (4). If a sample has no component above 0.5, it is considered as unclassified. The fraction of samples for which this happens gives $P_{\rm Unclassified}$.

ΛCDM. These modifications represent the minimum deviations from ΛCDM in each of our chosen extensions before the modifications become noise dominated and the five-label network determines the spectra to either be unclassifiable or ΛCDM.

We then repeat the procedure for the same noisy spectra with the two-label network. The result is shown in Fig. 8. Note that in the case of two labels the probability of "not classified" is always zero as it is not possible to have two samples which are simultaneously above 0.5. We find that for DGP and wCDM the two-label network classifies the examples correctly with a higher probability than the five-label network. However the $f(R)$ spectrum is not correctly classified with a high probability.

E. Dependence on the noise

It is important to emphasize the role noise plays in determining the eventual classification probability for each example in Fig. 7. In particular, it is possible that a different draw from the Gaussian noise in Eq. (5) on top of the same underlying clean spectrum could change the resulting classification. In order to obtain a measure on how much the noise affects the resulting classification for a given underlying spectrum, we compute μ and Σ_{q_θ} again starting from the same underlying noiseless $f(R)$, DGP, and wCDM spectra used for Fig. 7, but this time we further average the result over *different* noise realizations for each spectrum.

In Fig. 9 we display the distribution of outputs from the five-label BNN varying the noise realization. We stress that, differently from Fig. 7, the histograms in Fig. 9 do not represent samples from the distribution $\mathcal{F}$ in Eq. (3) but are different realizations of μ_i defined in Eq. (1) corresponding to different noise realizations on top of a given clean spectrum. This is an illustration of the potential variability of the network's output with noise.

From the μ and Σ_{q_θ} obtained from averaging over the noise we compute P_I, which now becomes a noise-averaged classification probability. The corresponding values are shown in Fig. 9. This probability gives a measure on how likely it is the network will pick up a deviation from ΛCDM given the distribution of Gaussian noise in Eq. (5).

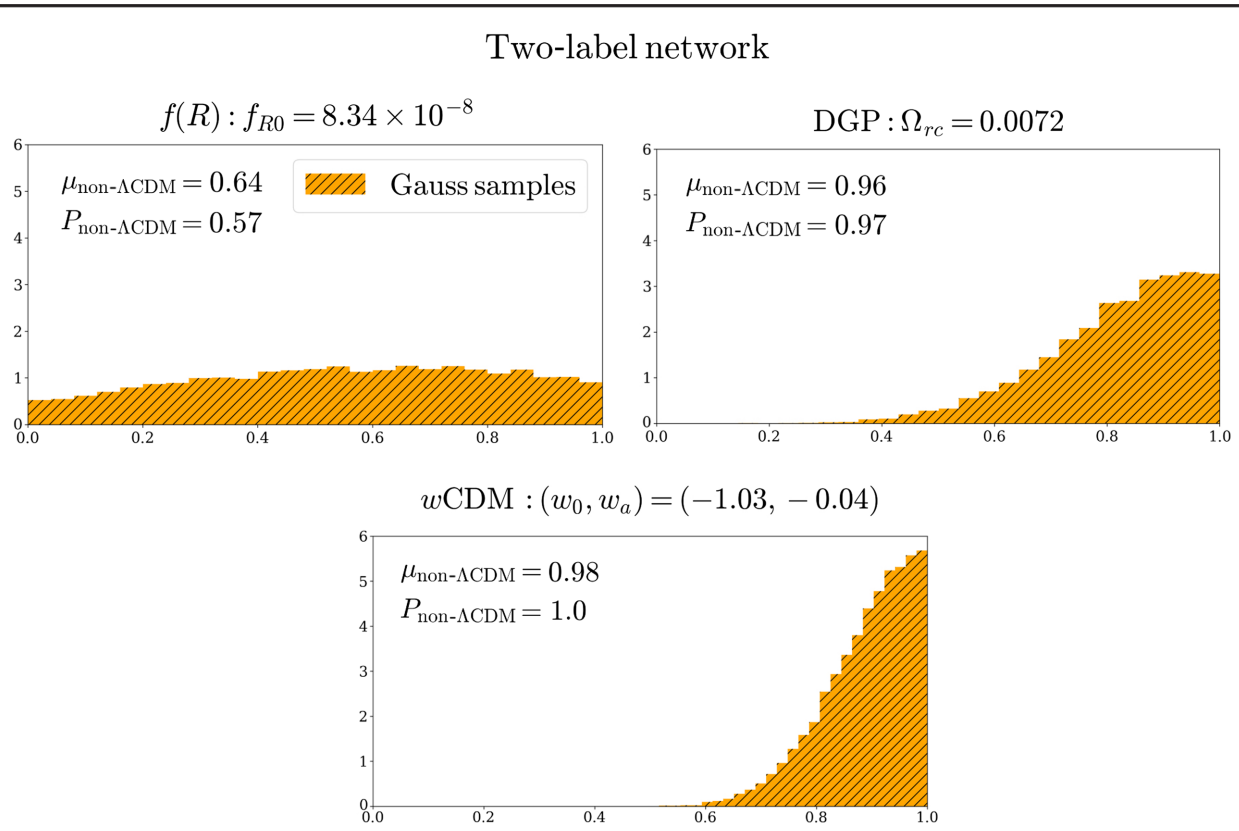

FIG. 8. We display the distributions constructed from the output of the two-label network for the same spectra and Gaussian noise which were passed to the five-label network in Fig. 7 in order to compare their relative performance in detecting deviations from ΛCDM in marginal examples. Both the wCDM and DGP examples are classified as non-ΛCDM with a higher confidence than the five-label network; however, the $f(R)$ spectrum is not correctly classified at high confidence.

For example, the $f(R)$ spectrum with $f_{R0} = 8.34 \times 10^{-8}$ possesses an average detection probability of ∼72%. In contrast, the DGP and wCDM examples both have noise-averaged detection probabilities of less than 50%. This implies that, even though the BNN classified each individual example correctly in Fig. 7, for our chosen model parameter values a correct classification was more likely to occur for the $f(R)$ example than for the DGP and wCDM examples given another realization of Gaussian noise. Note that this noise-averaged detection probability can be considered to be an invariant measure of the network's performance in classifying spectra with particular values of the model parameters.

F. Performance of five- and two-label BNNs

To compare the performance of the two-label and five-label networks more robustly, we compute the ΛCDM classification probability on the test set in each network, as well as compare the $P_{\text{non-}\Lambda\text{CDM}}$ from the two-label network with $1 - P_{\Lambda\text{CDM}}$ of the five-label network. We find that in ∼98% of the cases where the example is correctly predicted as non-ΛCDM, the five-label network can correctly classify spectra at a higher confidence than the two-label network. This is likely a result of the fact that the five-label network, possessing more final classes, can tune its layers to pinpoint specific features of each subclass, resulting in a higher confidence. By contrast, the two-label network needs to compress the information from any deviation into a single class, which can result in lower confidence due to contamination from the classes that are more difficult to distinguish from ΛCDM.

Of the 2% of spectra where the two-label network was more confident, the probability in the five-label network was either split principally between two non-ΛCDM classes, not classified, or belonged to wCDM. This indicates that the two-label network may classify non-ΛCDM spectra which do not belong to any of the classes in the training set more confidently. Such spectra are more evenly split by the five-label network between separate classes or classified as random (see Sec. IV G). However, further investigation is required to determine the necessary conditions for the two-label network to outperform the five-label network and vice versa.

G. Classification of out-of-distribution examples

To investigate how each BNN classifies examples that do not belong to either the training, validation, or test distributions, known as out-of-distribution examples, in this

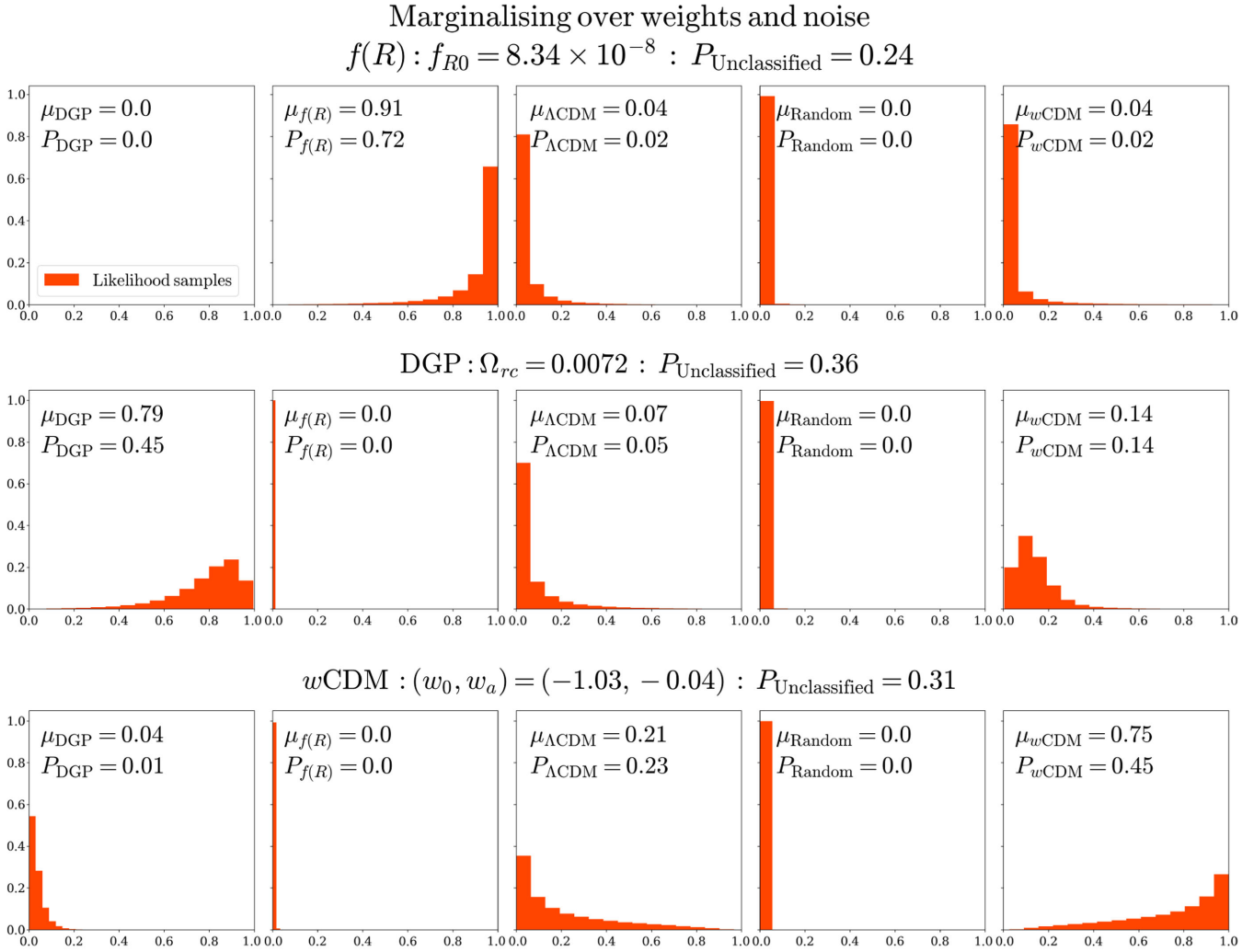

FIG. 9. Taking the same clean wCDM, $f(R)$ and DGP spectra as were passed through the network in Fig. 7, we now pass them through the network 1000 times with each pass drawing a new sample from the weight distribution and with a new realization of Gaussian noise added to the clean spectra. The resulting distributions therefore give a measure on how the network's output varies when marginalizing over the observational noise and the weights.

section we examine how each BNN classifies spectra generated both from the growth-index parameter γ [68–70] and from a painting by Gregory Horndeski.

The growth index is a frequently used phenomenological parametrization designed to pick up deviations in the growth rate of structure from its ΛCDM value of 0.55 arising from extensions to ΛCDM. The parametrization is defined by $D'(a) = \Omega_m(a)^\gamma$, where $D(a)$ is the linear density perturbation growth factor, Ω_m is the cosmological total matter density fraction and the prime denotes a logarithmic scale factor derivative. To generate nonlinear spectra with varying values of γ we first modify the linear power spectrum by applying the following parametrized growth factor:

$$D(\gamma; a_f) = \int_{a_i}^{a_f} \left[\frac{\Omega_{m,0}}{H(a)^2 a^3}\right]^\gamma \frac{a_i}{a} da, \tag{8}$$

where a is the scale factor, $H(a)$ is the ΛCDM Hubble rate and $a_i = 0.0001$ is the initial scale factor. The modified linear spectrum is then simply $P_L(k, \gamma; a) = D(\gamma; a)^2 P_0(k)$, where $P_0(k)$ is the primordial power spectrum. The modified nonlinear spectrum is produced by supplying the modified linear spectrum to the same halofit formula [13] used in producing the training data.

We find that, while a spectrum generated with a growth index of $\gamma = 0.55$ is correctly classified as ΛCDM, the associated confidence lies between 1σ and 2σ reflecting the fact that this parametrization is only an approximation of ΛCDM. Passing spectra generated with $\gamma = 0.54$ or $\gamma = 0.56$ to the five-label BNN shifts the ΛCDM classification probability to below 0.5. In Fig. 10 we display the sampled classification probabilities and the multinomial mean for each class for a spectrum generated with $\gamma = 0.52$. With the classification probability of ΛCDM being ≈5%, this value

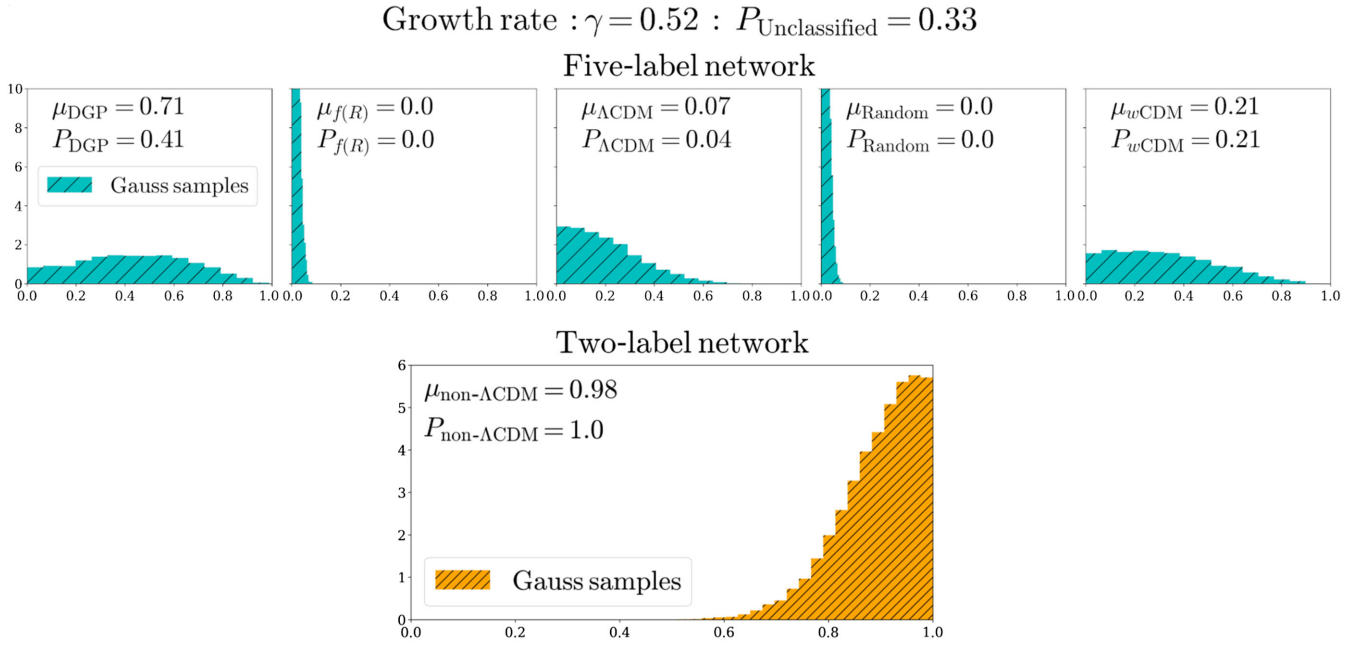

FIG. 10. If an example spectrum generated with a growth-rate parameter of $\gamma = 0.52$ with a fixed noise realization is passed through the five-label (top) BNN, this estimates this spectrum is not ΛCDM at the 2σ confidence level. However, the remaining probability is distributed between the other labels with no overall favored class, highlighting the utility BNNs possess in determining a spectrum does not belong to any of the classes in the training set. If the same spectrum with the same noise realization is passed to the two-label network (bottom), it is classified as non-ΛCDM with a higher confidence than the five-label network.

of γ represents the smallest deviation from 0.55 such that the network can confidently classify the spectrum as not being ΛCDM. Nonetheless, as no probability exceeds 0.5, no class is favored.

This ability to determine that a spectrum does not belong to the training set distribution demonstrates a unique capability of BNNs. Note also that the failure of the five-label BNN to classify a spectrum generated from the growth index as either wCDM, DGP, or $f(R)$ further highlights the limitations of the growth-index parametrization. Taking the same spectrum with $\gamma = 0.52$ with the same noise and passing it through the two-label network, we find that it is classified as non-ΛCDM with a higher confidence than the five-label network. Although this suggests the two-label network is better suited to placing constraints on the growth index, given the five-label network did not confidently classify the spectrum into any of the five labels, it is an open question how useful such constraints would be in constraining more physically motivated models. As a further test that the five-label network can identify spectra that do not belong to any known class of physical models, we pass the painting *Blustery mountain road on an autumn day* by Gregory Horndeski (see Fig. 11) to the BNN. Firstly, we convert it into a gray-scale image with 100×4 pixels. This then acts as a similar filter to those constructed in Appendix E which we then apply to the fiducial Planck spectrum before finally adding Gaussian noise [see Eq. (E1)]. The resulting matrix is then equivalent to a normalized input for the network. These deviations are large enough such that the network can accurately determine that the painting is not a ΛCDM power spectrum. However, it is also not "not classified." Rather, it is classified into the random class with 100% probability, indicating that the random class is capable of picking up examples that contain deviations which are not comparable to any model included in the training set.

H. Dependence on the strength of the modification

We have seen in Sec. IV E that a more reliable estimator of a BNN's ability to classify a non-ΛCDM spectrum with a particular modification strength is to pass the spectrum through the BNN multiple times with different realizations of the noise. The resultant probability distribution quantifies not only whether a detection is possible, but also how probable it is the noise will alter the classification. In this section we repeat this procedure for multiple $f(R)$, DGP and wCDM power spectra in the parameter range defined by the region where the five-label network transitions from classifying spectra as non-ΛCDM at low confidence to high confidence. Specifically, we build a batch of power spectra composed of different noise realizations on top of the same underlying spectrum and predict the average classification likelihood μ marginalized over the weights for all the elements in the batch. By further averaging the result over the batch, we obtain a noise-averaged classification likelihood for every example. Using this to construct the probability distribution in Eq. (3), we then compute the

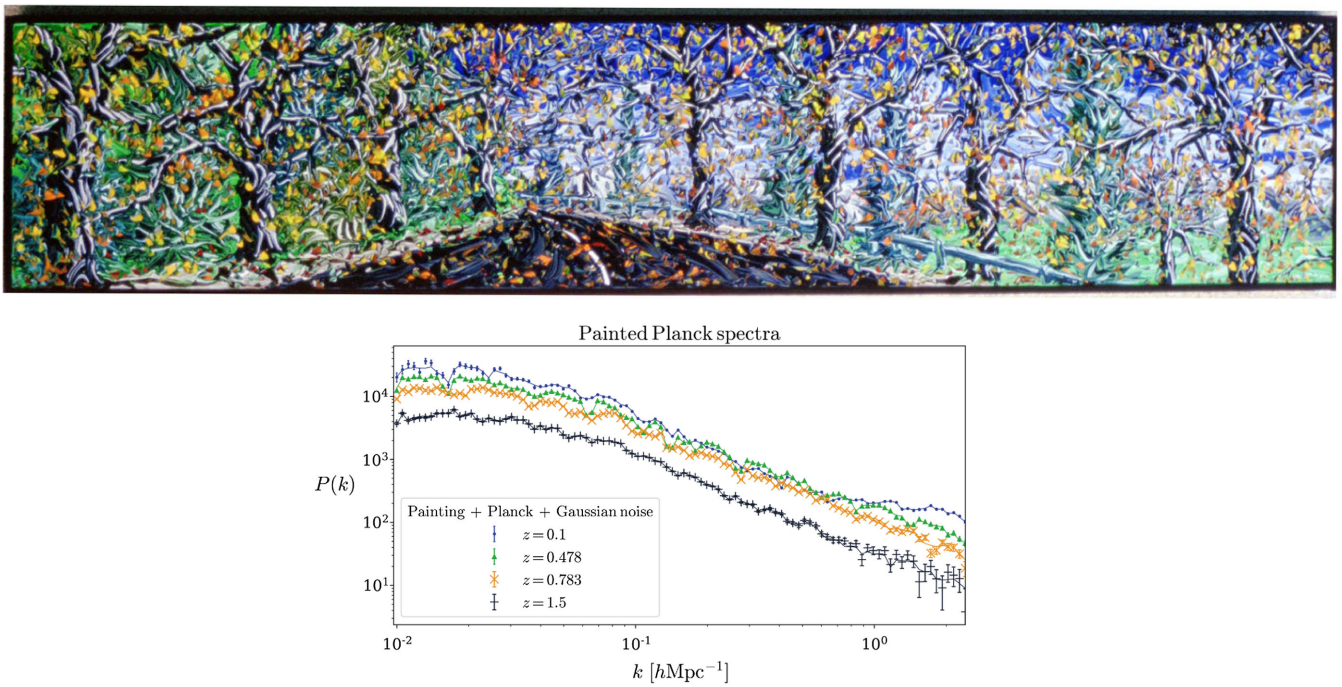

FIG. 11. If an example that does not belong to any of the pretrained classes is passed to the BNN, in this case the painting *Blustery mountain road on an autumn day* by Gregory Horndeski (top), we find that both the five-label and two-label network classifies it as non-ΛCDM with a confidence $\gg 2\sigma$. In addition, the five-label BNN classifies it as random with extremely high confidence. To pass the painting to the BNN, the pixels were rebinned into a 100×4 pixels gray-scale image which was multiplied by the fiducial Planck spectrum. Gaussian noise was then added and the resulting noisy spectra were normalized to the Planck spectrum at each redshift bin (bottom). The final normalized spectrum has significant deviations from zero, consistent with the fact that the painting imprints large random deviations from the ΛCDM spectrum.

corresponding average classification probability that the spectrum belongs to its true class and the average probability it is non-ΛCDM, defined as $1 - P_{\Lambda\mathrm{CDM}}$ for both the five- and two-label networks. This process is then repeated for spectra with different modification strengths.

In order to obtain an estimate on how much the noise can shift the classification for particular values of the model parameters, we also construct a confidence band around the average classification probability as follows. First, we remove the noise realizations such that any of the components of its predicted μ fall below the corresponding fifth or above the corresponding 95th percentile of the batch. For each network the upper bound is then obtained by selecting the noise realization among those remaining such that the probability μ_i [with i being $f(R)$, DGP or wCDM for the five-label network and non-ΛCDM for the two-label network] is maximized. The minimum bound is obtained by taking the noise realization that maximizes the difference $\mu_{\Lambda\mathrm{CDM}} - \mu_i$. While for the two-label network this is equivalent to minimizing $\mu_{\mathrm{non-}\Lambda\mathrm{CDM}}$, in the case of the five-label network it ensures that the lower bound is the minimum of both P_i and $P_{\mathrm{non-}\Lambda\mathrm{CDM}}$. This would not be guaranteed by only taking the noise realization that minimizes μ_i, due to the fact that we allow for an unclassified probability.

In Figs. 12 and 13 we present the results for $f(R)$, DGP and wCDM. One can see that in the case of $f(R)$ gravity the five-label network's non-ΛCDM classification probability is more capable of recognizing small deviations in f_{R0}, on average classifying spectra as non-ΛCDM when $f_{R0} > 8 \times 10^{-8}$. The same network becomes more confident that a spectrum specifically belongs to $f(R)$ for $f_{R0} > 1 \times 10^{-7}$. Conversely, the two-label network's ability to confidently classify spectra as non-ΛCDM remains highly sensitive to the noise up to $f_{R0} > 1.4 \times 10^{-7}$.

In the case of DGP, while the five-label network's non-ΛCDM classification probability again provides the most reliable predictions, the two-label network's ability to classify spectra as non-ΛCDM outperforms the five-label network's ability to classify the spectra as DGP. For values of $\Omega_{\mathrm{rc}} > 0.016$ both networks definitively determine all spectra are not ΛCDM independently of the noise.

Turning our attention to each network's ability to detect evolving dark energy, we show in Fig. 13 the noise-averaged classification probabilities for a range of wCDM power spectra. In each case we vary either w_0 or w_a fixing the nonvarying parameter to their ΛCDM fiducial values of $(w_0, w_a) = (-1, 0)$. Again we find that for both wCDM parameters the five-label non-ΛCDM classification probability is the most reliable indicator of a deviation from

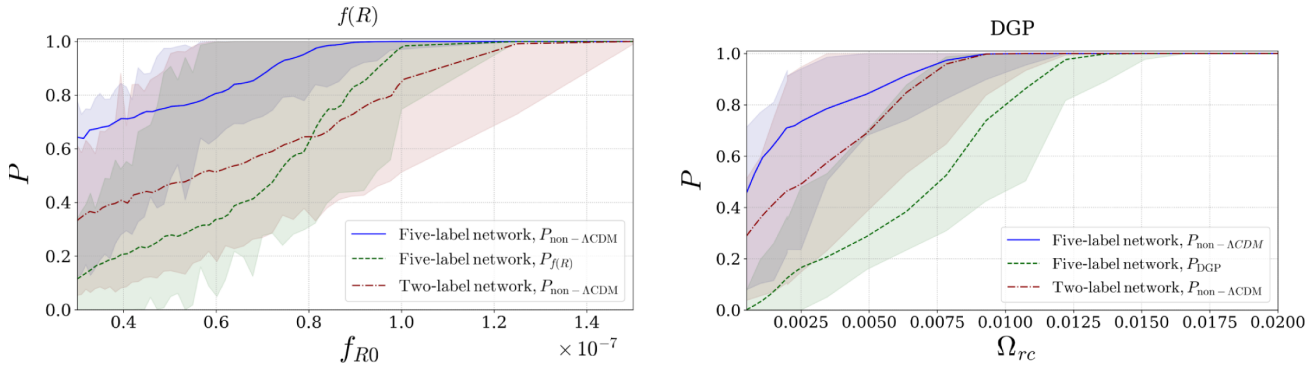

FIG. 12. Noise-averaged non-ΛCDM classification probabilities and associated confidence bands for $f(R)$ (top) and DGP (bottom) spectra for the five- and two-label networks as a function of f_{R0} and Ω_{rc}. The five-label classification probabilities $P_{f(R)}$ and P_{DGP} are also shown. One can see that the average non-ΛCDM classification probability for the five-label network provides the most robust indicator of the presence of a modification, confidently classifying spectra as non-ΛCDM for $f_{R0} \approx 9 \times 10^{-8}$ and $\Omega_{\text{rc}} \approx 0.008$ independently of the noise for $f(R)$ and DGP, respectively.

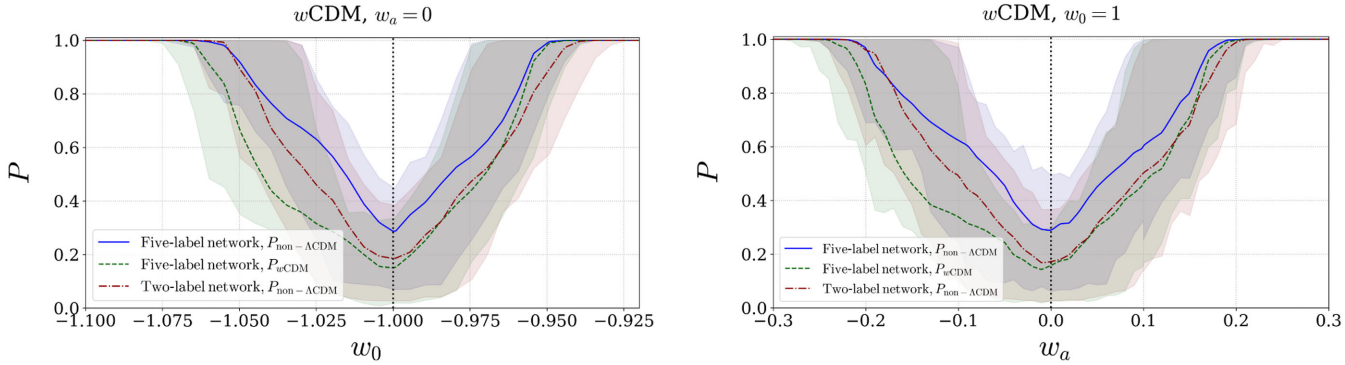

FIG. 13. Noise-averaged non-ΛCDM and wCDM classification probabilities for deviations of w_0 (top) and w_a (bottom) around their fiducial values. On average, the five-label network is better at detecting deviations from ΛCDM in each case. However the performance of each network remains highly sensitive to the noise in the ranges $-1.07 \lesssim w_0 \lesssim -0.94$ and $-0.25 \lesssim w_a \lesssim 0.25$.

ΛCDM. Despite both networks on average classifying wCDM as non-ΛCDM for deviations of $\Delta w_0 \sim 0.05$ and $\Delta w_a \sim 0.2$, the five-label network is less sensitive to the noise. We leave a detailed analysis of how the degeneracies between w_0 and w_a affect the noise-averaged classification probability to future work.

Having completed these tests for models belonging to the training set, we repeat the procedure for spectra generated with varying values of the growth index γ as outlined in Sec. IV G.

In Fig. 14 we show the noise-averaged classification probability for deviations around the ΛCDM fiducial value of $\gamma = 0.55$. Due to the absence of a specific label for γ, in this case the lower and upper bounds for the confidence bands are constructed by selecting the noise realizations that maximize and minimize $\mu_{\Lambda\text{CDM}}$, respectively. Although it appears that the two-label network can pick up smaller deviations than the five-label network for $\gamma > 0.55$, in this case it is because the five-label network recognizes the spectrum does not belong to the models in the training set

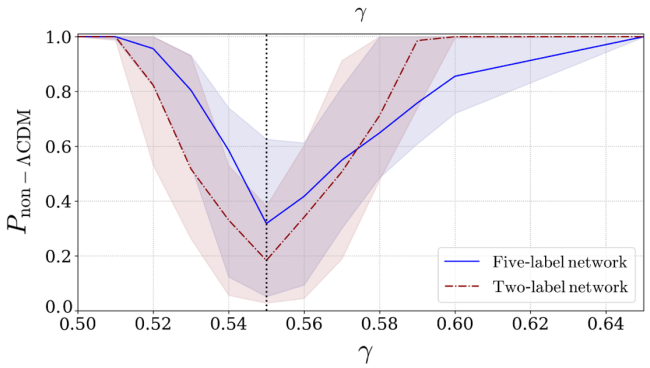

FIG. 14. Noise-averaged five-label and two-label classification probability for spectra generated using the growth index γ. For values of $\gamma > 0.6$ the two-label network classifies the spectrum as non-ΛCDM with a much higher confidence than the five-label network. However, we attribute this to the fact that the five-label network is better able to determine that this spectrum is an out-of-distribution example.

TABLE IV. Values of the minimum deviation in each of the model parameters such that both the five-label and two-label networks classify a spectrum as non-ΛCDM at 95% confidence when averaged over Gaussian noise realizations. In the case of wCDM, these bounds are obtained by fixing either w_0 or w_a to their fiducial value and allowing the other to vary. We refer the reader to Figs. 12–14 for estimates of the variance in the classification probability for each of these values.

	Extensions				
Parameter	w_0	w_a	f_{R0}	$\Omega_{\rm rc}$	γ
Five-label	$(-1.05, -0.95)$	$(-0.20, 0.17)$	8×10^{-8}	0.007	(0.52, 0.63)
Two-label	$(-1.05, -0.95)$	$(-0.20, 0.19)$	10^{-7}	0.008	(0.51, 0.59)

and thus assigns a lower classification probability. For values of $\gamma < 0.55$ the spectra are more degenerate with DGP which may help each network recognize the spectrum as non-ΛCDM. Even in this instance, however, the probability is split between DGP and other non-ΛCDM labels as in Fig. 10. Note that the bounds on the growth index γ were obtained from a network which was not trained on spectra generated from γ. One would therefore expect these bounds to improve if such spectra were included in the training set.

I. Impact on future experiments

In Table IV we summarize the values of the minimum magnitude of each model parameter for every ΛCDM extension such that the noise-averaged non-ΛCDM classification probability is approximately 95% for both the five-label and the two-label networks. The uncertainty estimated in our approach to the BNN output is of similar order of magnitude of stage IV astronomical survey forecasts, despite the two methods not being directly comparable, nor do we use the observational probes of upcoming surveys, e.g., weak lensing and galaxy clustering. We note the 1σ cosmic shear forecasts of Ref. [19] were $f_{R0} \leq 10^{-7.25}$ and $\Omega_{\rm rc} \leq 0.08$ which assumes an LSST-like survey and a multipole scale cut of $\ell_{\rm max} = 1500$. The official Euclid Fisher forecast of Ref. [62] gives $w_0 = -1 \pm 0.097(0.077)$ and $w_a = 0 \pm 0.32(0.24)$, which combines both galaxy clustering and weak lensing probes and pessimistic (optimistic) scale cuts. The 2σ constraints estimated for Euclid on γ are $\gamma = 0.55 \pm 0.036(0.026)$ for the pessimistic (optimistic) analyses of Ref. [62] for $\mathrm{WL} + \mathrm{GC}_s$. This means that the method outlined here is able to pinpoint deviations from ΛCDM down to a level relevant for Euclid.

V. TRAINING SPECIALIST NETWORKS

Given the promising performance of the five-label network in detecting deviations in $f(R)$ models down to $f_{R0} \approx \mathcal{O}(10^{-8})$, in this section we discuss the potential gains that could be achieved by training additional networks on subsets of the original five classes in the training set. Heuristically this follows the philosophy of an MCMC analysis in that in order to constrain a specific model it is beneficial to choose the most appropriate set of model parameters in the MCMC. In the case of BNNs, if one is only interested in constraining a single model beyond ΛCDM, then in order to maximize the performance of the BNN it is beneficial to only train the network on this model alongside ΛCDM. Such a network would be "specialized" to pick up any deviation from the particular source of new physics one is interested in, at the expense of losing information on potential degeneracies between different models when trained on multiple theories. In this subsection we discuss one such specialist network trained on 18 475 $f(R)$ and 18 475 ΛCDM power spectra, each of which is passed to the network during the training and validation process with ten different realizations of the noise for a total training and validation set of 369 500 power spectra. We use the architecture displayed in Fig. 3 where the final layer is now a binary classifier for the two new labels $f(R)$ and ΛCDM, finding that the training and validation accuracies reach approximately 99.5%, exceeding that of the five-label network. We now study how capable this specialist network is in constraining f_{R0} in comparison with the five-label network. In Fig. 15 we display a plot of how the noise-averaged $f(R)$ classification

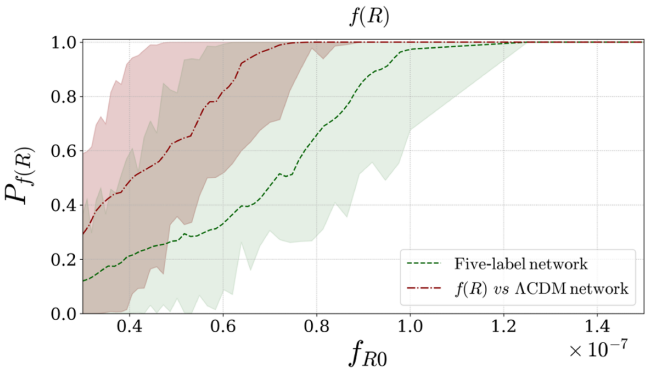

FIG. 15. We compare how the ability of both the five-label network and a specialist $f(R)$ network to correctly classify $f(R)$ spectra varies with the modification strength f_{R0} when averaged over noise realisations. It is clear that the specialist network outperforms the five-label network, with $f(R)$ spectra with $f_{R0} > 8 \times 10^{-8}$ being correctly classified largely independently of the noise realization.

probability varies for power spectra generated with values of $f_{R0} \in [3 \times 10^{-8}, 1 \times 10^{-7}]$ with associated confidence bands. We find that the performance of the specialist $f(R)$ network exceeds that of the five-label network's $f(R)$ classification capability, retaining a noise-averaged detection confidence of 1σ for $f_{R0} \approx 5.5 \times 10^{-8}$ where the equivalent noise-averaged detection probability for the five-label network is < 0.3. Furthermore, for values of $f_{R0} > 8 \times 10^{-8}$ the classification probability for the specialist BNN asymptotes to one with only a few noise realizations, decreasing this probability to 85%. Spectra with values of $f_{R0} > 9 \times 10^{-8}$ are classified at high confidence regardless of the noise realization. By contrast, the five-label network correctly classifies spectra independently of the noise when $f_{R0} > 1.2 \times 10^{-7}$. Given the limited performance of the generic two-label ΛCDM vs non-ΛCDM network and the enhanced performance of the specialized ΛCDM vs $f(R)$ network we conclude that training a two-label network is principally beneficial when trained between well-defined physical models. Tighter constraints on model parameters can also be attained with such a specialized network.

VI. OUTLOOK

The potential of BNNs to provide insights into whether upcoming cosmological datasets contain signatures of physics beyond ΛCDM motivates further exploration. For example, BNNs could be used to identify high-priority areas in the theory space by selecting the most likely known theory or motivating the need for further model development, before performing parameter estimation and Bayesian model selection with standard techniques that require specific benchmarks from which deviations can be detected. Indeed, our analysis of how the five-label BNN classified an example generated from the growth index γ in Fig. 10 demonstrates the advantages of using data from specific models over more generic parametrizations.

With enough training examples from enough ΛCDM and non-ΛCDM power spectra generated from a larger set of model classes considered in this work, one may envision a sequence of pretrained specialist networks such that the first is trained on as many deviations as possible from ΛCDM, with the latter networks trained on smaller subsets of the total number of classes. When an unknown spectrum is passed to the first network it would determine which of the subsequently more specialized networks to pass the spectrum onto. As the specialist networks are better at recognizing the specific imprints of the models they are trained on, they would further classify the spectra until it falls into either a single class or is not confidently classified. If a single model is indeed preferred following such an analysis, one could then proceed to constrain the model parameters, for example with a traditional MCMC analysis. An additional advantage is that, once the network has been trained, one has a tool to rapidly indicate the presence of new physics in contrast to the many hours it would take to obtain constraints with MCMC which must be run on a theory-by-theory basis.

Importantly, however, an MCMC possesses a well-defined notion of confidence such that for a given dataset and a given parametrization it will converge to a unique set of confidence intervals for each parameter if allowed to run for a sufficient amount of time. Moreover, it allows the computation of quantities such as the Bayesian evidence that have a well-defined interpretation in terms of model selection and a solid statistical ground, albeit being nontrivial to compute in practice [71]. As we have seen, defining a classification confidence from the output of a BNN can prove challenging as it is by no means trivial to account for the uncertainty arising in the training process, the noise in the data or the chosen network architecture. In particular, the notion of confidence introduced in this paper ensures that the resulting classification is not "overconfident" by encoding the uncertainty due to the noise. However, we stress that this quantity is not directly comparable to the Bayesian evidence or other goodness-of-fit tests. If and how these notions are comparable remains an open question (see Appendix F). In conclusion, performing a fair comparison between the two methods is not a straightforward endeavor and a more thorough analysis of their relative strengths and weaknesses in performing cosmological analyses will be a subject of future work. For now, we see the use of our BNN as a supplementary and precursory tool to MCMC analyses that accelerates the search for new fundamental physics.

While we have restricted ourselves to the matter power spectrum in this work, it is not a directly observable quantity. An additional study would therefore be to train BNNs on mock galaxy clustering and weak lensing data for a range of different theories to determine how capable BNNs are in detecting deviations from ΛCDM directly from observational data. An interesting study in this context was performed in Ref. [72], using a convolutional neural network trained on simulated convergence maps. Moreover, while we have restricted to a selected number of popular extensions to ΛCDM, this process is generically applicable to any nonstandard theory for which it is possible to rapidly generate accurate power spectra. Of particular interest would be to study the capability of BNNs to pick up signatures in the power spectrum from the presence of massive neutrinos and baryonic effects as well as modifications arising from Horndeski scalar-tensor theory. In the case of massive neutrinos, training a specialist massive neutrino network in a similar manner to that performed for $f(R)$ gravity in Fig. 15 could yield an estimate on how capable BNNs could be in indicating the presence of a nonvanishing neutrino mass. This could be achieved with the latest version of ReACT [51]. With the ever-growing ability to model cosmological observables for

a multitude of extensions to ΛCDM, in Sec. V we discussed the possibility of training a hierarchy of increasingly more specialist networks to obtain a more confident classification for a spectrum belonging to an unknown class. Note that one could also fine-tune an N-label BNN to distinguish between subclasses by incorporating an additional layer on top of a previously trained N-label network and retrain on a smaller dataset containing the new labels. Although training a two-label network from scratch takes a few hours on a GPU, we have included the option in BaCoN to fine-tune on new data in the event one wishes to adapt the BNN to classify new theories with a limited training set size.

With the capability of BNNs to extract the particular features in the data that were important in the resulting classification, it may also be possible to provide information on the length scales or redshift bins which should be probed to detect signatures of a particular theory. One can then train a BNN on spectra from specific redshift or scale bins. This may be especially useful for studying models such as wCDM whose signatures can be noise dominated at large scales and low redshift. It is also important to note that in this paper we have restricted to a k range of $(0.01 - 2.5)\ h\mathrm{Mpc}^{-1}$. We expect that the capability of BNNs to accurately classify models such as $f(R)$ will only increase with improvements in the ability to rapidly and accurately model power spectra at higher values of k.

Further avenues of exploration include studying the potential benefits of different choices of network architecture and hyperparameters. It is also of interest to examine different methods of constructing probability distributions to account for various sources of uncertainty in the output of the BNN. In this paper we have focused on the application of BNNs to a classification problem. However, the question remains of how capable neural networks are in obtaining cosmological parameter constraints from unknown spectra in comparison with more traditional approaches such as MCMC. Finally, although we have trained the network on data which lie within the bounds of Euclid errors, it is important to determine how effective BNNs could be in detecting new physics from other surveys such as LSST [11] or Dark Energy Survey [73] as well as to investigate how the results vary with different choices of systematic error.

VII. CONCLUSIONS

Over the coming years many new cosmological surveys will provide vast datasets which will determine whether ΛCDM remains concordance cosmology. In this paper we have studied the ability of Bayesian neural networks to determine if a matter power spectrum is representative of ΛCDM or not. By constructing a mapping from the output of a BNN to a well-defined probability distribution we were able to define a classification confidence for an individual spectrum that considers the uncertainty from the noise in the data, variations in the training procedure, the modeling uncertainty of the BNN and choice of hyperparameters. We found that a five-label network trained to classify between ΛCDM, $f(R)$ gravity, DGP gravity, wCDM and a "random" class provided more reliable predictions than a two-label network trained to distinguish simply between ΛCDM and non-ΛCDM. While generally being less sensitive to variations in the noise distribution, it can also determine whether a power spectrum does not belong to any class included in the training set. Since the selection of the correct model is crucial when performing conventional statistical analyses such as with MCMCs, this ability could prove beneficial in indicating prospective models to consider. However, the network used in this work is currently limited to classification tasks while the notion of model selection on firm statistical grounds in the context of BNNs remains an open problem. Nevertheless, we found that when averaged over noise realizations the five-label BNN was able to recognize spectra as not being ΛCDM down to values of $f_{R0} \lesssim 10^{-7}$, $\Omega_{\mathrm{rc}} \lesssim 10^{-2}$, $-1.05 \lesssim w_0 \lesssim 0.95$, $-0.2 \lesssim w_a \lesssim 0.2$, and $0.52 \lesssim \gamma \lesssim 0.59$, all of which are comparable with current forecasts, as discussed in Sec. IV I. Specialist networks trained on specific subsets of the classes in the training set have the potential to improve such bounds even further.

We conclude that BNNs may provide a powerful new means to search for hints of new physics in cosmological datasets. In particular, we anticipate they will serve as a powerful "filter," allowing us to narrow down the theory space before moving on to constrain model parameters with MCMCs while perhaps even signaling the presence of new physics that does not belong to any known model.

Alongside this paper we publish the publicly available code Bayesian cosmological network (BaCoN) which can be accessed at the github repository [74] with the training and test data available [75].

ACKNOWLEDGMENTS

We thank Tom Charnock for useful discussions. We thank Gregory Horndeski for suggesting and allowing the use of *Blustery mountain road on an autumn day* in this work. J. K., B. B., and L. L. acknowledge the support by a Swiss National Science Foundation (SNSF) Professorship grant (No. 170547). The work of M. M. is supported by the SwissMap National Center for Competence in Research. BNN development was conducted on Google Colab. Please contact the authors for access to research materials.

APPENDIX A: NEURAL NETWORKS FOR SUPERVISED LEARNING

In this appendix we introduce some basic concepts on neural network classifiers. Consider a dataset of the form $\mathcal{D} = \{(X, y)_k\}$, $k \in [1, \ldots, M]$ where each element consists of a pair of *features* X and an associated label y and M denotes the size of the dataset. Let us further denote N to be the number of possible labels. In a supervised classification

task the aim is to use the labeled examples in $\mathcal{D}$ in such a way to be capable of predicting $y^\star$ for a previously unseen $X^\star \notin \mathcal{D}$. Note that, for our purposes, each X_k consists of a collection of matter power spectra in different redshift bins with y_k labeling the associated underlying physical model (see Sec. III A).

Neural networks provide a powerful means to model nonlinear features in labeled data by combining a hierarchy of nonlinear functions in a succession of *layers*, each with optimized parameters, which map given features to a predicted label. Different choices can be made for the number of layers and the type and size of each layer, all of which constitute the network's *architecture*. To begin, the labeled dataset is split into a *training set*, *validation set* and a *test set*. By passing the training data through the network, usually in a series of batches, the network parameters are tuned using an optimization algorithm to minimize a *loss function* which quantifies how close the output of the network matches the associated label of the input data. An *epoch* occurs when every batch of data in the training set has been passed through the network.

Central to the optimization procedure is *gradient descent* which updates the parameters in the direction where the derivative of the loss, computed using the *backpropagation* algorithm, is maximally negative. Many modifications to gradient decent have been developed in order to aid the optimization efficiency and the choice of optimization algorithm and its associated parameters, known as *hyperparameters*, is an important factor in determining the performance of the network. In this paper we use the *adam* optimization algorithm [76]. The most relevant hyperparameter is the *learning rate* which sets the amplitude of the step made in the direction of the gradient.

Following each epoch a performance metric is computed to evaluate how effectively the network maps features to labels in both the training set and the validation set, the latter giving a measure on how well the network generalizes to previously unseen data. Note that this metric need not be the same function as the loss. In particular, the loss function must be differentiable with respect to the weights while the metric does not. Different choices for the loss function and the metric depend on the problem at hand and are a key consideration in the network design. Typically in classification problems the performance metric is the accuracy which is simply the fraction of correctly classified examples. This process is then repeated until the loss stabilizes to a minimum. As the performance metric on the validation set can remain biased, the final stage is to evaluate the network performance on the test set which has not been used in the training process. If the performance metric on the test set is comparable to that on the training and validation sets, then one can be confident the network is robust.

APPENDIX B: CLASSIFICATION IN BNNs

Evaluating the performance of a neural network cannot be limited to evaluation of a performance metric on the test set, especially if the network is to be used in a scientific context. In this case it is imperative to assess its reliability on any individual prediction and to define a probability that quantifies how much the prediction can be trusted.

Using traditional DNNs to compute both aleatoric and epistemic uncertainties (defined in Sec. II) would be both computationally expensive and time consuming. This appendix details why this is so before discussing how BNNs are better suited to model classification uncertainties. We define the labels to be *one-hot encoded*, such that they are vectors of length N with a one at the position of the true label and with zeros otherwise. For example, the vector $y = (1, 0, \ldots)$ is a label for an example belonging to the first class. Classification occurs when the final layer of the network outputs an N-dimensional vector with components that sum to one and can therefore be interpreted as probabilities. Denoting $f(X|w, a)$ the vector-valued output of the final layer given the weights w and an architecture a, a probability that X belongs to the ith class can be obtained by passing it to the *softmax* function

$$p(y_i = 1|X, w, a) = \frac{e^{-f_i(X|w,a)}}{\sum_{i=1}^{N} e^{-f_i(X|w,a)}}. \tag{B1}$$

We can then choose a multinomial probability distribution as a likelihood such that

$$\mathcal{L}(\mathcal{D}|w, a) = \prod_{k=1}^{M}\prod_{i=1}^{N}[p(y_{k,i} = 1|X_k, w, a)]^{y_{k,i}},$$
$$\sum_{i=1}^{N} p(y_{k,i} = 1|X_k, w, a) = 1, \tag{B2}$$

with the loss function being the negative log-likelihood. From now on we shall drop the explicit dependence on the architecture a but it should be kept in mind that all the results are conditioned on the choice of a.

The training procedure yields a maximum likelihood estimate set of weights $\hat{w}$. When predicting the label for a new example with features $X^\star$ the network outputs the probability

$$p(y_i^\star = 1|X^\star, \hat{w}, \mathcal{D}) \quad \forall \; i = 1, \ldots, N, \tag{B3}$$

where the conditioning on $\mathcal{D}$ and $\hat{w}$ indicates that the training has been performed with this dataset resulting in a particular maximum likelihood estimate for the weights. Note however that $\hat{w}$ is not a unique value dependent on $\mathcal{D}$ and the optimization process, due to the inherent stochasticity of the training process. A prediction for the label is obtained by assigning the label to the maximum output probability $y^\star_{\text{pred}} = \arg\max_i p(y_i^\star = 1|X^\star, \hat{w}, \mathcal{D}, a)$ if this exceeds a chosen threshold probability p_{th}. One must be careful not to interpret Eq. (B3) as the confidence in the prediction due to the explicit dependence on $\hat{w}$, $\mathcal{D}$,

variations in the training procedure, choice of optimization algorithm or initialization of the weights, and the presence of aleatoric uncertainty. Estimating the uncertainty would require the expensive procedure of averaging the results from an ensemble of independently trained DNNs.

Fortunately, BNNs can quantify the uncertainty more efficiently by replacing each weight in the network by a parametrized distribution [24,25]. The training objective is then to infer the posterior distribution of the weights conditioned on the training data

$$p(w|\mathcal{D}) = \frac{\mathcal{L}(\mathcal{D}|w)p(w)}{p(\mathcal{D})}. \tag{B4}$$

In practice, $p(w|\mathcal{D})$ is intractable and so approximations or sampling techniques are employed. One such approximation approach is variational inference where the posterior is approximated by a variational distribution $q_\theta(w)$ which describes a family of distributions parametrized by the parameter θ [77–79]. In training the BNN, the objective is to ensure the resulting variational distribution $q_\theta(w)$ matches the posterior weight distribution $p(w|\mathcal{D})$ as accurately as possible. To achieve this it is necessary to have a measure on the difference between two distributions which could serve as a loss function. One such measure capable of quantifying how much the two distributions $q_\theta(w)$ and $p(w|\mathcal{D})$ differ is the Küllback-Leibler (KL) divergence given by [80]

$$\mathrm{KL}[q_\theta(w)\|p(w|\mathcal{D})] \equiv \int dw q_\theta(w) \log \frac{q_\theta(w)}{p(w|\mathcal{D})}. \tag{B5}$$

Using Bayes theorem to reexpress the posterior $p(w|\mathcal{D})$ in terms of the likelihood $\mathcal{L}(\mathcal{D}|w)$ and the prior distribution over the weights $p(w)$ this can be reexpressed as [26,81,82]

$$\mathrm{KL}[q_\theta(w)\|p(w|\mathcal{D})] = \mathrm{KL}[q_\theta(w)\|p(w)] - \mathbb{E}_{q_\theta(w)}[\log \mathcal{L}(\mathcal{D}|w)] + \text{const}, \tag{B6}$$

where the constant term arises from the Bayesian evidence which does not affect the optimization process.

The KL divergence between the variational distribution and the prior can be interpreted as a regularization term that ensures the variational distribution does not become too complex, potentially leading to overfitting, while the second term is the usual negative log-likelihood. By sampling the weights from the variational distribution $w \sim q_\theta(w|\mathcal{D})$ one can obtain a Monte Carlo (MC) estimate for the loss in Eq. (B6). However, given that the weights w are now random variables it is not possible to take derivatives directly to perform gradient descent.

To circumvent this issue, Refs. [83,84] detail a reparametrization trick which, rather than sampling directly from the variational distribution, samples a new random variable ϵ from a standard Gaussian such that $\epsilon \sim p(\epsilon)$, where $p(\epsilon) = \mathcal{N}(0,1)$. This in turn is related to the weights via a deterministic function such that $w = g(\epsilon, \theta)$. Now that the weights are expressed as a deterministic function, itself now a function of the random variable ϵ, it is possible to perform backpropagation. The drawback of this approach is that the resulting sampled weights are the same for each batch, correlating the resulting gradients and slowing the convergence of the optimization algorithm. In order to decorrelate the gradients across the batch Ref. [85] proposed the *flip out* method. Assuming the variational distribution can be expressed as a mean plus a perturbation, by randomly multiplying each perturbation by either $\{1, -1\}$ one can ensure that the weights across a batch are at least partially decorrelated. This method has proven to be effective in recent applications of BNNs [31,33], with the additional advantage of being available as prebuilt implementations in popular deep learning libraries like TensorFlow [86] for both dense and convolutional layers [87]. In this paper we make use of TensorFlow [88] and TensorFlow Probability [87] throughout. Following training, the posterior weight distribution can be used to obtain predictions by marginalizing over the weights, generalizing Eq. (B3) to

$$p(y_i^\star = 1|X^\star, \mathcal{D}) = \int p(y_i^\star = 1|X^\star, w, \mathcal{D}) p(w|\mathcal{D}) dw. \tag{B7}$$

In practise, this equation is evaluated by Monte Carlo sampling from the distribution $q_\theta(w)$, yielding Eq. (1) in the main text.

APPENDIX C: CONSTRUCTION OF A PROBABILITY DISTRIBUTION FROM THE OUTPUT OF A BNN

1. *N*-label distribution

In this appendix we shall detail the construction of the probability distribution introduced in Eq. (3). We aim to define a probability for a random variable x, with mean μ and covariance Σ_{q_θ}. The random variable x represents the softmax output of the network. It is therefore subject to the following conditions that each component of x lies between 0 and 1 and that the components sum to 1, namely:

$$\sum_{i=1}^{N} x_i = 1, \qquad \sum_{i=1}^{N} \mu_i = 1. \tag{C1}$$

If the components of x were independent, we could define its distribution as a multivariate Gaussian

$$\mathcal{N}(x; \mu, \Sigma_{q_\theta}), \tag{C2}$$

truncated between 0 and 1. However, due to the constraint in Eq. (C1) the matrix Σ_{q_θ} as defined in Eq. (2) is

degenerate so that $\det\Sigma_{q_\theta} = 0$ implying the multivariate Gaussian distribution is not defined. In particular, the columns of the matrix satisfy [89]

$$[\Sigma_{q_\theta}]_{Ni} = -\sum_{k=1}^{N-1}[\Sigma_{q_\theta}]_{ik}, \tag{C3}$$

and Σ_{q_θ} has a null eigenvalue. To circumvent this problem we can introduce a small perturbation ϵ such that Eq. (C1) becomes

$$\sum_{i=1}^{N} x_i = 1-\epsilon, \qquad \sum_{i=1}^{N}\mu_i = 1-\epsilon. \tag{C4}$$

Defining $\tilde{\Sigma}_{q_\theta}$ to be the covariance obtained from the definition in Eq. (2) with the perturbed constraint (C4), the degeneracy condition (C3) becomes

$$[\tilde{\Sigma}_{q_\theta}]_{Ni} = -\sum_{k=1}^{N-1}[\tilde{\Sigma}_{q_\theta}]_{ik} + \epsilon\mu_i. \tag{C5}$$

The matrix $\tilde{\Sigma}_{q_\theta}$ is now invertible and the distribution in Eq. (C2) is well defined. Now we take the limit $\epsilon \to 0$. Since Σ_{q_θ} is symmetric, there exists an orthogonal matrix B such that $\Sigma_{q_\theta} = BUB^{-1}$, with $U = \mathrm{diag}(u_1, \ldots, u_{N-1}, 0)$, where u_i are the eigenvalues of Σ_{q_θ}, $u_i \neq 0 \ \forall \ i = 1, \ldots, N-1$. The effect of the correction ϵ is to shift the value of the eigenvalues. In particular, the last eigenvalue becomes nonzero, resulting in the following diagonal matrix:

$$\tilde{U} = \mathrm{diag}(u_1 + \alpha_1\epsilon, \ldots, u_{N-1} + \alpha_{N-1}\epsilon, \alpha_N\epsilon), \tag{C6}$$

where the form of the coefficients $\alpha_1, \ldots, \alpha_N$ is not relevant to the present discussion. Defining $\tilde{B}$ to be the matrix such that $\tilde{\Sigma}_{q_\theta} = \tilde{B}\,\tilde{U}\,\tilde{B}^{-1}$, the variable

$$\tilde{Z} = \tilde{B}^{-1}(x-\mu) \tag{C7}$$

is distributed as

$$\begin{aligned}\mathcal{N}(\tilde{Z}; 0, \tilde{U}) &= \left(\prod_{i=1}^{N-1}\mathcal{N}(\tilde{Z}_i; 0, \tilde{U}_{ii})\right) \times \mathcal{N}(\tilde{Z}_N; 0, \sqrt{\alpha_N\epsilon}) \\ &\equiv \tilde{\mathcal{F}}(\tilde{Z}; 0, \tilde{U}).\end{aligned} \tag{C8}$$

In the limit $\epsilon \to 0$ we have

$$\begin{aligned}\tilde{\mathcal{F}}(\tilde{Z}; 0, \tilde{U}) &= \left(\prod_{i=1}^{N-1}\mathcal{N}(\tilde{Z}_i; 0, \tilde{U}_{ii})\right) \times \mathcal{N}(\tilde{Z}_N; 0, \sqrt{\alpha_N\epsilon}) \\ &\to \left(\prod_{i=1}^{N-1}\mathcal{N}(Z_i; 0, u_i)\right) \times \delta(Z_N).\end{aligned} \tag{C9}$$

Hence, we can define the distribution of $Z \equiv B^{-1}(x-\mu)$ as

$$\mathcal{F}(Z; 0, U) = \lim_{\epsilon\to 0}\tilde{\mathcal{F}}(\tilde{Z}; 0, \tilde{U}) = \delta(Z_N) \times \prod_{i=1}^{N-1}\mathcal{N}(Z_i; 0, u_i). \tag{C10}$$

From the definition in Eq. (C7), it follows the distribution of x can be defined as

$$\begin{aligned}\mathcal{F}(x; \mu, \Sigma_{q_\theta}) &= \lim_{\epsilon\to 0}\tilde{\mathcal{F}}(\tilde{B}^{-1}(x-\mu); 0, \tilde{B}^{-1}\tilde{\Sigma}_{q_\theta}\tilde{B}) \\ &= \delta([B^{-1}(x-\mu)]_N) \\ &\quad \times \prod_{i=1}^{N-1}\mathcal{N}([B^{-1}(x-\mu)]_i; 0, [B^{-1}\Sigma_{q_\theta}B]_{ii}).\end{aligned} \tag{C11}$$

It can be shown that the matrix B has elements [90]

$$\begin{aligned}B_{ij} &= M_i(u_j - M_i)^{-1}\gamma_j, \\ \gamma_j &\equiv \left[\sum_{k=1}^{N} M_k^2(M_k - u_j)^{-2}\right]^{-1/2},\end{aligned} \tag{C12}$$

where we remind the reader that u_j is the jth eigenvalue of Σ_{q_θ}. Therefore, for the Nth element of the vector Z which has a zero eigenvalue, we have

$$\begin{aligned}Z_N = [B^{-1}(x-\mu)]_N &= \sum_{j=1}^{N} B_{jN}(x_j - \mu_j) \\ &= \sum_{j=1}^{N}(\mu_j - x_i)\gamma_N \\ &= \frac{1}{\sqrt{N}}\left(1 - \sum_{j=1}^{N} x_j\right).\end{aligned} \tag{C13}$$

Substituting the above relation into Eq. (C11) and enforcing the requirement that each value of x_i must lie between 0 and 1, one finally obtains Eq. (3). Formally, this is accomplished by multiplying the distribution (C11) by a multidimensional indicator function of the interval [0, 1] and properly renormalizing, which yields the truncated Gaussian distribution denoted by $\tilde{\mathcal{N}}$ in (3). In practice, we are interested in sampling from the distribution in order to compute the probability P_I in Eq. (4). In order to draw the samples we proceed with the following algorithm: where goal is the desired number of samples and n_{samples} is the total number of valid samples obtained at each step.

2. Two-label distribution

It is instructive to consider the special case of $N = 2$, where the derivation in Appendix C 1 results in a simple analytic closed form. In particular, we find that

Algorithm 1.

while $n_{\rm samples} \leq$ goal **do**
- draw samples $Z_i, i = 1, \ldots N-1$, from $\mathcal{N}(0, u_i)$
- set $Z_N = 0$
- compute $x = BZ + \mu$

if $0 < x_i < 1 \ \forall \ i$, **then**
accept sample
else
reject sample
end if
end while

$$\mu = (\mu_1, \mu_2), \qquad \mu_2 = 1 - \mu_1, \tag{C14}$$

$$\tilde{\Sigma}_{q_\theta} = \begin{pmatrix} \sigma^2 & -\sigma^2 + \mu_1\epsilon \\ -\sigma^2 + \mu_1\epsilon & \sigma^2 + \epsilon(1 - 2\mu_1 - \epsilon) \end{pmatrix}, \quad \sigma^2 \equiv \mu_1 - \mu_1^2, \tag{C15}$$

$$\tilde{B} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 + \frac{(1-2\mu_1)}{2\sigma^2}\epsilon + \mathcal{O}(\epsilon^2) & 1 + \frac{(1-2\mu_1)}{2\sigma^2}\epsilon + \mathcal{O}(\epsilon^2) \\ 1 & 1 \end{pmatrix}, \tag{C16}$$

$$\tilde{U} = \begin{pmatrix} 2\sigma^2 + (1-2\mu_1)\epsilon + \mathcal{O}(\epsilon^2) & 0 \\ 0 & \epsilon/2 + \mathcal{O}(\epsilon^2) \end{pmatrix}, \tag{C17}$$

and

$$\tilde{Z} = \frac{1}{\sqrt{2}} \begin{pmatrix} -X_1 + X_2 + \mu_1 - \mu_2 + \mathcal{O}(\epsilon) \\ X_1 + X_2 - \mu_1 - \mu_2 + \mathcal{O}(\epsilon) \end{pmatrix}. \tag{C18}$$

After applying Eq. (C11) we obtain

$$\begin{aligned} \mathcal{F}(x; \mu, \Sigma_{q_\theta}) &= \delta\left(\frac{X_1 + X_2 - \mu_1 - \mu_2}{\sqrt{2}}\right) \\ &\quad \times \tilde{\mathcal{N}}((-X_1 + X_2 + \mu_1 - \mu_2)/\sqrt{2}; 0, \sqrt{2}\sigma) \\ &= \sqrt{2} \times \delta(X_1 + X_2 - 1) \\ &\quad \times \tilde{\mathcal{N}}(\sqrt{2}(-X_1 + \mu_1); 0, \sqrt{2}\sigma) \\ &= \delta(X_1 + X_2 - 1) \times \tilde{\mathcal{N}}(X_1; \mu_1 \sigma). \end{aligned} \tag{C19}$$

Using these relations it is then possible to verify that the marginal probabilities are

$$p(x_1) \equiv \int dx_2 \mathcal{F}(x; \mu, \Sigma_{q_\theta}) = \tilde{\mathcal{N}}(x_1; \mu_1, \sigma), \tag{C20}$$

$$p(x_2) \equiv \int dx_1 \mathcal{F}(x; \mu, \Sigma_{q_\theta}) = \tilde{\mathcal{N}}(x_2; \mu_2, \sigma), \tag{C21}$$

while the cumulative distribution functions satisfy

$$P(x_1) \equiv \int_0^{x_1} dx_1' p(x_1') = 1 - P(x_2). \tag{C22}$$

APPENDIX D: GENERATING MATTER POWER SPECTRA

In order to obtain the training, validation and test data, we use the recently developed code ReACT [19] which calculates modified power spectra using the halo-model reaction method developed in Ref. [18]. This method has been shown to be accurate to ≈2% up to $k \sim 2.5\ h/\mathrm{Mpc}$ at $z \leq 1$ when compared to full N-body simulations in $f(R)$, DGP and wCDM models (see Figs. 3, 8 and 10 in Ref. [18]). In this paper we model power spectra as described in Ref. [18] with the exception that the pseudo power spectrum is modeled using the halofit formula of Ref. [13] which has the same level of accuracy as the approach used in Ref. [18]. We refer the reader to Refs. [19,91] for more details on the generation of the training data. Power spectra are generated in four redshift bins $z \in \{1.5, 0.785, 0.478, 0.1\}$ and one hundred k bins in the range $0.01 \leq k \leq 2.5\ h/\mathrm{Mpc}$ at equal intervals in log space. These binning choices are made according to that expected from a Euclid-like survey [10,62]. The maximum cutoff in k is chosen to maintain a ∼2% accuracy between the power spectrum generated from ReACT and simulations as shown in Ref. [18]. Each power spectrum is then generated by sampling the parameter space defining each model and passing these values to ReACT. The ΛCDM parameter space is sampled using a Gaussian distribution centered on the Planck 2018 best fit parameters [92], with each standard deviation given by the Euclid "pessimistic" forecast results using weak lensing plus spectroscopic galaxy clustering ($\mathrm{WL} + \mathrm{GC_s}$) [62]. For $f(R)$, DGP and wCDM we use a Gaussian centered at the values which are equivalent to ΛCDM, namely [93] $f_{R0} = \Omega_{\rm rc} = w_a = 0$ and $w_0 = -1$. The standard deviations for $f(R)$ and DGP parameters are given by the recent results of Ref. [19] and are summarized in Table I. The standard deviation for the wCDM parameters are taken again from the pessimistic $\mathrm{WL} + \mathrm{GC_s}$ forecasts for Euclid of Ref. [62].

APPENDIX E: GENERATING "RANDOM" POWER SPECTRA

In this appendix we describe how we generate power spectra with random features representing potentially exotic extensions to ΛCDM not encompassed by any of the wCDM, $f(R)$ or DGP models. The random features will be encoded in a filter in the form of a 100×4 array such that each filter has the same dimension as each example in the training set. We denote the ith row and jth column filter entry as $F[i, j]$, where i corresponds to a $k \in [0.01, 2.5]\ h/\mathrm{Mpc}$ and j to a

$z \in \{1.5, 0.785, 0.478, 0.1\}$. $F[i,j]$ assumes values centered around 1 with a value of 1 indicating no modification in that k and z bin. This filter can then be applied to an example from our training set, $P_{\rm ref}[i,j]$, to obtain a randomly modified power spectrum

$$P_{\rm random}[i,j] = F[i,j] \times P_{\rm ref}[i,j]. \tag{E1}$$

The filter $F[i,j]$ is constructed in the following manner:

(1) Randomly select an $i_0 \in [1,100]$ and $j_0 \in [1,4]$.
(2) Assign $F[i_0,j_0] = 1 + \Delta k \times R(-1,1)$, where $R(-1,1)$ denotes a random real value between -1 and 1.
(3) For all integers $j \in [1,4]$, assign $F[i_0, j \pm 1] = F[i_0,j] + \Delta z \times R(-1,1)$, starting with j_0.
(4) For all integers $j \in [1,4]$, assign $F[i_0 \pm 1, j] = F[i_0,j] + \delta k \times R(-1,1)$.
(5) Repeat steps 3 and 4 for all $i \in [1,100]$ starting with i_0.

The quantities Δk, Δz and δk denote the maximum initial modification, the maximum difference between neighboring columns and the maximum difference between neighboring rows, respectively. These are free parameters which we set to $\Delta k = 0.1$, $\Delta z = 0.2$, and $\delta k = 0.005$. Our initial modification can therefore be no more than 10%, neighboring z points cannot vary by more than 20% and neighboring k points cannot vary by more than 0.5%.

Steps 1–5 alone generate a filter that is very noisy since there is no smoothing. We thus apply a further step that averages each entry along the k direction over a bin of width N_k [94].

(6) For all $j \in [1,4]$ and $i \in [1,100]$, assign $F[i,j] = \frac{1}{N_k}\sum_{m=0}^{m=N_k} F[i - \frac{N_k}{2} + m, j]$.

Step 6 is then repeated N_s times to further smooth the filter. This leaves us with two additional free parameters: the bin width N_k and the smoothing step iterations N_s. Changing each of these parameters alters the scale in the k direction of the induced features. Finally, the maximum deviation from one in each component of the filter is then given by

$$\begin{aligned}\delta_M = {} & \Delta k + \mathrm{Max}[|4 - j_0|, |j_0 - 4|] \times \Delta z \\ & + \mathrm{Max}[|100 - i_0|, |i_0 - 100|] \times \delta k.\end{aligned} \tag{E2}$$

With the selected values for Δk, Δz and δk, this means that we can have $\delta_M = 25.65$ which corresponds to modifying a power spectrum by 2500%. Any such modification would naturally already be ruled out by observations at extremely high confidence. In order to moderate the imposed modifications in order that they are more likely to be consistent with current constraints we follow an additional procedure. We begin by defining a new filter $\hat{F}[i,j]$ which conforms at some level with current observations of the power spectrum

$$\hat{F}[i,j] = \epsilon[i,j](F[i,j] - 1) + 1, \tag{E3}$$

where $\epsilon[i,j]$ is an array we shall calculate given some observational constraints. In order for a modification around one to be consistent with an observation at a certain confidence level we use the reduced χ^2 statistic $\chi^2_{\rm red}$, defined as

$$\chi^2_{\rm red} = \frac{1}{N_\nu}\sum_{i=1}^{N_\nu} \frac{[P_{\rm random}(k_i) - P_{\rm ref}(k_i)]^2}{\sigma^2(k_i)}, \tag{E4}$$

where $P_{\rm random}$ and $P_{\rm ref}$ are the randomly modified and unmodified power spectra at scale k_i, respectively. The quantity N_ν is the number of degrees of freedom which in this case is the number of k data points so that $N_\nu = 100$ Finally, $\sigma(k_i)$ is the error on the ith k data point. To represent *current* knowledge about the LSS, we model these errors based on recently completed surveys such as the BOSS survey [95]. Assuming the errors are Gaussian we have [64]

$$\sigma_p(k) = \sqrt{\frac{4\pi^2}{k^2 \Delta k V(z)} \times \left(P_{\rm ref}(k) + \frac{1}{\bar{n}(z)}\right)^2 + \sigma^2_{\rm sys}}, \tag{E5}$$

where Δk is the separation between k data points and $P_{\rm ref}$ is taken to be the halofit nonlinear spectrum with the Planck 2018 best fit parameters [92]. We take $V_{\rm eff} = 1.27\ \mathrm{Gpc}^3/h^3$ and $\bar{n} = 5 \times 10^{-4}\ h^3/\mathrm{Mpc}^3$ as an approximate volume and number density of tracers for the BOSS survey, respectively [96]. We also take $\sigma^2_{\rm sys} = 25\ \mathrm{Mpc}^6/h^6$ as our modeling systematic error as in the main text (see Sec. III). Note we do not use the future survey specifications because these modifications should be within current constraints. Substituting Eq. (E5) into Eq. (E4) and using the definition of the filter in Eq. (E1) we get

$$\chi^2_{\rm red}(j) = \frac{1}{N_\nu}\sum_{i=1}^{N_\nu} \frac{(F[i,j] - 1)^2 P_{\rm ref}[i,j]^2 \epsilon[i,j]^2}{\sigma_p^2(k_i)}, \tag{E6}$$

where we perform the calculation for a fixed redshift z_j. Taking the ansatz $\epsilon[i,j] = p(j)\sigma(k_i)$ we can write the unknown $p(j)$ as

$$p(j) = \left[\frac{\chi^2_{\rm red} N_\nu}{\sum_{i=1}^{N_\nu}(F[i,j] - 1)^2 P_{\rm ref}[i,j]^2}\right]^{\frac{1}{2}}. \tag{E7}$$

Once we specify the level of deviation we want from the true spectrum in terms of the $\chi^2_{\rm red}$, we can calculate $p(j)$ for all $j \in [1,4]$. Using Eqs. (E5) and (E3), we can then obtain our constrained random spectrum array [see Eq. (E1)]. We select $\chi^2_{\rm red} \in [1,3]$ to represent modifications which deviate on average up to $\sim 3\sigma$ from the reference spectrum.

Finally, to construct the filters, we assume $P_{\rm ref} = P_{\rm pl}$, i.e., a halofit-generated power spectra with the Planck 2018 best fit cosmology (see Table I). To enhance the

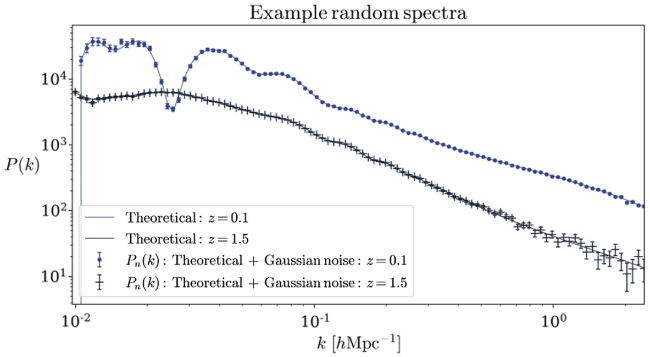

FIG. 16. Example of randomly generated matter power spectra generated using the algorithm outlined in Appendix E. One can see that, while deviations at high k remain small, large deviations are produced at low k due to the poorer constraining power at these scales. The presence of such large deviations generally allows the five-label and two-label BNNs to classify the majority of such spectra at high confidence, enabling them to pick up potential exotic theories that do not fall within the other considered classes in the training set.

randomness of the random spectra, we then apply the filter to a randomly selected reference spectrum in the training dataset, i.e., from the classes $f(R)$, DGP, wCDM and ΛCDM. This assumes the approximation $P_{\rm ref} \approx P_{\rm pl}$ in Eq. (E6). We have a freedom to choose the level of deviation of the random spectra from Planck and we produce many thousand random spectra, making this approximation not of significant consequence to our final results. Further, recall our reference spectra are chosen within a Gaussian distribution centered about the Planck best fits with standard deviations using recent Euclid forecasts [62,92]. Examples of the shape of random power spectra generated using this algorithm can be seen in Fig. 16.

APPENDIX F: COMPARISON WITH MCMC

In this appendix we elaborate further on the comparison and interplay among a classifier based on a BNN such as the one presented in this work and the widely used statistical inference method of MCMC. We already discussed in the main text the fact that a fair comparison between the two methods presents some significant issues and the related reasons. One may still wonder if an MCMC analysis of mock data such as those used for our classification examples would be able to correctly identify the underlying theory and yield the corresponding Bayesian evidence, should one be willing to pay the corresponding much higher computational cost. It turns out that this is not so straightforward. The first issue is that the Gaussian errors associated with the data vector $P(k;z)$ are minute at small scales, making this analysis very sensitive to even the smallest systematic modeling errors. We introduced $\sigma_{\rm sys}$ in Eq. (5) for this very reason. Nevertheless, this error was chosen for a very particular model and data vector, namely ΛCDM with a Planck cosmology, and it is unclear how much modeling inaccuracies and noise would bias other MCMC analyses. Indeed, we explicitly checked this issue by running MCMCs on the examples discussed in Sec. IV D. The resulting constraints show significant biases that make any attempt to compute a Bayesian evidence meaningless. A meaningful result could only be obtained by increasing $\sigma_{\rm sys}$ until the final contours include the fiducial values, a process that is clearly unjustified and not applicable when considering observational data where the true value is not known. These issues indicate the need of a thorough investigation that goes beyond the scope of this paper. Furthermore, in a realistic context where one does not know the actual underlying theory, in order to run an MCMC analysis to constrain extensions to ΛCDM it is necessary to choose an appropriate parametrization which picks up the modification via a deviation from the fiducial parameter values. Typically generic parameters are proposed such as the growth index which serve to account for the presence of physics beyond ΛCDM should they deviate from their fiducial value. A deviation in a non-ΛCDM parameter could result in significant biases in the standard ΛCDM cosmological parameters. Classifiers such as those considered in this work possess an advantage over MCMC in this regard as they do not rely on a well-chosen parametrization to detect deviations, at least at test time. This feature is particularly interesting if we consider the possibility that signatures of new physics may be present in the power spectrum that do not come from any theory for which numerical codes are available. Detecting such deviations with MCMCs may not be possible in the absence of accurate modeling, while the method described in this paper would still be able to provide hints of deviations from ΛCDM.

[1] E. J. Copeland, M. Sami, and S. Tsujikawa, Dynamics of dark energy, Int. J. Mod. Phys. D **15**, 1753 (2006).
[2] T. Clifton, P. G. Ferreira, A. Padilla, and C. Skordis, Modified gravity and cosmology, Phys. Rep. **513**, 1 (2012).
[3] A. Joyce, B. Jain, J. Khoury, and M. Trodden, Beyond the cosmological standard model, Phys. Rep. **568**, 1 (2015).
[4] P. Bull *et al.*, Beyond ΛCDM: Problems, solutions, and the road ahead, Phys. Dark Universe **12**, 56 (2016).

[5] K. Koyama, Cosmological tests of modified gravity, Rep. Prog. Phys. **79**, 046902 (2016).
[6] A. Joyce, L. Lombriser, and F. Schmidt, Dark energy versus modified gravity, Annu. Rev. Nucl. Part. Sci. **66**, 95 (2016).
[7] M. Ishak, Testing general relativity in cosmology, Living Rev. Relativity **22**, 1 (2019).
[8] S. Alam *et al.* (BOSS Collaboration), The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: Cosmological analysis of the DR12 galaxy sample, Mon. Not. R. Astron. Soc. **470**, 2617 (2017).
[9] M. Asgari *et al.* (KiDS Collaboration), KiDS-1000 Cosmology: Cosmic shear constraints and comparison between two point statistics, Astron. Astrophys. **645**, A104 (2021).
[10] R. Laureijs *et al.* (EUCLID Collaboration), Euclid definition study report, arXiv:1110.3193.
[11] v. Ivezić *et al.* (LSST Collaboration), LSST: From science drivers to reference design and anticipated data products, Astrophys. J. **873**, 111 (2019).
[12] M. Knabenhans *et al.* (Euclid Collaboration), Euclid preparation: II. The EuclidEmulator—A tool to compute the cosmology dependence of the nonlinear matter power spectrum, Mon. Not. R. Astron. Soc. **484**, 5509 (2019).
[13] R. Takahashi, M. Sato, T. Nishimichi, A. Taruya, and M. Oguri, Revising the Halofit model for the nonlinear matter power spectrum, Astrophys. J. **761**, 152 (2012).
[14] A. Mead, J. Peacock, C. Heymans, S. Joudaki, and A. Heavens, An accurate halo model for fitting non-linear cosmological power spectra and baryonic feedback models, Mon. Not. R. Astron. Soc. **454**, 1958 (2015).
[15] H. Winther, S. Casas, M. Baldi, K. Koyama, B. Li, L. Lombriser, and G.-B. Zhao, Emulators for the nonlinear matter power spectrum beyond ΛCDM, Phys. Rev. D **100**, 123540 (2019).
[16] A. Mead, C. Heymans, L. Lombriser, J. Peacock, O. Steele, and H. Winther, Accurate halo-model matter power spectra with dark energy, massive neutrinos and modified gravitational forces, Mon. Not. R. Astron. Soc. **459**, 1468 (2016).
[17] G.-B. Zhao, Modeling the nonlinear clustering in modified gravity models. I. A fitting formula for the matter power spectrum of $f(R)$ gravity, Astrophys. J. Suppl. Ser. **211**, 23 (2014).
[18] M. Cataneo, L. Lombriser, C. Heymans, A. Mead, A. Barreira, S. Bose, and B. Li, On the road to percent accuracy: Non-linear reaction of the matter power spectrum to dark energy and modified gravity, Mon. Not. R. Astron. Soc. **488**, 2121 (2019).
[19] B. Bose, M. Cataneo, T. Tröster, Q. Xia, C. Heymans, and L. Lombriser, On the road to percent accuracy IV: ReACT—computing the non-linear power spectrum beyond ΛCDM, Mon. Not. R. Astron. Soc. **498**, 4650 (2020).
[20] J. Kennedy, L. Lombriser, and A. Taylor, Screening and degenerate kinetic self-acceleration from the nonlinear freedom of reconstructed Horndeski theories, Phys. Rev. D **100**, 044034 (2019).
[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, Intriguing properties of neural networks, arXiv:1312.6199.
[22] A. Nguyen, J. Yosinski, and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, arXiv:1412.1897.
[23] Y. Gal and Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, arXiv:1506.02142.
[24] D. J. C. MacKay, A practical bayesian framework for backpropagation networks., Neural Comput. **4**, 448 (1992).
[25] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer-Verlag, Berlin, Heidelberg, 1996).
[26] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, Weight uncertainty in neural networks, arXiv:1505.05424.
[27] T. Charnock, L. Perreault-Levasseur, and F. Lanusse, Bayesian neural networks, arXiv:2006.01490.
[28] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, Hands-on Bayesian neural networks—A tutorial for deep learning users, arXiv:2007.06823.
[29] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, Weight uncertainty in neural networks, arXiv:1505.05424.
[30] H. Shen, E. A. Huerta, E. O'Shea, P. Kumar, and Z. Zhao, Statistically-informed deep learning for gravitational wave parameter estimation, Mach. Learn. Sci. Tech. **3**, 015007 (2022).
[31] Y.-C. Lin and J.-H. P. Wu, Detection of gravitational waves using Bayesian neural networks, Phys. Rev. D **103**, 063034 (2021).
[32] T. L. Killestein *et al.*, Transient-optimised real-bogus classification with Bayesian Convolutional Neural Networks—sifting the GOTO candidate stream, Mon. Not. R. Astron. Soc. **503**, 4838 (2021).
[33] H. J. Hortua, R. Volpi, D. Marinelli, and L. Malagò, Parameter estimation for the cosmic microwave background with Bayesian neural networks, Phys. Rev. D **102**, 103509 (2020).
[34] H. J. Hortua, R. Volpi, D. Marinelli, and L. Malago, Accelerating MCMC algorithms through Bayesian Deep Networks, in *Proceedings of the 34th Conference on Neural Information Processing Systems* (2020).
[35] R. Michelmore, M. Wicker, L. Laurenti, L. Cardelli, Y. Gal, and M. Kwiatkowska, Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control, arXiv:1909.09884.
[36] G. Deodato, C. Ball, and X. Zhang, Bayesian neural networks for cellular image classification and uncertainty analysis, bioRxiv (2020), https://www.biorxiv.org/content/10.1101/824862v2.
[37] D. K. Ramanah, N. Arendse, and R. Wojtak, AI-driven spatio-temporal engine for finding gravitationally lensed supernovae, arXiv:2107.12399.
[38] A. Möller and T. de Boissière, SuperNNova: An open-source framework for Bayesian, neural network-based supernova classification, Mon. Not. R. Astron. Soc. **491**, 4277 (2020).
[39] D. J. C. Mackay, Probable networks and plausible predictions—A review of practical bayesian methods for supervised neural networks, Network **6**, 469 (1995).
[40] D. J. MacKay, Choice of basis for laplace approximation, Mach. Learn. **33**, 77 (1998).
[41] A. Graves, Practical variational inference for neural networks, in *Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11* (Curran Associates Inc., Red Hook, NY, 2011), pp. 2348–2356.

[42] B. Lakshminarayanan, A. Pritzel, and C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, arXiv:1612.01474.
[43] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning, arXiv:1710.07283.
[44] H. Ritter, A. Botev, and D. Barber, A scalable laplace approximation for neural networks, in *International Conference on Learning Representations* (2018).
[45] K. Shridhar, F. Laumann, and M. Liwicki, Uncertainty estimations by Softplus normalization in Bayesian convolutional neural networks with variational inference, arXiv:1806.05978.
[46] Q. Wu, H. Li, L. Li, and Z. Yu, Quantifying intrinsic uncertainty in classification via deep Dirichlet mixture networks, arXiv:1906.04450.
[47] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift, arXiv:1906.02530.
[48] M. Hobbhahn, A. Kristiadi, and P. Hennig, Fast predictive uncertainty for classification with Bayesian deep networks, arXiv:2003.01227.
[49] Y. Fan and S. A. Sisson, Abc samplers, arXiv:1802.09650.
[50] J. Alsing, T. Charnock, S. Feeney, and B. Wandelt, Fast likelihood-free cosmology with neural density estimators and active learning, Mon. Not. R. Astron. Soc. **488**, 5093 (2019).
[51] B. Bose, B. S. Wright, M. Cataneo, A. Pourtsidou, C. Giocoli, L. Lombriser, I. G. McCarthy, M. Baldi, S. Pfeifer, and Q. Xia, On the road to percent accuracy V: The nonlinear power spectrum beyond ΛCDM with massive neutrinos and baryonic feedback, Mon. Not. R. Astron. Soc. **508**, 2479 (2021).
[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016), http://www.deeplearningbook.org.
[53] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to Machine Learning for physicists, Phys. Rep. **810**, 1 (2019).
[54] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).
[55] A. D. Kiureghian and O. Ditlevsen, Aleatory or epistemic? does it matter?, Struct. Safety **31**, 105 (2009), risk Acceptance and Risk Communication.
[56] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation, Computational Statistics and Data Analysis **142**, 106816 (2020).
[57] A. Malinin and M. Gales, Predictive uncertainty estimation via prior networks, arXiv:1802.10501.
[58] W. Hu and I. Sawicki, Models of f(R) cosmic acceleration that evade solar-system tests, Phys. Rev. D **76**, 064004 (2007).
[59] G. Dvali, G. Gabadadze, and M. Porrati, 4-D gravity on a brane in 5-D Minkowski space, Phys. Lett. B **485**, 208 (2000).
[60] M. Chevallier and D. Polarski, Accelerating universes with scaling dark matter, Int. J. Mod. Phys. D **10**, 213 (2001).
[61] E. V. Linder, Exploring the Expansion History of the Universe, Phys. Rev. Lett. **90**, 091301 (2003).
[62] A. Blanchard *et al.* (Euclid Collaboration), Euclid preparation: VII. Forecast validation for Euclid cosmological probes, Astron. Astrophys. **642**, A191 (2020).
[63] H. A. Feldman, N. Kaiser, and J. A. Peacock, Power spectrum analysis of three-dimensional redshift surveys, Astrophys. J. **426**, 23 (1994).
[64] H.-J. Seo and D. J. Eisenstein, Improved forecasts for the baryon acoustic oscillations and cosmological distance scale, Astrophys. J. **665**, 14 (2007).
[65] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, On calibration of modern neural networks, in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, edited by D. Precup and Y. W. Teh, Proceedings of Machine Learning Research Vol. 70 (JMLR, 2017), pp. 1321–1330.
[66] V. Kuleshov, N. Fenner, and S. Ermon, Accurate uncertainties for deep learning using calibrated regression, arXiv:1807.00263.
[67] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, Measuring calibration in deep learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (IEEE, Piscataway, 2019).
[68] P. J. E. Peebles, *Research supported by the National Science Foundation* (Princeton University Press, Princeton, NJ, 1980), p. 435.
[69] L.-M. Wang and P. J. Steinhardt, Cluster abundance constraints on quintessence models, Astrophys. J. **508**, 483 (1998).
[70] E. V. Linder, Cosmic growth history and expansion history, Phys. Rev. D **72**, 043529 (2005).
[71] A. Heavens, Y. Fantaye, A. Mootoovaloo, H. Eggers, Z. Hosenie, S. Kroon, and E. Sellentin, Marginal likelihoods from Monte Carlo Markov chains, arXiv:1704.03472.
[72] A. Peel, F. Lalande, J.-L. Starck, V. Pettorino, J. Merten, C. Giocoli, M. Meneghetti, and M. Baldi, Distinguishing standard and modified gravity cosmologies with machine learning, Phys. Rev. D **100**, 023508 (2019).
[73] M. Troxel *et al.* (DES Collaboration), Dark energy survey year 1 results: Cosmological constraints from cosmic shear, Phys. Rev. D **98**, 043528 (2018).
[74] https://github.com/Mik3M4n/BaCoN.
[75] 10.5281/zenodo.4309918.
[76] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.
[77] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, An introduction to variational methods for graphical models, Mach. Learn. **37**, 183 (1999).
[78] Y. Gal, Uncertainty in deep learning, Ph.D. thesis, University of Cambridge, 2016.
[79] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, Variational inference: A review for statisticians, J. Am. Stat. Assoc. **112**, 859 (2017).
[80] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).

[81] T. S. Jaakkola and M. I. Jordan, Bayesian parameter estimation via variational methods, Stat. Comput. **10**, 25 (2000).
[82] R. Neal and G. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, Learn. Graph. Models **89**, 355 (2000).
[83] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, arXiv:1312.6114.
[84] D. P. Kingma, T. Salimans, and M. Welling, Variational dropout and the local reparameterization trick, arXiv:1506.02557.
[85] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse, Flipout: Efficient pseudo-independent weight perturbations on mini-batches, in *International Conference on Learning Representations* (2018).
[86] https://www.tensorflow.org/probability/overview.
[87] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, TensorFlow distributions, arXiv:1711.10604.
[88] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.
[89] This and other relations used in this section are a consequence of the fact that Eq. (2) is the covariance matrix of a multinomial distribution with parameters μ_i.
[90] C. S. Withers and S. Nadarajah, The spectral decomposition and inverse of multinomial and negative multinomial covariances, Braz. J. Probab. Stat. **28**, 376 (2014).
[91] M. Cataneo, D. Rapetti, F. Schmidt, A. B. Mantz, S. W. Allen, D. E. Applegate, P. L. Kelly, A. von der Linden, and R. G. Morris, New constraints on $f(R)$ gravity from clusters of galaxies, Phys. Rev. D **92**, 044009 (2015).
[92] N. Aghanim *et al.* (Planck Collaboration), Planck 2018 results. VI. Cosmological parameters, Astron. Astrophys. **641**, A6 (2020).
[93] For model parameter definitions we refer the reader to Appendix C of Ref. [19].
[94] Note that we pad the edges of the array with additional values using steps (3) and (4) to smooth the boundaries.
[95] http://www.sdss3.org/.
[96] A. J. Cuesta *et al.*, The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: Baryon Acoustic Oscillations in the correlation function of LOWZ and CMASS galaxies in Data Release 12, Mon. Not. R. Astron. Soc. **457**, 1770 (2016).