Constructing a Maximum Tension Coordinate with Neural Networks

Yi Jer Loh[*] and Will Handley[†]
Cavendish Laboratory, 19 J.J. Thomson Avenue, Cambridge CB3 0HE, UK

(Dated: May 5, 2021)

Tensions in cosmology, such as the Hubble tension and the Ω_m - σ_8 tension, are rising as measurement precision improves over recent years. We develop on the idea of finding the maximum tension possible between two cosmological datasets. We use a simple neural network with a single hidden layer to learn a function of the parameter space which maximises the tension between the two datasets. This function is a mapping from a high-dimensional parameter space to a 1D coordinate, which we call the *tension coordinate*. We apply this method to the *Planck* and DES datasets on the following parameters: physical 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 power law index n_s. We first use all six parameters on our neural network, and obtain a marginalised Bayes factor of $\log R_t = -16.8 \pm 0.5$, which is significantly more tension that if we assumed that the distributions were Gaussian. We then apply our neural network to pairs of parameters to identify the parameters which contain the most tension. Our two methods are able to identify H_0, Ω_m and $\Omega_b h^2$ as the parameters which contribute most to the tension.

I. INTRODUCTION

With cosmological measurements becoming more precise over recent years, disagreements between different datasets and methods have began to emerge. Observations of parameters surrounding the ΛCDM model have yielded discrepancies, or more commonly referred to as *tensions*, of close to 5σ – the indication of a significant result 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*.

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 Kilo Degree Survey (KiDS) [9]. There has also 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 [10].

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?

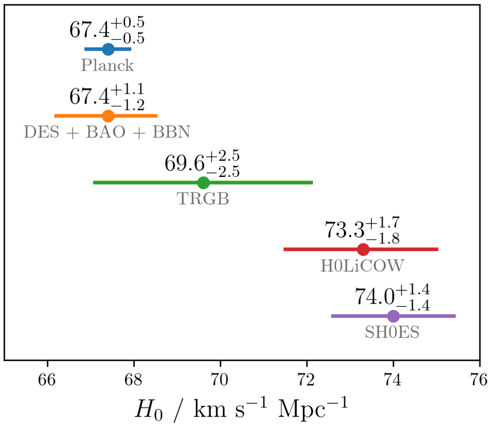

FIG. 1. A compilation of recent measurements of the Hubble constant H_0. The top two measurements are from early-universe datasets using ΛCDM cosmology [4, 5], while the remaining three are from late-universe datasets based off local distance ladder measurements [6–8]. The tension between the *Planck* 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 [11]. 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 [12] to better understand the problem at hand.

This paper aims to develop on the idea of maximum

[*] yjl34@cam.ac.uk
[†] wh260@cam.ac.uk

tension. With cosmological datasets, larger tensions often exist across multiple parameters rather than within each parameter on its own. A good example 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 non-existent when the parameters are inspected individually. In a high-dimensional parameter space, it is thus likely that there exists a combination of parameters which exacerbates and maximises tension.

In this paper, we explore how a high-dimensional parameter space can be mapped onto a *tension coordinate* – a lower-dimensional coordinate which maximises the tension between two datasets. A neural network is used 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

To describe Bayesian statistics, we use the following notation, 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.

The Bayesian evidence $\mathcal{Z}_D$ is defined as

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

Also known as the marginal likelihood in statistical literature [13], the evidence is often deemed to be a natural value for model and dataset comparisons within the Bayesian framework. However, the calculation of the evidence is generally computationally inhibitive, as it involves a multi-dimensional integral over the entire parameter space. Fortunately, there are now several tried-and-tested numerical methods that can reliably estimate the evidence, including thermodynamic integration [14] and nested sampling [15, 16].

B. Bayes Factor R

The canonical Bayes factor R [17] 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 datasets that are in agreement, 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 expression to work with, since it sidesteps the numerical complexity of computing evidences. This form of the Bayes factor also makes explicit its prior-dependence, which is arguably a concern [11]. 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 disagreeing datasets, hence low values of R must be taken more seriously than higher values of R.

C. Tension Coordinate

Most tension quantification methods, including the Bayes factor, are independent of the choice of parameters and the direction in the parameter space. These methods are not 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 the 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 [11]

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

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

Define the one-dimensional tension coordinate as a linear combination of the parameters, $t = n^T\theta$. The vector n can be naturally described as the direction of maximum 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

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

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

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

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

Substituting this back into the tension coordinate returns

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

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

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

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

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 $w = \arg\min_w R_t$. All of this can be achieved by representing T as a neural network, 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 [18].

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 2. 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 or regression, we do not need to worry about the problem of overfitting our datasets. In fact, one might argue that overfitting might even be a desired effect of the neural network, hence the large number of hidden nodes relative to the size of the input.

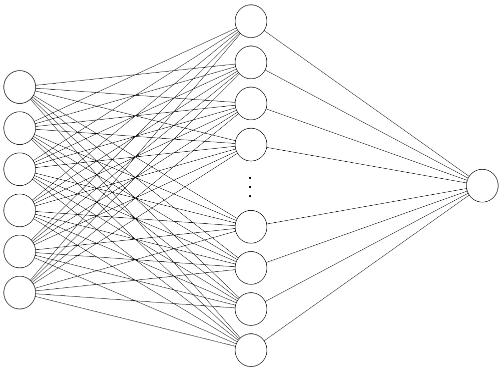

FIG. 2. A fully-connected neural network with 6 nodes in the input layer, an arbitrary number of nodes in the hidden layer, and a single node in the output layer.

The inputs to this neural network are the samples from two discrepant datasets with the relevant cosmological parameters, and the output node is a 1D tension coordinate. Due to its popularity and effectiveness in modern deep learning applications [19], a rectified linear unit (ReLU) [20] is chosen as the activation function between the hidden layer and output layer. We can express our neural network as

$$t = T(\theta; w) = \sum_i w_i^{(2)}\ \mathrm{ReLU}\bigg(\sum_j w_{ij}^{(1)}\theta_j + b_i\bigg) \tag{12}$$

where we have cosmological parameters θ_j, neural network weights $w_{ij}^{(l)}$, and bias b_i. As mentioned in Section 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 [21], we use all the data points of the two datasets to compute our gradient, and hence calculate the loss function. This method is preferred because it is crucial to provide the Bayes factor, which is our loss function, with a substantial amount of data points to better quantify the tension. We use the recently-popularised Adam optimisation algorithm [22] 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* [23, 24] 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

$$\hat{f}_h(x) = \frac{1}{h} \sum_{i=1}^{n} w_i K\left(\frac{x - x_i}{h}\right) \tag{13}$$

where K is the kernel function and h the bandwidth parameter (a notation first introduced by Rosenblatt [23]). 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

$$K(x) = \frac{1}{\sqrt{2\pi h^2}} e^{-x^2}. \tag{14}$$

Selecting the bandwidth parameter is more complicated. A review of the various choices of bandwidth parameters and bandwidth selection methods can be found in [25]. 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 $h = n_{eff}^{d+4}$, where $n_{eff} = (\sum w_i)^2 / \sum {w_i}^2$ 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

$$R_t \approx \sum_{j=1}^{M} \frac{\hat{\mathcal{P}}_A^t(t_j)\hat{\mathcal{P}}_B^t(t_j)}{\hat{\pi}^t(t_j)} \Delta t, \tag{15}$$

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, we find it 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 $\log R_t$. The LogSumExp method of calculating a log probability can be found in the source code of `Scipy`'s `gaussian_kde` method [26].

An alternative method to calculating R_t that might be viable is using a histogram density estimator with a Gaussian envelope rather than a top-hat function [27]. However, we find that this method suffers from underflow problems, which undesirably results in $\log R_t \to -\infty$. Hence, we stuck with the Gaussian kernel density estimator.

C. Toy Examples

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 tweaked 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()
$epochs \leftarrow 1000$

for $i \leftarrow 0$ to $epochs$ do
    $X_A^t \leftarrow net(X_A)$                              ▷ 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 very simplest case to begin with would be two disagreeing 2D Gaussian distributions. In our example, we place these two distributions around 5σ apart, and encase them in an arbitrary square prior, as seen in Figure 3(a). Our neural network is able to fit rough hyperplanes perpendicular to a line connecting the two distributions. This is a result that agrees with the analytical tension co-

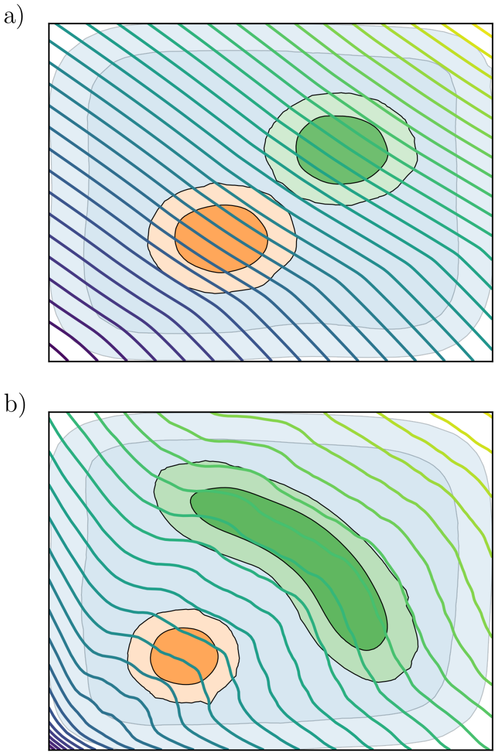

FIG. 3. These contour plots illustrate the iso-tension coordinate hypersurfaces shaped by the neural network after training for 500 epochs on our toy examples. a) shows two Gaussian distributions and b) shows a Gaussian distribution accompanied by a 'banana-shaped' distribution. The faded blue distribution in the background is the prior.

ordinate for the Gaussian case derived in Section II C 1. Our neural network is working so far.

The next example can be seen in Figure 3(b). We are still working in two dimensions, but now we have a Gaussian distribution accompanied by a concave quarter-circle distribution. Again, we have a square prior. 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 3(b). It is important to note that the statistical power of our neural network only lies in the region between and close to our two distributions, and starts to falter as we navigate further away. This effect can be noticed at the ends of the 'banana', where we see the contour lines beginning to bend away from the curve of the distribution.

D. Cosmological Dataset

In this paper, we work solely with 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 [11, 28]. The tensions mentioned in Section I of this paper, such as the 3σ tension in the Ω_m and σ_8 plane, also do exist between these two datasets.

We use the following six cosmological parameters – physical 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 power law index n_s. The derived parameters of Ω_m, H_0 and σ_8 are chosen over three other independent cosmological parameters because tensions in these parameters are well-known. 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 [29], and use the prior from the DES dataset as the prior for calculating the Bayes factor. This source provides nested sampling chains for all of the relevant independent, derived and nuisance parameters of both datasets. We use the `anesthetic Python` library [30] to extract weighted samples from the nested sampling chains.

E. Identifying the Source(s) of Tension

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 Y1 and *Planck* datasets. To evaluate the contribution of each parameter to the tension, we replace a single parameter with noise and feed this into the trained neural network, effectively eliminating its influence on the tension. We accomplish this by randomly permuting the values of the single parameter across all samples, while maintaining the order of the other parameters. By removing the effects of a more influential parameter, we would expect the two datasets to be in much less tension when compared to removing a less significant parameter.

To verify our results from training our neural network on all six parameters, we make pair-wise comparisons between the six parameters. We construct separate neural networks with two input nodes for each of the fifteen combinations of parameter pairs. This allows us to identify pairs of parameters that contribute most to the discrepancy between the two datasets. This hopefully will also allow us to rank the importance of each parameter, and compare the results obtained here against those from the above paragraph.

IV. RESULTS AND DISCUSSION

In this section, instead of expressing the 1D tension coordinate in its 'raw' form, 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 distribution, 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 range of $[0, 1]$ for the transformed 1D tension coordinate t'. We will drop the prime from here on, and let t refer to the transformed coordinate. Note that this transformation does not affect the marginalised Bayes Factor R_t.

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. Knowing that neural networks are only able to optimise to a local minimum rather than the global minimum, we perform 10 training repetitions to give a rough estimate of the lowest possible marginalised Bayes factor.

From the 10 training runs, we obtain a marginalised Bayes factor of $\log R_t = -16.8 \pm 0.5$. If we use the tension coordinate derived from the Gaussian assumption detailed in Section II C 1, we get a marginalised Bayes Factor of only $\log R_t = -0.94$. The significant difference between theses two values gives an initial impression of the neural network's ability to identify a tension coordinate that maximises tension. The density plots of the marginalised posteriors drawn in Figure 4 further demonstrates how the neural network is able to significantly amplify the distance and discrepancy between the two datasets.

The neural network's effectiveness in capturing the non-linearity of the tension between the two datasets is illustrated in Figure 5. In these 2D plots, we observe how our method is able to carve out hypersurface contours isolating the narrower distribution of the *Planck* dataset. For some of the plots, including the $\Omega_b h^2$ - τ and σ_8 - τ plots, the contours even encircle around the *Planck* dataset. The plots between the tension coordinate and cosmological parameters in the final row of the same figure also begin to describe how each parameter contributes to the tension. However, the wide constraints of the DES dataset renders an analysis via correlation coefficients inconclusive.

Before we go further, it is immensely important to reiterate the fact that many local minima exist in our loss function landscape. The many possibilities of a maximum tension coordinate implies that there are a multitude of forms in which the iso-tension coordinate hypersurfaces can be drawn. Figure 5 is only one example out of the many configurations, thus drawing conclusions from solely this figure is not a particularly good course of action.

To identify the parameters which contribute most to the tension, we introduce noise to a single parameter by randomly shuffling the values of the parameter across all samples of the dataset, while leaving the order of the other parameters undisturbed. This 'new' dataset is then fed into the trained neural network from before, and a marginalised Bayes factor is calculated from the output. We do this for each of the six parameters. The outcome from this procedure is detailed in Table I.

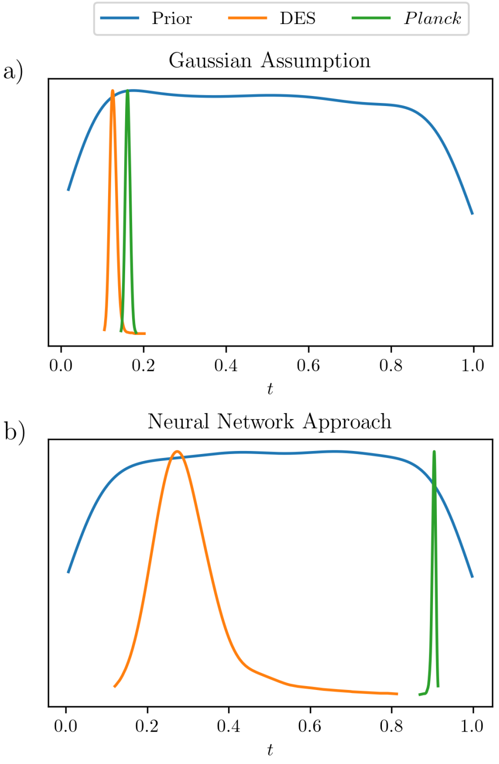

FIG. 4. Density plots of the marginalised posteriors of the DES and Planck datasets, and their prior, in the transformed 1D tension coordinate. (a) uses the Gaussian assumption described in Section II C 1 and (b) uses the neural network approach. They give marginalised Bayes factors of $\log R_t = -0.94$ and $\log R_t = -16.8 \pm 0.5$ respectively. Note that the prior is uniform across the range of the tension coordinate, and the bend at the edges of the prior is an just artefact of the kernel density estimator method.

Shuffled Parameter	$\log R_t$
H_0	-1.9 ± 0.3
$\Omega_b h^2$	-2.0 ± 0.3
Ω_m	-2.6 ± 0.3
n_s	-3.1 ± 0.4
σ_8	-6.4 ± 0.8
τ	-7.3 ± 0.9
None	-16.8 ± 0.5

TABLE I. Marginalised Bayes factors from using a dataset containing a single shuffled parameter as input to a trained neural network with six input nodes, as detailed in Section IV A. The more positive the $\log R_t$, the more influential the parameter is to defining the tension coordinate.

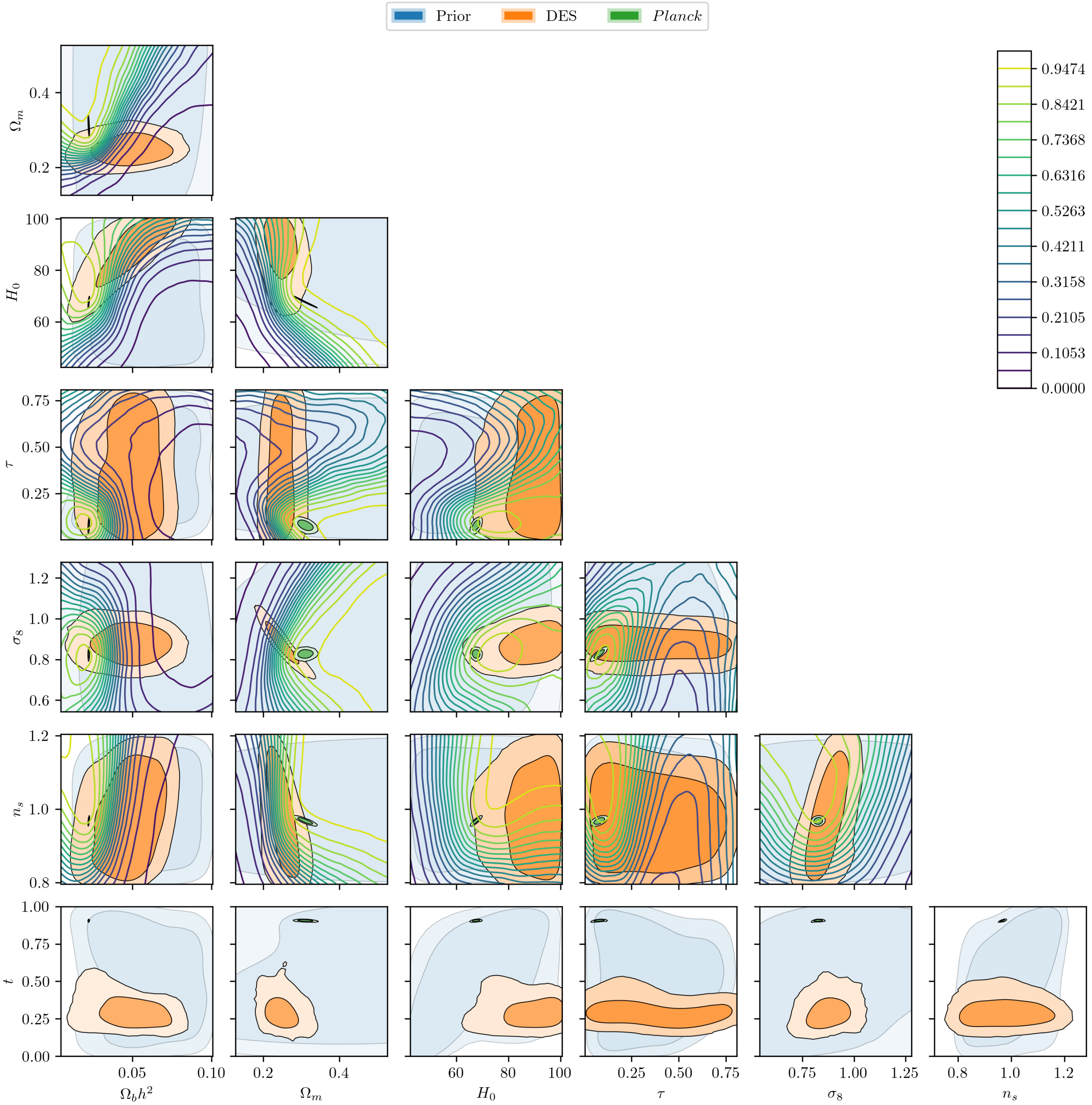

FIG. 5. The background of these plots shows the two-dimensional marginalised distributions of the DES and *Planck* datasets, and the prior. These distributions are represented by solid iso-probability contours containing 68% and 95% of the marginalised probability mass. The foreground in these plots, bar the last row, illustrates contour plots of the iso-tension coordinate hypersurfaces. These contour plots are drawn by varying the two relevant cosmological parameters and fixing the other four, then feeding these data points into the trained neural network. The four data points are fixed at the weighted parameter means of the combined DES-*Planck* dataset. The final row plots the tension coordinate t against the six cosmological parameters, visually emphasising the maximum possible tension found between these two datasets. The marginalised Bayes factor for this particular optimised neural network is computed to be -16.9, while we get an estimate of $\log R_t = -16.8 \pm 0.5$ from 10 separate gradient descent runs on the same neural network setup.

A more positive $\log R_t$ (less tension) here indicates a more important parameter in defining the tension coordinate, since shuffling its values effectively removes the influence of the parameter. One can think of the absolute difference between the unadulterated dataset and the shuffled dataset to be a value quantifying the contribution to the tension of a particular parameter. The larger the difference, the larger the contribution.

The result here points to H_0 and $\Omega_b h^2$ being the largest contributor to the tension, with Ω_m coming in at a close third. This result supports the examples of cosmological tensions mentioned in Section I, particularly for H_0 and Ω_m. The result also indicates that τ and σ_8 are the least significant influencers of the tension. It is not unexpected for the former since the DES dataset doesn't constrain τ. However, it is curious that the latter is considered to be of low significance given the existence of the Ω_m - σ_8 tension in the community. Finally, we notice a considerable contribution to tension from the n_s parameter. This is somewhat unanticipated as DES does not set any constraints on n_s. A density plot from the perspective of the n_s parameter (can be inferred from the bottom-right plot in Figure 5) also does not reveal any obvious tensions. The neural network appears to have picked out a non-trivial source of tension, but further exploration is needed to better understand this behaviour.

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} with the Adam optimiser. We perform 5 training repetitions for each parameter pair. The results are given in Table II. 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 results here generally correlate well with the results from the previous section. We still find that Ω_m, H_0 and $\Omega_b h^2$ are the top three parameters containing the most tension, albeit permuted in a slightly different order. The largest differences compared to the previous section's results is the increase of tension found in σ_8 and the drop in rank to last for n_s.

The rank of parameters containing the most tension here is more in line with what we would expect. The unconstrained parameters of the DES dataset – τ and n_s, are now considered as the least discrepant. On the opposite end of the scale, the results are able to reinforce known cosmological tensions found in H_0 and the Ω_m - σ_8 plane. However, it is interesting to note that our neural networks do not consider the pairing of Ω_m - σ_8 to contain the most tension, but find pairings such as $\Omega_b h^2$ - Ω_m and H_0 - τ to have the largest discrepancies. A possible reason for these large tensions could be the significant difference between the widths of the *Planck* and DES datasets for these parameters. For example, one could argue that the wide constraint favouring a larger value of H_0 or $\Omega_b h^2$ (or disfavouring a lower value) is a key contributor to the tension.

Parameter Pair	$\log R_t$	Rank
$\Omega_b h^2$ - Ω_m	-3.66 ± 0.16	1
$\Omega_b h^2$ - H_0	-3.26 ± 0.11	4
$\Omega_b h^2$ - τ	-2.17 ± 0.44	10
$\Omega_b h^2$ - σ_8	-3.40 ± 0.17	3
$\Omega_b h^2$ - n_s	-1.95 ± 0.22	12
Ω_m - H_0	-3.02 ± 0.21	7
Ω_m - τ	-3.23 ± 0.22	6
Ω_m - σ_8	-2.73 ± 0.25	8
Ω_m - n_s	-2.62 ± 0.41	9
H_0 - τ	-3.57 ± 0.04	2
H_0 - σ_8	-3.25 ± 0.06	5
H_0 - n_s	-1.94 ± 0.47	13
τ - σ_8	-2.13 ± 0.17	11
τ - n_s	-0.73 ± 0.01	15
σ_8 - n_s	-1.86 ± 0.06	14

Parameter	Mean $\log R_t$	Rank
Ω_m	-3.05 ± 0.25	1
H_0	-3.01 ± 0.18	2
$\Omega_b h^2$	-2.89 ± 0.22	3
σ_8	-2.68 ± 0.14	4
τ	-2.37 ± 0.17	5
n_s	-1.82 ± 0.24	6

TABLE II. The upper table details the marginalised Bayes factors calculated from trained neural networks with parameter pairs as its input. We rank the pairings by increasing $\log R_t$, or decreasing tension, in the rightmost column. The lower table gives the mean $\log R_t$ from the results of parameter pairings containing the respective parameter.

It is important to highlight that despite n_s ranking as the parameter containing the least tension, a mean marginalised Bayes factor of $\log R_t = -1.82 \pm 0.24$ is by no means an indication of a concordant parameter. This value of $R_t = 0.162$ is definitely considered to be in tension, but care needs to be taken here. Is the source of discrepancy from the other parameters paired to n_s, or does n_s itself contain some tension? These results do not reveal such information, and more work needs to be done to isolate each individual parameter.

V. CONCLUSIONS

Applying our neural network to the six cosmological parameters of $\Omega_b h^2$, Ω_m, H_0, τ, and σ_8 and n_s yields a maxmimum tension of $\log R_t = -16.8 \pm 0.5$ between the *Planck* and DES datasets. The neural network was successful in capturing the non-linearity of the iso-tension coordinate hypersurfaces, and exaggerating the distance between the two datasets. The marginalised Bayes factor in this tension coordinate is significantly more discrepant than if we were to use an analytical approach assuming that the distributions were Gaussian.

We used two different methods in an attempt to identify the source of and contribution to tension from each parameter. We found that H_0, Ω_m and $\Omega_b h^2$ were the three main contributors to tension in both methods. This lends support to the well-established tensions found in H_0 and the Ω_m - σ_8 plane. On the lower end of the scale of lesser tension, there were some disagreements between the two approaches. The first approach ranked the unconstrained parameter in the DES dataset of n_s to be more influential than one would expect, and σ_8 to be less important. The second approach gave results that were more in line with what was expected, where the unconstrained parameters in the DES dataset of n_s and τ were ranked as the least discrepant.

We mentioned in Section I that tensions could be an artefact of systematic errors, or could point to new physics. These results unfortunately do not hint at either direction, however it reinforces on the direction in which current research is heading toward. This paper points to H_0, Ω_m and $\Omega_b h^2$ as highly discrepant cosmological parameters, hence these are worthwhile areas to be looking into. It would also be interesting to see this method applied to other dataset pairings, which could potentially reveal new underlying cosmological tensions.

ACKNOWLEDGMENTS

Appendix A: Appendixes

Appendix B: A little more on appendixes

1. A subsection in an appendix

[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] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] N. Lartillot and H. Philippe, Computing Bayes Factors Using Thermodynamic Integration, Systematic Biology **55**, 195 (2006), https://doi.org/10.1080/10635150500433722.

[15] J. Skilling, Nested sampling for general Bayesian computation, Bayesian Analysis **1**, 833 (2006), https://doi.org/10.1214/06-BA127.

[16] W. J. Handley, M. P. Hobson, and A. N. Lasenby, polychord: nested sampling for cosmology, Monthly Notices of the Royal Astronomical Society: Letters **450**, L61 (2015), https://doi.org/10.1093/mnrasl/slv047.

[17] 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`.

[18] 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`.

[19] 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].

[20] 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.

[21] 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].

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

[23] 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`.

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

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

[26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods **17**, 261 (2020).

[27] K. Schütt, F. Arbabzadah, S. Chmiela, and et al, Quantum-chemical insights from deep tensor neural networks, Nat Commun **8** (2017), `https://doi.org/10.1038/ncomms13890`.

[28] 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].

[29] 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`.

[30] W. Handley, anesthetic: nested sampling visualisation, The Journal of Open Source Software **4**, 1414 (2019).