Constructing Non-Linear Cosmological Tension Coordinate with Neural Networks

Candidate Number: 8260Q
(Dated: May 16, 2021)

Tensions in cosmology, such as the Hubble tension and the weak lensing Ω_m - σ_8 tension, are rising as measurement precision has improved over recent years. We develop the idea of a one-dimensional coordinate which maximises tension between two cosmological datasets. We use a simple neural network with a single hidden layer to represent a non-linear function of cosmological parameters, whose output is the 1D *tension coordinate*. We apply this method to the *Planck* and DES datasets on the following parameters: baryon density $\Omega_b h^2$, matter density Ω_m, Hubble constant H_0, optical depth to reionisation τ, matter fluctuation amplitude σ_8, and scalar spectral index n_s. Across all six parameters, we obtain a marginalised Bayes factor of $\log R_t = -16.8 \pm 0.5$, which is equivalent to a non-linear tension of $(5.47 \pm 0.08)\sigma$. Calculating the gradient of the tension coordinate with respect to each parameter, we find the Hubble constant H_0 to be the largest contributor to the six-parameter tension. Performing a comparison between parameter pairs using a neural network with two input parameters, we again find the Hubble constant to contain the most tension. This is in contrast to traditional approaches, which usually identifies Ω_m - σ_8 to contain the majority of the tension.

I. INTRODUCTION

With cosmological measurements becoming more precise over recent years, disagreements between different datasets and methods have begun to emerge. Observations of parameters surrounding the cosmological constant cold dark matter (ΛCDM) model have yielded discrepancies, more commonly referred to as *tensions*, of close to 5σ – the gold standard of significance in particle physics [1].

One such tension is the *Hubble tension*. The debate over the Hubble constant's value is one that is hardly new, but in recent years has risen to prominence in cosmology. Disagreement over the Hubble constant began between de Vaucouleurs and Sandage in the 1980s [2, 3], and it has now developed into an area of contention between early- and late-universe cosmologists [4–8]. As it stands, measurements by these two factions are at significant tension of around 5σ at the most extreme, as shown in Figure 1. This has earned the Hubble tension an apt label of a cosmological *crisis* [9].

In addition to the Hubble constant, less severe tensions also exist. Discrepancies of 3σ have been reported with respect to the matter density Ω_m and rate of growth of structure σ_8, between the Cosmic Microwave Background (CMB) data collected by *Planck* and the weak lensing-based surveys, namely the Kilo Degree Survey (KiDS) [10] and the Dark Energy Survey (DES). Similarly, there has been arguments made for the existence of a "curvature tension", with inconsistencies of 2.5σ to 3σ between CMB data alluding to a closed curved universe and the tenet of flat curvature in ΛCDM cosmology [11].

These tensions raise questions surrounding the validity of the well-established, well-tested standard cosmological model – the ΛCDM. Are these tensions just an artefact of systematic errors from collecting and analysing datasets, or do these tensions hint at something more fundamental – perhaps a modification to the standard model, or more excitingly new physics to take the place of the old one?

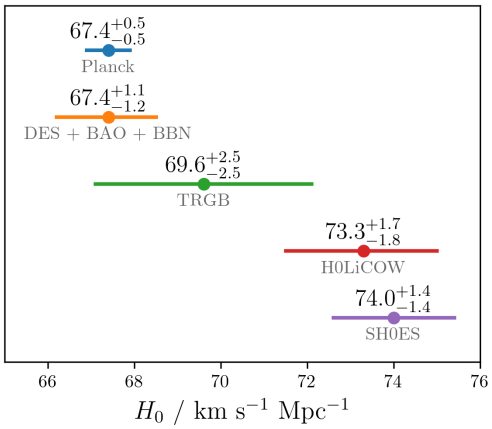

FIG. 1. A compilation of recent measurements of the Hubble constant H_0. The top two measurements are from early-universe datasets using ΛCDM cosmology [4, 5], while the remaining three are from late-universe datasets based off local distance ladder measurements [6–8]. The tension between the *Planck* and SH0ES measurements currently stands at 4.7σ.

However, before we make that leap into the realm of new physics, it is essential for us to examine how tension is quantified. With cosmological datasets being multi-dimensional, the problem of quantifying discrepancies is non-trivial. Datasets that appear to be in mild tension, such as the Dark Energy Survey (DES) Y1 and Planck datasets, have been reported to be consistent when using the canonical Bayes factor R [12]. This is troubling, and is a reflection of the difficulty of the problem. With tensions likely to increase as measurement precision increases, a variety of tension metrics have been proposed in recent literature [13].

This paper aims to develop the idea of coordinates of maximum tension. With cosmological datasets, larger tensions often exist across multiple parameters rather than within each parameter on its own. A good example

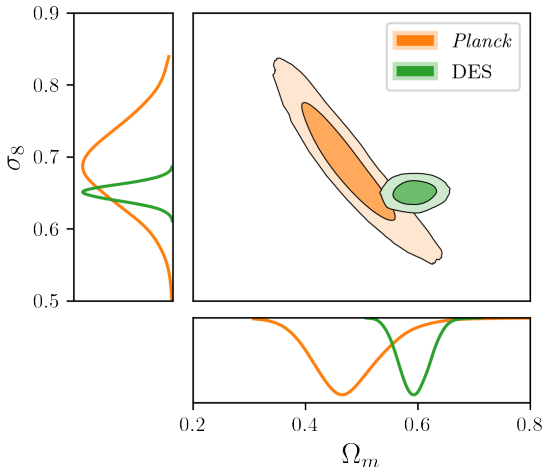

FIG. 2. The main plot shows a two-dimensional marginalised distribution of the *Planck* and DES datasets in the Ω_m - σ_8 plane, where the solid contours contain 68% and 95% of the marginalised probability mass. The two smaller plots are the one-dimensional marginalised distributions of both datasets in each parameter.

would be the 3σ tension between Ω_m and σ_8 – the tension is obvious in a two-dimensional plot between these two parameters, but is not as significant when the parameters are inspected individually, as illustrated in Figure 2. In a high-dimensional parameter space, it is thus likely that there exists a hidden combination of parameters which exacerbates and maximises tension.

In this paper, we explore a non-linear function of cosmological parameters for which the marginalised tension is maximised. In theory, this function can be represented by any parameterised function. We choose to use a neural network to achieve this mapping, since the non-Gaussian nature of certain cosmological parameters renders an analytical approach challenging. This tension coordinate is then applied to the *Planck* and DES Y1 datasets. Such an approach could allow us to develop a better intuition of the source of tension, and verify the large tensions that currently exist in H_0 and the Ω_m– σ_8 plane.

II. BACKGROUND

A. Bayesian statistics

We use the following notation of Bayesian statistics, with Bayes' theorem written as

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \longrightarrow \mathcal{P}_D(\theta) = \frac{\mathcal{L}_D(\theta)\pi(\theta)}{\mathcal{Z}_D}. \quad (1)$$

The posterior is denoted as $\mathcal{P}_D$, likelihood as $\mathcal{L}_D$, prior as π, and evidence as $\mathcal{Z}_D$. Note that the subscript here represents the dataset-dependence of these distributions.

1. *Bayesian evidence*

The Bayesian evidence $\mathcal{Z}_D$ is a normalisation constant determined by the likelihood and prior to be

$$\mathcal{Z}_D = \int \mathcal{L}_D \pi \, \mathrm{d}\theta. \quad (2)$$

Also known as the marginal likelihood in statistical literature [14], the evidence is often deemed to be a cornerstone quantity for model and dataset comparisons within the Bayesian framework.

B. Bayes factor R

The canonical Bayes factor R [15, 16] is our tension metric of choice, and forms the basis of our method in this paper. With two datasets A and B, the Bayes factor is expressed as

$$R = \frac{\mathcal{Z}_{AB}}{\mathcal{Z}_A \mathcal{Z}_B}, \quad \text{with } \mathcal{Z}_{\mathrm{AB}} = \int \mathcal{L}_{\mathrm{A}} \mathcal{L}_{\mathrm{B}} \pi \, \mathrm{d}\theta. \quad (3)$$

This factor is defined as the ratio between the probability of A and B describing the same parameter space and the probability of A and B describing separate parameter spaces. A value of $R \gg 1$ indicates concordant datasets, and $R \ll 1$ indicates discrepant datasets.

The Bayes factor can be re-expressed as

$$R = \int \frac{\mathcal{P}_A \mathcal{P}_B}{\pi} \, \mathrm{d}\theta. \quad (4)$$

This is a more convenient and useful expression for us to work with, since we work with posterior and prior samples in this paper. This form of the Bayes factor also makes explicit its prior-dependence, which is arguably a concern [12]. A broader prior would increase the factor R, whilst a narrower prior would reduce R and thus increase tension. Such a dependence means that R is able to hide potential discrepancies with a broad prior. On the other hand, consistent datasets with sensible priors *cannot* be reported to be in tension by the Bayes factor. This means that a value of R indicating tension does in fact point to discrepant datasets, hence low values of R must be taken more seriously than higher values of R.

C. Tension coordinate

Many tension quantification methods, including the Bayes factor, have the desirable characteristic of being independent of the choice of parameters and the direction in the parameter space. However, it would also be interesting to be able to describe the spatial structure of the tension, i.e. which parameters contribute most to the tension. As alluded to in the introduction of this paper, it is very probable that there exists a combination of parameters, or more generally a function of the

parameter space, which maximises tension between two datasets. The output of such a function is what we call the *tension coordinate*.

In this paper, tension is quantified using the Bayes factor R. We minimise R, calculated in the tension coordinate, to achieve maximum tension. We first demonstrate a linear tension coordinate for the ideal case of two Gaussian distributions representing two separate datasets. Then, we approach the tension coordinate more practically to tackle non-Gaussian distributions.

1. Gaussian example

Let there be two datasets A and B represented by two Gaussian distributions with shared parameters θ. A and B have mean vectors of μ_A and μ_B, and covariance matrices of Σ_A and Σ_B, respectively. The natural log of the Bayes factor R between these two distributions is written as [12]

$$\log R = -\frac{1}{2}(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}(\mu_A - \mu_B) \quad (5)$$

$$= -\frac{1}{2}\mu^T\Sigma^{-1}\mu \quad (6)$$

where $\mu = \mu_A - \mu_B$ and $\Sigma = \Sigma_A + \Sigma_B$.

Define the one-dimensional tension coordinate as a linear combination of the parameters, $t = n^T\theta$. The vector n describes the combination of parameters which maximises tension. To map the Gaussian distributions onto t, we marginalise the distributions onto the hyperplanes perpendicular to n. This gives a marginalised Bayes factor of

$$\log R_t = -\frac{1}{2}\left(n^T\mu\right)^T\left(n^T\Sigma n\right)^{-1}\left(n^T\mu\right) \quad (7)$$

$$= -\frac{\left(n^T\mu\right)^2}{2n^T\Sigma n}. \quad (8)$$

Minimising $\log R_t$ with respect to n, which is equivalent to maximising the marginalised tension, gives the direction of maximum tension as

$$n \propto \Sigma^{-1}\mu = (\Sigma_A + \Sigma_B)^{-1}(\mu_A - \mu_B). \quad (9)$$

Substituting this back into the tension coordinate returns

$$t = n^T\theta \propto (\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}\theta \quad (10)$$

where t is defined up to a normalisation constant.

2. Non-Gaussian case

Datasets with non-Gaussian distributions bring more complexity to the tension coordinate. Instead of marginalising the distributions onto hyperplanes, it is more general to marginalise the distributions onto hypersurfaces and define a tension coordinate of $t = T(\theta)$. Note that it is not necessary for t to be one-dimensional, but we restrict ourselves to this in this paper. With this definition of the tension coordinate, the marginalised Bayes factor R_t is expressed as

$$R_t = \int \frac{\mathcal{P}_A^t(t)\mathcal{P}_B^t(t)}{\pi^t(t)}\ \mathrm{d}t, \quad (11)$$

$$\mathcal{P}_D^t(t) = \int \mathcal{P}_D(\theta)\delta(t - T(\theta))\ \mathrm{d}\theta. \quad (12)$$

In general, we can define a more practical tension coordinate $t = T(\theta; w)$ described by function parameters w. To obtain maximum tension, the function parameters need to be $w = \arg\min_w R_t$. We are free to choose any type of function for T, where in the simplest case can be represented by a linear combination of parameters, as demonstrated in Section II C 1. We choose to represent T as a neural network with a single hidden layer, with w as the weights between nodes and R_t as the loss function for gradient descent. A neural network is an appropriate choice given the *universal approximation theorem*, which states that a multilayer feedforward network is able to approximate any bounded continuous function [17]. To replicate the simplest case of a linear combination of parameters as in Section II C 1, we can choose to use a neural network without any hidden layers.

III. METHOD

A. Neural network and training

In this paper, we use a fully-connected neural network with only a single hidden layer with 4096 nodes, as illustrated in Figure 3. We find that a single hidden layer is sufficient to fit our maximum tension hypersurfaces around continuous non-Gaussian distributions. The choice of the number of nodes is somewhat arbitrary. Given that this neural network is not used for classification/regression and does not have separate training and testing datasets, we are less concerned with the problem of over-fitting our datasets, hence the large number of nodes.

The inputs to this neural network are the samples from two discrepant datasets with the relevant cosmological parameters, and the output node is the 1D tension coordinate. Due to its effectiveness and widespread proven usage in modern deep learning applications [18], a rectified linear unit (ReLU) [19] is chosen as the activation function between the hidden layer and output layer. We can now express our neural network as

$$t = T(\theta; w) = \sum_i w_i^{(2)}\ \mathrm{ReLU}\left(\sum_j w_{ij}^{(1)}\theta_j + b_i\right) \quad (13)$$

where we have cosmological parameters θ_j, neural network weights $w_{ij}^{(l)}$, and bias b_i. As mentioned in Section

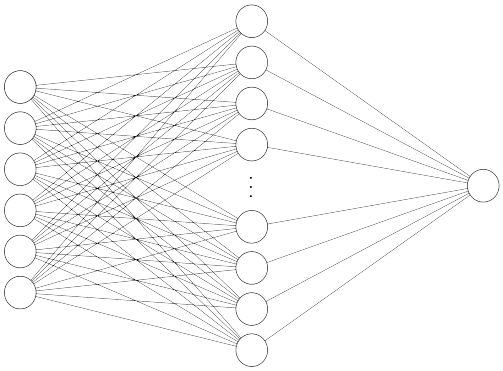

FIG. 3. A fully-connected neural network with 6 nodes in the input layer, an arbitrary number of nodes in the hidden layer, and a single node in the output layer.

II C 2, the loss function is the Bayes factor calculated in the 1D tension coordinate.

Training this neural network is different from training one for a classification problem. Unlike the 'traditional' method of approximating the gradient using small batches of training data [20], we use all the data points of the two datasets to compute our gradient, and hence calculate the loss function. This is done to provide the Bayes factor, which is our loss function, with a substantial amount of data points to better quantify tension. However, we note that no attempts were made in this paper to experiment with a smaller batch size, e.g. half the data points, thus we cannot confirm that an optimal method was used here. We use the recently popularised Adam optimisation algorithm [21] as our stochastic gradient descent method, with an initial learning rate of 10^{-4}.

B. Numerical calculation of Bayes factor

The calculation of the marginalised Bayes factor R_t forms part of the neural network's optimisation loop. This requires the speed of the numerical computation of R_t to be as rapid and efficient as possible. More importantly, the gradient-based optimisation method chosen necessitates the calculation of R_t to be smooth and differentiable. This rules out the use of binning methods that rely on square top-hat functions. Thus to numerically calculate R_t, we rely on a non-parametric smoothing method known as the *kernel density estimator* [22, 23] to estimate the marginalised posteriors and prior in Equation (11). This method uses (weighted) samples of distributions to makes these approximations.

For an arbitrary one-dimensional weighted dataset $\{(x_i, w_i),\, i = 1, \ldots, n, \sum_i^N w_i = 1\}$, the kernel density estimator is expressed as

$$\hat{f}_h(x) = \frac{1}{h}\sum_{i=1}^{n} w_i K\left(\frac{x - x_i}{h}\right) \tag{14}$$

where K is the kernel function and h the bandwidth parameter [22]. The kernel function has to satisfy $\int_\infty^\infty K(x)\,\mathrm{d}x = 1$, thus we pick the kernel to be a Gaussian given by

$$K(x) = \frac{1}{\sqrt{2\pi h^2}} e^{-x^2}. \tag{15}$$

Selecting the bandwidth parameter is more complicated. A review of the various choices of bandwidth parameters and bandwidth selection methods can be found in [24]. Given our choice of a Gaussian kernel, we choose the general rule of thumb, also known as Scott's rule, as our bandwidth. This rule is written as $h = n_{\text{eff}}^{d+4}$, where $n_{\text{eff}} = (\sum w_i)^2/\sum {w_i}^2$ is the effective number of data points and d is the number of dimensions of the distribution being estimated. This setup allows us to effectively approximate unimodal distributions, which the majority of cosmological datasets are.

We use this kernel density estimator to provide an estimate of the marginalised posteriors and the prior found in Equation (11), which can be denoted by $\hat{\mathcal{P}}_A^t$, $\hat{\mathcal{P}}_B^t$ and $\hat{\pi}^t$ respectively. This gives a marginalised Bayes Factor of

$$R_t \approx \sum_{j=1}^{M} \frac{\hat{\mathcal{P}}_A^t(t_j)\hat{\mathcal{P}}_B^t(t_j)}{\hat{\pi}^t(t_j)}\Delta t, \tag{16}$$

where we sample the posteriors and prior at points t_j in the 1D tension coordinate at regular intervals, separated by width Δt.

In practice, it is more effective to use $\log R_t$ instead of R_t as our loss function for the neural network. If we were to use R_t, the optimisation steps taken by the neural network becomes increasing small and inconsequential as $R_t \to 0$. On the other hand, $\log R_t$ with its range of $(-\infty, \infty)$ allows gradient descent to make bolder steps towards the local minimum. Taking the natural log of the Bayes factor also prevents underflow issues for extremely discrepant datasets ($R_t \to 0$) by exploiting the LogSumExp method to calculate the posteriors, prior, and ultimately $\log R_t$.

C. Toy examples

We use `PyTorch` to construct our neural network, and hence also use it to numerical calculate the Bayes factor. Algorithm 1 takes us through the training of our neural network using `PyTorch`-esque language. Note that the learning rate of the optimisation method and number of epochs can be adjusted in order to reach a stable local minimum of the Bayes Factor R_t.

Algorithm 1 Training of Neural Network

$X_A \leftarrow$ Dataset A; $w_A \leftarrow$ Weights A
$X_B \leftarrow$ Dataset B; $w_B \leftarrow$ Weights B
$X_\pi \leftarrow$ Prior; $w_\pi \leftarrow$ Prior Weights

$net \leftarrow$ NeuralNetwork(in=6, hidden=4096, out=1)
$optim \leftarrow$ AdamOptimizer($net.parameters()$, learning_rate=10^{-4})
$loss \leftarrow$ LogBayesFactor()
$epochs \leftarrow 500$

for $i \leftarrow 0$ to $epochs$ **do**
    $X_A^t \leftarrow net(X_A)$ ▷ Samples in tension coordinates
    $X_B^t \leftarrow net(X_B)$
    $X_\pi^t \leftarrow net(X_\pi)$

    $R_t \leftarrow loss(X_A^t, X_B^t, X_\pi^t, w_A, w_B, w_p)$
    $R_t.backward()$ ▷ Compute gradients of neural network parameters
    $optim.step()$ ▷ Make gradient descent step
end for

return net

Before applying the neural network onto cosmological datasets, we first verify our method using toy datasets. We generate 10000 samples from each of the distributions described below to create our toy datasets.

The simplest case to begin with would be two discrepant 2D Gaussian distributions, that are also discrepant in each dimension individually. In our example, we place these two distributions around 5σ apart, and encase them in a square prior, as seen in Figure 4(a). Our neural network is able to fit rough hyperplanes perpendicular to a line connecting the two distributions. This result agrees with the analytical tension coordinate for the Gaussian case derived in Section II C 1.

The next example can be seen in Figure 4(b). We are still working in two dimensions, but now we have a Gaussian distribution accompanied by a concave quarter-circle distribution. These two distributions are discrepant in two dimensions, but have a considerable overlap in their 1D marginalised distributions. We use this albeit exaggerated example to reinforce the notion that multi-dimensionality is able to hide non-linear tensions. With this setup, we expect the hypersurfaces, or more simply the contour lines, to be shaped around the 'banana-like' distribution, with directions of maximum tension radiating outwards from the central Gaussian distribution. This is exactly what we see from our results, as illustrated in Figure 4(b).

It is crucially important to note that the statistical power of our neural network only lies in the region between our two distributions, and begins to falter as we navigate further away. This effect is obvious in both of our toy examples. In the first case, we see the contour lines bending in the bottom-right and upper-left corners of the plot. In the second example, the curve of the contours do not exactly follow the outer curve of the 'banana-shaped' distribution as we would have expected if the neural network was able to extrapolate well.

D. Cosmological dataset

In this paper, we examine the discrepancy between the cosmological datasets collected by the DES and the *Planck* satellite. We choose these two datasets because of the well reported tensions between them in recent years [12, 25]. The tensions mentioned in Section I of this paper, such as the 3σ tension in the Ω_m - σ_8 plane, also exist between these two datasets.

We use the following six cosmological parameters – baryon density $\Omega_b h^2$, matter density parameter Ω_m, Hubble constant H_0, optical depth to reionisation τ, matter fluctuation amplitude σ_8, and scalar spectral index n_s. The derived parameters of Ω_m, H_0 and σ_8 are chosen because tensions in these parameters are well-known. They take the place of the three other independent cosmological parameters of dark matter density $\Omega_c h^2$, angular acoustic scale $100\theta_*$ and logarithmic primordial fluctuation amplitude $\log A_s$. Applying our neural network to these parameters with established discrepancies allows us to verify and reinforce the existing tension in them.

We obtain the DES Y1 and *Planck* datasets from [26], and use the wide prior from the DES dataset as the shared prior. This source provides nested sampling chains for all of the independent, derived and nuisance parameters of both datasets. We use the `anesthetic` `Python` library [27] to extract weighted samples from the nested sampling chains. We normalise the datasets by dividing each sample by the maximum value in both posteriors, such that the DES and *Planck* posteriors lie in the range of $[0, 1]$ for each parameter. This normalisation allows our neural network to learn and converge faster to a minimum.

E. Identifying the source(s) of tension

1. Six parameters

We first train our neural network on all six cosmological parameters to produce a 1D tension coordinate. This gives us an initial intuition of the maximum tension that can be formed between the DES and *Planck* posteriors in these six parameters. To evaluate the contribution of each parameter to the tension, we calculate the derivative of the 1D tension coordinate t with respect to the six parameters. Performing a Taylor expansion on t

$$t = t_0 + \Delta\theta\nabla t + \mathcal{O}((\Delta\theta)^2), \tag{17}$$

we would expect a parameter with a larger absolute gradient to be more influential in defining the tension coordinate, hence contributing more to the tension. It is important to remind ourselves that our DES and *Planck* posteriors have been normalised to a range of [0, 1], hence

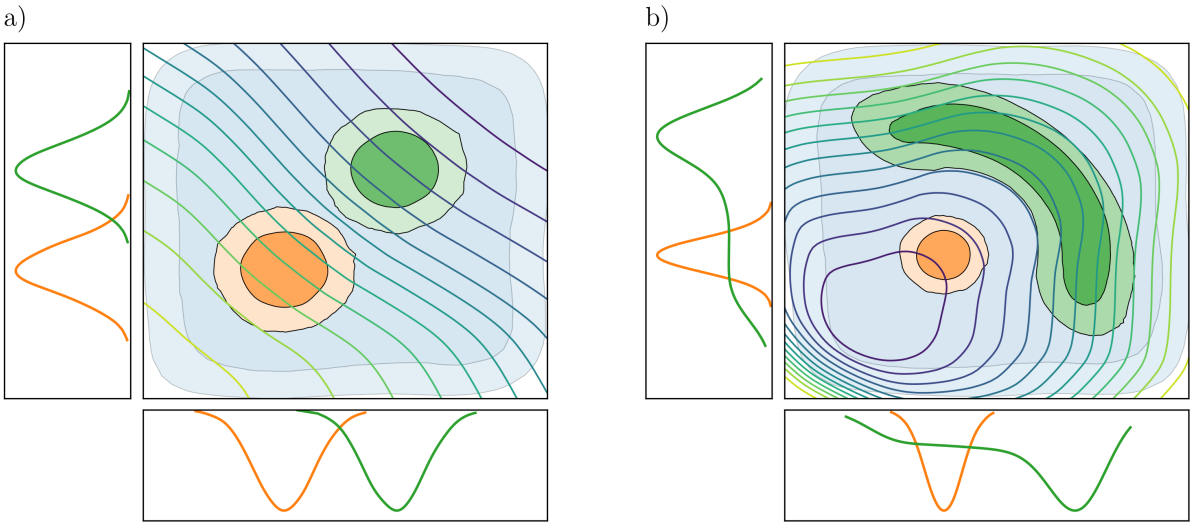

FIG. 4. The main contour plots illustrate the iso-tension coordinate hypersurfaces shaped by the neural network after training for 500 epochs on our toy examples. a) shows two Gaussian distributions and b) shows a Gaussian distribution accompanied by a 'banana-shaped' distribution. The faded blue distribution in the background is the prior. The smaller plots along the x and y axes show the marginalised distributions of these toy examples in their respective axes.

such a comparison of gradients is dimensionally consistent.

2. *Pair-wise comparison*

To verify our results from training our neural network on all six parameters, we make pair-wise comparisons between the six cosmological parameters. We construct separate neural networks with two input nodes for each of the fifteen combinations of parameter pairs. This allows us to rank parameter pairings in terms of their contribution to the discrepancy between the two posteriors. This will also enable us to identify parameters containing the most/least tension, and compare the results obtained via this method with the method of calculating gradients from the six-parameter tension.

F. Standardising the tension coordinate

Performing training repetitions on the same neural network architecture with identical data samples often yields 1D tension coordinates with varying ranges. This happens likely because of the tension coordinate only being defined up to a monotonic reparameterisation, resulting in degenerate minima. This renders comparison between training runs difficult. It is thus important to transform the tension coordinate to one that is standardised. We describe the transformation we perform in Appendix A.

G. Computing a more interpretable tension

The marginalised Bayes factor is a good method to identify tension. However, it is arguably less interpretable compared to using σ. To get from the output of the neural network to a σ value, we follow the steps detailed in [28].

We first compute the Suspiciousness statistic S_t between the two posteriors, which is expressed as

$$S_t = \frac{R_t}{I_t}, \quad \text{where } I_t = \frac{\mathcal{D}_A^t \mathcal{D}_B^t}{\mathcal{D}_{AB}^t}, \tag{18}$$

where $\mathcal{D}_d$ is the Kullback-Leibler divergence, as detailed in Appendix B. The subscript and superscript t reiterate that we are working in the 1D tension coordinate. The numerical calculation of S_t can be done in a very similar manner to the marginalised Bayes factor R_t using kernel density estimators, as described in Section III B. We may then approximate our two posteriors in the 1D tension coordinate with a Gaussian (which is a reasonable assumption, given Figure 5), and arrive at an expression of

$$\log S = \frac{d}{2} - \frac{\chi^2}{2}, \tag{19}$$

$$\chi^2 = (\mu_A - \mu_B)(\Sigma_A + \Sigma_B)^{-1}(\mu_A - \mu_B), \tag{20}$$

where d is the Bayesian combined model dimensionality [12] of the two datasets, and μ_D and Σ_D are the means and covariances of the two marginalised posteriors. χ^2 here follows a chi-squared distribution. This is all derived in [12].

We can obtain a p value via the probability density

	Gaussian assumption	Neural network
$\log R_t$	-0.9	-16.8 ± 0.5
$\log S_t$	-2.4	-17.9 ± 0.5
p value	6.6%	$(4.6 \pm 2.2) \times 10^{-6}$%
tension	1.84σ	$(5.47 \pm 0.08)\sigma$

TABLE I. The quantities related to the six-parameter marginalised tension obtained using two methods – the Gaussian assumption detailed in Section II C 1, and our neural network approach. There are no errors for the Gaussian assumption method because the tension coordinate can be constructed analytically.

function of the chi-squared distribution

$$p = \int_{d-2\log S}^{\infty} \frac{x^{d/2-1}e^{-x/2}}{2^{d/2}\Gamma(d/2)}\,\mathrm{d}x \tag{21}$$

and convert this to a σ value using the inverse error function

$$\sigma(p) = \sqrt{2}\mathrm{erf}^{-1}(1-p). \tag{22}$$

The final conversion is possible because we approximate the marginalised posteriors as Gaussians.

We choose to use a value of $d \approx 4$ for the DES and *Planck* datasets, as presented in Table II from [12], since we source our data from the same paper [26].

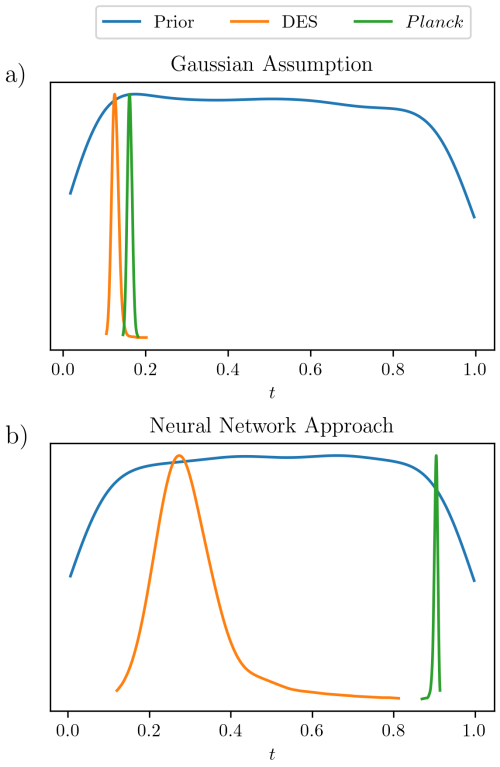

FIG. 5. Density plots of the marginalised posteriors of the DES and Planck posteriors and the prior, in the 1D tension coordinate. (a) uses the Gaussian assumption described in Section II C 1 and (b) uses the neural network approach.

IV. RESULTS AND DISCUSSION

A. Six parameters

We train our neural network for 500 epochs, starting off with a learning rate of 10^{-4} for our Adam optimiser and dropping to 10^{-5} as we approach $\log R_t \approx -8$. We find that this setup gives a quick and somewhat stable gradient descent. We perform 10 training repetitions to give a rough estimate of the lowest possible marginalised Bayes factor.

The results from the 10 training runs are detailed in Table I, accompanied by the tension calculated from the coordinate derived from the Gaussian assumption from Section II C 1. The significant difference between these two values gives an initial impression of the neural network's ability to identify a tension coordinate which maximises tension. The density plots of the marginalised posteriors drawn in Figure 5 further demonstrates the neural network's capability of uncovering the discrepancy between the two datasets, bringing the tension to beyond 5σ.

The neural network's effectiveness in capturing the non-linearity of the tension between the two datasets is illustrated in Figure 7. In these 2D marginalised plots, we observe how our method is able to carve out hypersurface contours isolating the narrower posterior of the *Planck* dataset. For some of the plots, including the $\Omega_b h^2$ - τ and σ_8 - τ plots, the contours go to the extent of encircling the *Planck* posterior. The plots between the tension coordinate and cosmological parameters in the final row of the same figure also begin to describe the contribution of each parameter to the six-parameter tension. However, the wide constraints of the DES dataset renders an analysis via correlation coefficients inconclusive.

Before we go further, it is important to reiterate the fact that many degenerate minima exist in our loss function landscape. This implies that there are potentially a multitude of forms in which the iso-tension coordinate hypersurfaces can be drawn. Fortunately, upon further inspection, we note that the contour plots across all 10 training runs look similar, particularly in the region between the two posteriors. Figure 7 can be said to be representative of all the training runs, however one should still be hesitant in drawing conclusion solely from this figure.

To identify the parameters which contribute most to the six-parameter tension, we calculate the gradient of the tension coordinate with respect to the input parameters. We use `PyTorch`'s `torch.autograd` differentiation engine to help us do this, where we obtained the derivative of the tension coordinate with respect to each parameter at each data point. The mean and standard deviation of the absolute gradients for each parameter are

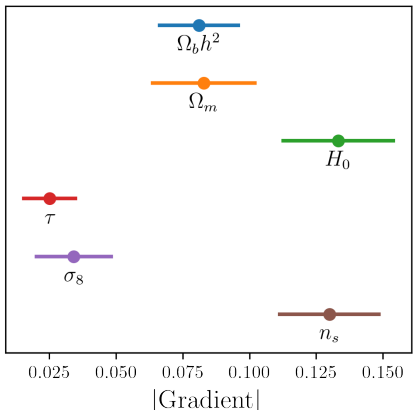

FIG. 6. Measured absolute gradient of the tension coordinate t with respect to each parameter across all data points of both datasets, provided alongside the standard deviation of the measurements

plotted in Figure 6.

Starting from the top, we find H_0 to have the largest gradient, and by extension contribute most to the six-parameter tension. This lends support to the well-established tension of H_0 within the cosmology community. It is surprising to see n_s coming in at a close second after the Hubble constant. With the *Planck* posterior falling squarely within the unconstrained DES posterior in the n_s parameter (as inferred from the bottom-rightmost 2D plot in Figure 7), we would have expected n_s to be one of the least influential parameters. However, the apparent importance of n_s does remind us of the aim of the neural network – to search for non-linear tensions. This result could potentially point to n_s being a significant contributor to the non-linear tension.

The parameters of $\Omega_b h^2$ and Ω_m have almost identical absolute gradients, where they both contribute moderately to the six-parameter tension. σ_8 and τ bring up the rear, contributing the least to the tension. It is slightly unexpected for the former to be considered less influential, given this parameter is part of the Ω_m - σ_8 tension, as discussed in Section I. The apparent insignificance of τ is more in line with what we would expect, unlike n_s, since the DES dataset does not put any constraints on this parameter.

B. Pair-wise comparison

We train our neural network for 500 epochs with parameter pairs as the input, and maintain a learning rate of 10^{-4}. We perform 10 training repetitions for each parameter pair. The results are given in Table II, and pair-wise contour plots are drawn in Figure 8. It is important to note here that there is a nuanced distinction between the results of this section and the previous section. Here, we seek for parameters *containing* the most tension via parameter pairs, whilst in the previous section we attempted to identify parameters which *contributed* most to the tension in the six-dimensional parameter space.

The outcome here generally correlates well with the results from the previous section in terms of the parameter importance. We find H_0 to again contain the most tension by a considerable margin, with a mean two-parameter tension of $(2.47 \pm 0.08)\sigma$. We now have three parameters – $\Omega_b h^2$, Ω_m and σ_8, rather than two, containing similar moderate tension. However, the largest difference with our previous result is the ranking of n_s – the parameter has dropped from second place to last and is now considered to contain the least tension. The contrast in results could potentially point to n_s being a more significant contributor to non-linear tensions in high-dimensional parameter spce, but less able to influence the tension coordinate in lower dimensions.

Looking at the ranking of parameter pairs, it is interesting to note that Ω_m - σ_8 is ranked as the 3rd least discrepant pair with a marginalised tension of around 2σ. This is considerably less than the well-known 3σ tension calculated in two dimensions. It appears that by considering non-linear tension coordinates, more tension can be found in other pairs of parameters, particularly those that consist of either H_0 or $\Omega_b h^2$.

V. CONCLUSIONS

In this paper, we have constructed a basic functioning neural network with an appropriate loss function, in the form of the Bayes factor, to seek out non-linear cosmological tensions. We applied this neural network to the six parameters of $\Omega_b h^2$, Ω_m, H_0, τ, σ_8 and n_s from the DES and *Planck* datasets, and were able to isolate a non-linear tension of $(5.47 \pm 0.08)\sigma$ – a discrepancy surpassing the 5σ gold standard of scientific discovery.

We identified the source of tension between the DES and *Planck* datasets via two methods – the first uses the gradient of the tension coordinate with respect to the cosmological parameters, and the second compares the tension found in pairs of parameters. We found the Hubble constant H_0 to be the largest *contributor* to the six-parameter tension, and also the parameter *containing* the most tension, both by considerable margins.

This result pinpoints the constraints of the Hubble constant H_0 to be the key source of discrepancy between the DES and *Planck* datasets. The weak lensing tension of the Ω_m - σ_8 pair discussed in Section I was surprisingly one of the parameter pairs which contained the least non-linear tension. This could be an indication of the weak lensing tension being a manifestation of the Hubble tension rather than an intrinsic tension within Ω_m - σ_8, especially given these are derived parameters which are not independent.

A surprising observation in our results shows the n_s parameter as a significant contributor to the non-linear six-parameter tension, but does not contain much ten-

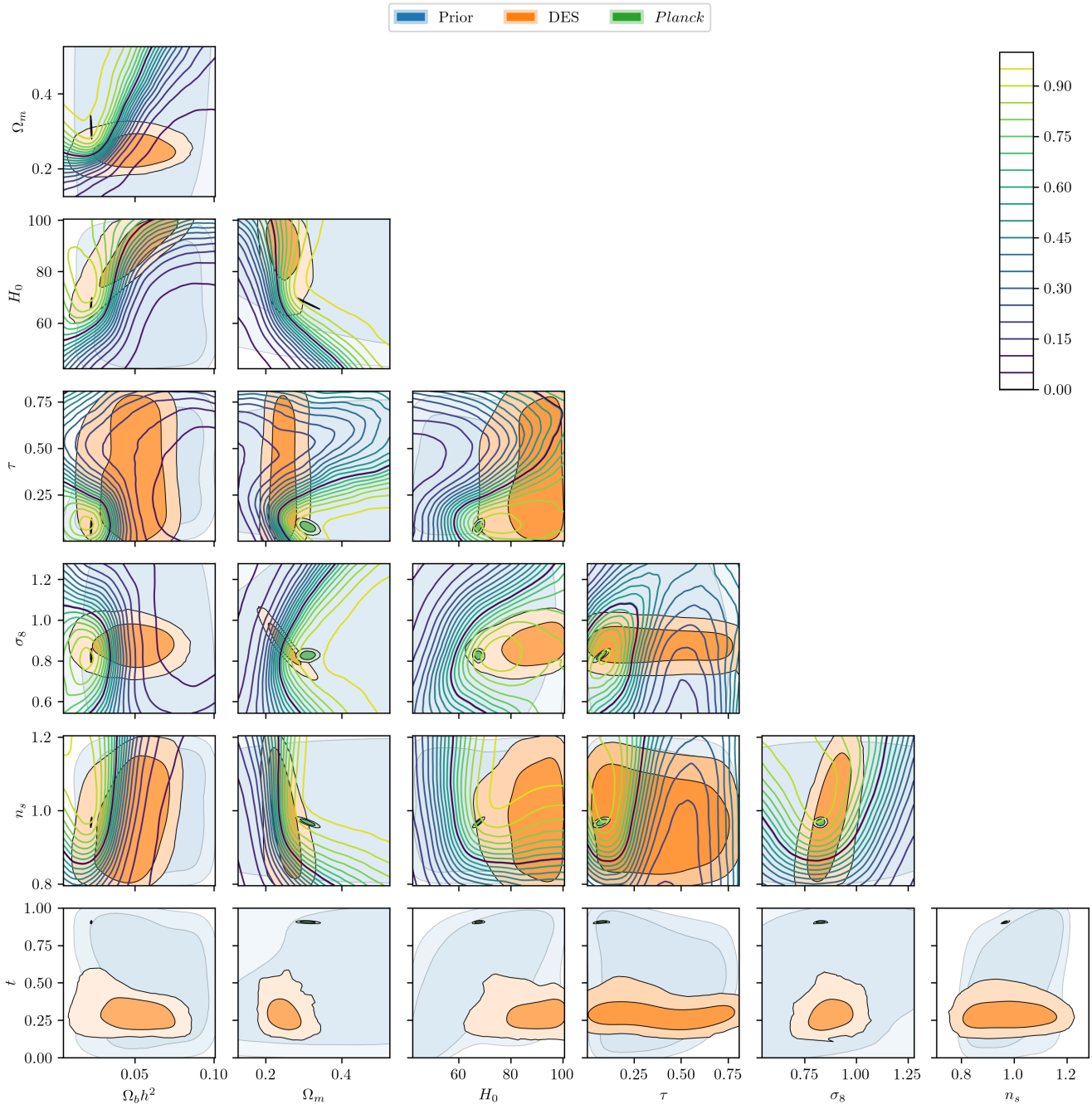

FIG. 7. The background of these plots shows the two-dimensional marginalised posteriors of the DES and *Planck* posteriors and their prior. The foreground in these plots, bar the last row, illustrates contour plots of the iso-tension coordinate hypersurfaces. The bold contour line is the midpoint between the two datasets in the 1D tension coordinate. When drawing these 2D contour plots, the four non-participating parameters are fixed at their respective means in the joint DES-*Planck* dataset. The final row plots the te;nsion coordinate t against the six cosmological parameters, visually emphasising the maximum tension found between these two datasets. The marginalised Bayes factor for this particular optimised neural network is computed to be -16.9.

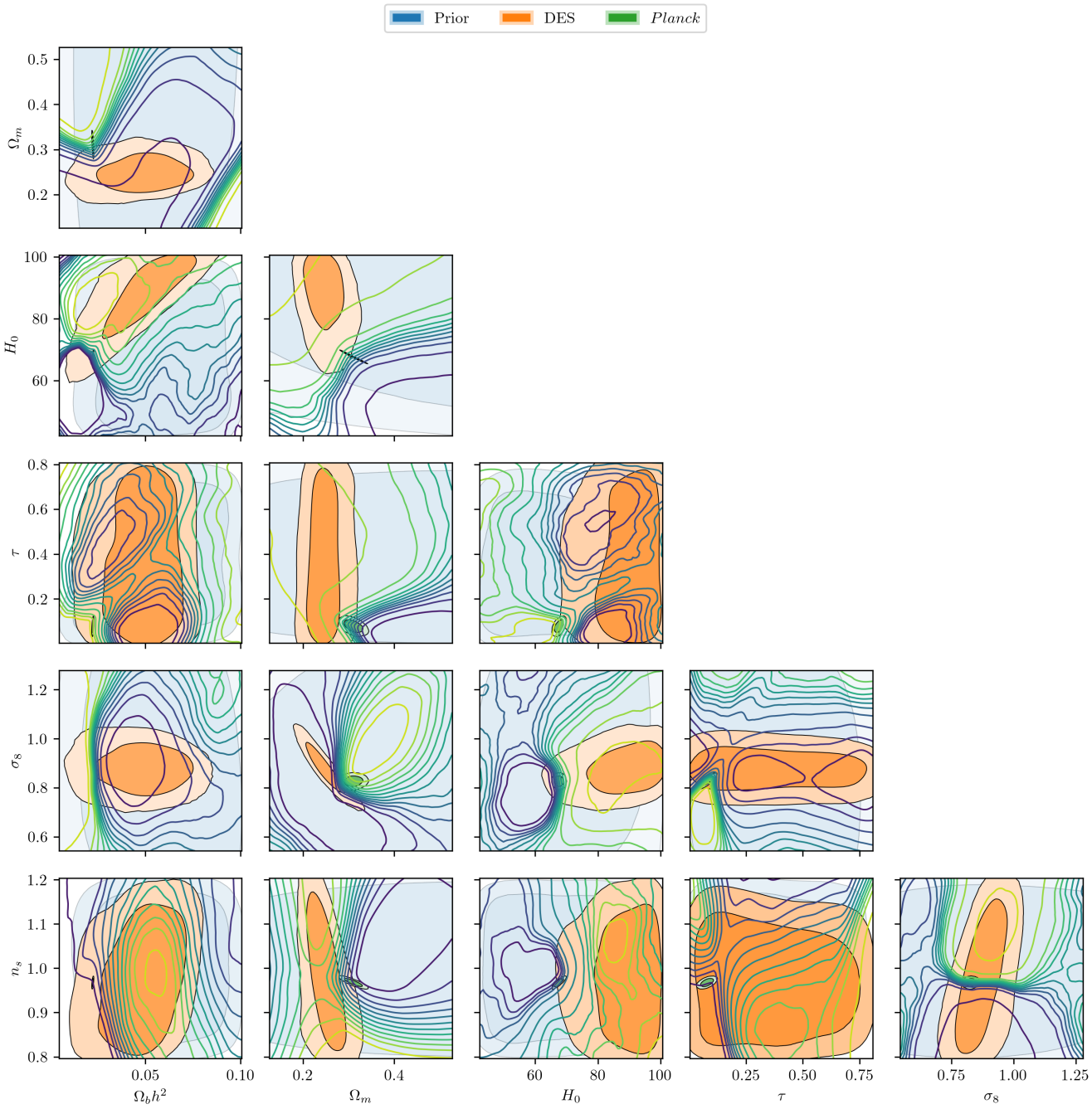

FIG. 8. Similar to Figure 7, the background shows 2D marginalised distributions of the two datasets and the prior, and the foreground illustrates iso-tension coordinate contour plots. The key difference in this figure is that each plot represents an independent neural network with the respective parameter pair as its input, unlike Figure 7 where all the plots collectively represent a single neural network with six input parameters. Note that the contour plots here are much noisier, particularly in regions that are further away from the two datasets. This likely points to an oversized hidden layer of the neural network, where it begins to over-fit to the statistical noise of the two datasets.

Parameter pair	$\log R_t$	$\log S_t$	p value	tension
H_0 - n_s	-3.18 ± 0.28	-4.42 ± 0.31	$(0.90 \pm 0.21)\%$	$(2.61 \pm 0.08)\sigma$
H_0 - τ	-3.54 ± 0.04	-4.80 ± 0.07	$(0.95 \pm 0.10)\%$	$(2.59 \pm 0.03)\sigma$
$\Omega_b h^2$ - σ_8	-3.04 ± 0.16	-4.49 ± 0.21	$(1.06 \pm 0.24)\%$	$(2.56 \pm 0.08)\sigma$
$\Omega_b h^2$ - H_0	-3.25 ± 0.12	-4.65 ± 0.10	$(1.09 \pm 0.15)\%$	$(2.55 \pm 0.05)\sigma$
$\Omega_b h^2$ - Ω_m	-3.35 ± 0.22	-4.30 ± 0.14	$(1.21 \pm 0.15)\%$	$(2.51 \pm 0.04)\sigma$
Ω_m - τ	-3.16 ± 0.13	-4.06 ± 0.06	$(1.66 \pm 0.08)\%$	$(2.40 \pm 0.02)\sigma$
H_0 - σ_8	-3.09 ± 0.13	-4.01 ± 0.04	$(1.73 \pm 0.11)\%$	$(2.38 \pm 0.02)\sigma$
Ω_m - n_s	-2.17 ± 0.43	-3.24 ± 0.45	$(1.85 \pm 0.82)\%$	$(2.36 \pm 0.16)\sigma$
Ω_m - H_0	-2.79 ± 0.23	-3.61 ± 0.13	$(2.53 \pm 0.38)\%$	$(2.24 \pm 0.06)\sigma$
τ - σ_8	-1.82 ± 0.26	-3.25 ± 0.28	$(2.81 \pm 0.32)\%$	$(2.20 \pm 0.05)\sigma$
σ_8 - n_s	-1.82 ± 0.05	-3.22 ± 0.13	$(3.62 \pm 0.62)\%$	$(2.09 \pm 0.07)\sigma$
$\Omega_b h^2$ - τ	-2.41 ± 0.25	-2.86 ± 0.19	$(3.95 \pm 0.47)\%$	$(2.06 \pm 0.05)\sigma$
Ω_m - σ_8	-2.25 ± 0.21	-2.93 ± 0.31	$(4.46 \pm 1.48)\%$	$(2.01 \pm 0.14)\sigma$
$\Omega_b h^2$ - n_s	-1.85 ± 0.22	-2.21 ± 0.35	$(5.00 \pm 0.94)\%$	$(1.96 \pm 0.08)\sigma$
τ - n_s	-0.67 ± 0.03	1.57 ± 0.05	$(92.09 \pm 2.84)\%$	$(0.10 \pm 0.04)\sigma$

Parameter	Mean $\log R_t$	Mean $\log S_t$	Mean p value	Mean tension
H_0	-3.17 ± 0.16	-4.30 ± 0.13	$(1.44 \pm 0.19)\%$	$(2.47 \pm 0.05)\sigma$
$\Omega_b h^2$	-2.78 ± 0.19	-3.70 ± 0.20	$(2.46 \pm 0.39)\%$	$(2.33 \pm 0.06)\sigma$
Ω_m	-2.74 ± 0.24	-3.63 ± 0.22	$(2.34 \pm 0.58)\%$	$(2.30 \pm 0.08)\sigma$
σ_8	-2.40 ± 0.16	-3.58 ± 0.19	$(2.74 \pm 0.55)\%$	$(2.25 \pm 0.07)\sigma$
τ	-2.32 ± 0.14	-2.68 ± 0.13	$(20.29 \pm 0.76)\%$	$(1.87 \pm 0.04)\sigma$
n_s	-1.94 ± 0.20	-2.30 ± 0.26	$(20.69 \pm 1.09)\%$	$(1.82 \pm 0.09)\sigma$

TABLE II. The upper table details the quantities of tension calculated from trained neural networks with parameter pairs as its input. The lower table gives the mean of the quantities of tension from the upper table for each parameter, where the mean is over the pairs which contain the respective parameter. Both tables are ordered in decreasing tension.

sion in itself. The unconstrained n_s posterior of the DES dataset should have meant that n_s does not have a large influence on the high dimensional tension, but the outcome of our first method shows otherwise. Given the unexpected nature of this result, it is important that more work is done in this area to either support or disfavour this peculiar observation.

VI. FURTHER WORK

Given the unexpected result with n_s, it would be beneficial to develop a more robust method of using the gradient of the tension coordinate to rank the importance of each parameter. Currently, the gradients have dimension of $[\text{parameter}]^{-1}$. It would be ideal if we can construct a dimensionless value to remove any undesirable effects from the posteriors and prior.

Other avenues to explore could involve looking into other methods to identify the source of tension. An example could be replacing a single parameter with noise to remove the parameter's influence on the neural network. We could also attempt to replace the loss function with other tension metrics, such as the Suspiciousness statistic.

Finally, a compelling question can be asked about whether it is possible to derive a meaningful expression from the neural network. This undoubtedly will allow an easier interpretation of the non-linear tension, and removes the need for secondary methods to identify the source of tension. An approach that might be of interest here is symbolic regression [29]. However, the more general problem of interpreting neural networks is non-trivial, and is currently an active area of research [30].

ACKNOWLEDGMENTS

I am deeply grateful to Dr. Will Handley for offering his unwavering support, guidance and expertise at every stage of this project.

Appendix A: Standardising the tension coordinate

We transform the coordinate to one where the prior is uniform and flat. To perform this transformation, we use the cumulative distribution function of the marginalised prior, which we denote as $\Pi^t(t) = \pi^t(X^t_\pi < t)$, to transform the posteriors and priors such that $\hat{\pi}^{t'}_P = \Pi(X^t_\pi)$, $\hat{\mathcal{P}}^{t'}_A = \Pi(X^t_A)$ and $\mathcal{P}^{t'}_B = \Pi(X^t_B)$. This gives a consistent range of $[0, 1]$ for the transformed 1D tension coordinate t'. We drop the prime in the main body of the paper, and let t refer to the transformed coordinate.

Note that this transformation does not affect the marginalised Bayes Factor R_t. We can see this in Equation (11), where the transformation of the posteriors are

cancelled out by the transformations of the prior and the integrated variable.

Appendix B: Kullback-Leibler divergence

The Kullback-Leibler (KL) divergence [31] $\mathcal{D}_D$ between the prior and posterior is given by

$$\mathcal{D}_D = \int \mathcal{P}_D(\theta) \log \frac{\mathcal{P}_D(\theta)}{\pi(\theta)} \, \mathrm{d}\theta. \tag{B1}$$

It describes the information compression between the prior and posterior, and can be interpreted as the average information provided by the posterior. The KL divergence is used in defining the Suspiciousness statistic, which itself is used to compute a more interpretable tension.

[1] A. Franklin, *Shifting Standards: Experiments in Particle Physics in the Twentieth Century* (University of Pittsburgh Press, Pittsburgh, 2013) Chap. Prologue, p. XXXVII, https://doi.org/10.2307/j.ctv80c9p7.

[2] G. de Vaucouleurs, New results on the distance scale and the Hubble constant, in *Galaxy Distances and Deviations from Universal Expansion*, edited by B. F. Madore and R. B. Tully (Reidel, Dordrecht, 1986) pp. 1–6, https://doi.org/10.1007/978-94-009-4702-3_1.

[3] A. Sandage and G. A. Tammann, Steps toward the Hubble constant. V. The Hubble constant from nearby galaxies and the regularity of the local velocity field., Astrophys. J. **196**, 313 (1975), https://doi.org/10.1086/153413.

[4] N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, and et al., Planck 2018 results, Astronomy and Astrophysics **641**, A6 (2020), https://doi.org/10.1051/0004-6361/201833910.

[5] T. M. C. Abbott, F. B. Abdalla, J. Annis, K. Bechtol, J. Blazek, B. A. Benson, R. A. Bernstein, G. M. Bernstein, E. Bertin, D. Brooks, and et al., Dark energy survey year 1 results: A precise H0 estimate from DES Y1, BAO, and D/H data, Monthly Notices of the Royal Astronomical Society **480**, 3879 (2018), https://doi.org/10.1093/mnras/sty1939.

[6] W. L. Freedman, B. F. Madore, T. Hoyt, I. S. Jang, R. Beaton, M. G. Lee, A. Monson, J. Neeley, and J. Rich, Calibration of the tip of the red giant branch, Astrophys. J. **891**, 57 (2020), https://doi.org/10.3847/1538-4357/ab7339.

[7] A. G. Riess, S. Casertano, W. Yuan, L. M. Macri, and D. Scolnic, Large magellanic cloud cepheid standards provide a 1the determination of the Hubble constant and stronger evidence for physics beyond ΛCDM, Astrophys. J. **876**, 85 (2019), https://doi.org/10.3847/1538-4357/ab1422.

[8] K. C. Wong, S. H. Suyu, G. C.-F. Chen, C. E. Rusu, M. Millon, and et al., H0LiCOW – XIII. A 2.4 per cent measurement of H0 from lensed quasars: 5.3σ tension between early- and late-universe probes, Monthly Notices of the Royal Astronomical Society **498**, 1420 (2019), https://doi.org/10.1093/mnras/stz3094.

[9] E. D. Valentino, O. Mena, S. Pan, L. Visinelli, W. Yang, A. Melchiorri, D. F. Mota, A. G. Riess, and J. Silk, In the realm of the hubble tension − a review of solutions (2021), arXiv:2103.01183 [astro-ph.CO].

[10] C. Heymans, T. Tröster, M. Asgari, C. Blake, H. Hildebrandt, B. Joachimi, K. Kuijken, C.-A. Lin, A. G. Sánchez, J. L. van den Busch, and et al., Kids-1000 cosmology: Multi-probe weak gravitational lensing and spectroscopic galaxy clustering constraints, Astronomy and Astrophysics **646**, A140 (2021), https://.doi.org/10.1051/0004-6361/202039063.

[11] W. Handley, Curvature tension: Evidence for a closed universe, Physical Review D **103**, 10.1103/physrevd.103.l041301 (2021), https://doi.org/10.1103/PhysRevD.103.L041301.

[12] W. Handley and P. Lemos, Quantifying tensions in cosmological parameters: Interpreting the DES evidence ratio, Physical Review D **100**, 10.1103/physrevd.100.043504 (2019), https://doi.org/10.1103/PhysRevD.100.043504.

[13] T. Charnock, R. A. Battye, and A. Moss, Planck data versus large scale structure: Methods to quantify discordance, Phys. Rev. D **95**, 123535 (2017), https://doi.org/10.1103/PhysRevD.95.123535.

[14] R. Trotta, Bayes in the sky: Bayesian inference and model selection in cosmology, Contemporary Physics **49**, 71 (2008), https://doi.org/10.1080/00107510802066753.

[15] P. Marshall, N. Rajguru, and A. c. v. Slosar, Bayesian evidence as a tool for comparing datasets, Phys. Rev. D **73**, 067302 (2006), https://doi.org/10.1103/PhysRevD.73.067302.

[16] P. Lemos, F. Köhlinger, W. Handley, B. Joachimi, L. Whiteway, and O. Lahav, Quantifying suspiciousness within correlated data sets, Monthly Notices of the Royal Astronomical Society **496**, 4647–4653 (2020).

[17] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural Networks **2**, 359 (1989), https://doi.org/10.1016/0893-6080(89)90020-8.

[18] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning (2018), arXiv:1811.03378 [cs.LG].

[19] V. Nair and G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10 (Omnipress, Madison, WI, USA, 2010) p. 807–814, https://dl.acm.org/doi/10.5555/3104322.3104425.

[20] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, On large-batch training for deep

learning: Generalization gap and sharp minima (2017), arXiv:1609.04836 [cs.LG].

[21] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), arXiv:1412.6980 [cs.LG].

[22] M. Rosenblatt, Remarks on Some Nonparametric Estimates of a Density Function, The Annals of Mathematical Statistics **27**, 832 (1956), https://doi.org/10.1214/aoms/1177728190.

[23] B. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability (Taylor & Francis, 1986).

[24] B. A. Turlach, Bandwidth selection in kernel density estimation: A review, in *CORE and Institut de Statistique* (1993).

[25] P. Lemos, M. Raveri, A. Campos, Y. Park, C. Chang, N. Weaverdyck, D. Huterer, A. R. Liddle, J. Blazek, R. Cawthon, A. Choi, and et. al, Assessing tension metrics with Dark Energy Survey and Planck data (2020), arXiv:2012.09554 [astro-ph.CO].

[26] W. Handley and P. Lemos, Quantifying tensions in cosmological parameters: Interpreting the DES evidence ratio (supplementary inference products), 10.5281/zenodo.4116393 (2020), https://doi.org/10.5281/zenodo.4116393.

[27] W. Handley, anesthetic: nested sampling visualisation, The Journal of Open Source Software **4**, 1414 (2019), https://doi.org/10.21105/joss.01414.

[28] W. Handley and P. Lemos, Quantifying the global parameter tensions between ACT, SPT, and Planck, Physical Review D **103**, 10.1103/physrevd.103.063529 (2021), http://dx.doi.org/10.1103/PhysRevD.103.063529.

[29] S.-M. Udrescu and M. Tegmark, Ai feynman: A physics-inspired method for symbolic regression, Science Advances **6** (2020), https://pubmed.ncbi.nlm.nih.gov/32426452.

[30] G. Montavon, W. Samek, and K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing **73**, 1 (2018), https://www.sciencedirect.com/science/article/pii/S1051200417302385.

[31] S. Kullback and R. A. Leibler, On Information and Sufficiency, The Annals of Mathematical Statistics **22**, 79 (1951), https://doi.org/10.1214/aoms/1177729694.