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: April 12, 2021)

An article usually includes an abstract, a concise summary of the work Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac nisi ex. Sed finibus, sem sit amet tristique vestibulum, elit nibh sagittis eros, et vestibulum odio enim et ante. Interdum et malesuada fames ac ante ipsum primis in faucibus. Donec condimentum libero ligula, nec facilisis diam suscipit eget. Sed accumsan velit nec nisi commodo elementum nec sed ligula. Curabitur rutrum, massa vitae eleifend dictum, enim orci vulputate mi, ac venenatis ante elit quis dolor. Phasellus vitae sapien quis enim convallis bibendum.Mauris nec nulla tellus. Curabitur augue mauris, tristique eget elit vel, ornare congue lectus. Etiam ac vestibulum odio. Nullam blandit ante a turpis maximus tristique. Pellentesque at convallis metus. Etiam elit neque, tincidunt vitae placerat sit amet, rhoncus a justo.

I. INTRODUCTION

With cosmological measurements becoming more precise over recent years, disagreement 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?

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.

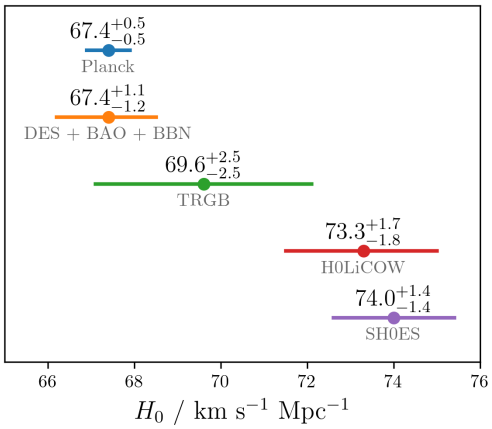

FIG. 1. A compilation of recent measurements of the Hubble constant H_0. The top two measurements are from early-universe datasets using ΛCDM cosmology [4, 5], while the remaining three are from late-universe datasets based off local distance ladder measurements [6–8]. The tension between the *Planck*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

This paper aims to develop on the idea of maximum 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

* yjl34@cam.ac.uk
† wh260@cam.ac.uk

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 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 $R \gg 1$ thus 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 does not explicitly call for any 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. More importantly, consistent datasets with sensible priors *cannot* be reported to be in tension by the Bayes factor, which 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.

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

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

where t is defined up to a normalisation constant.

2. Non-Gaussian case

Dataets 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

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

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 maxiumum 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

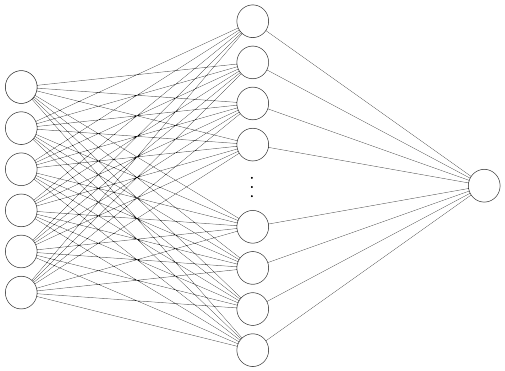

FIG. 2. A fully-connected neural network with 6 nodes in the input layer, an arbitrary number of nodes in the hidden layer, and a single node in the output layer.

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 nodes.

The inputs to this neural network are the samples from two discrepant datasets with 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. 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 chosen 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 0.001.

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. In this paper, we deal with datasets containing weighted samples of probability distributions in parameter space. The weighted samples of the distributions in the 1D tension

coordinate is expressed as

$$S_D = \left\{(t_D^{(i)}, w_D^{(i)}), i = 1, \ldots, N_D\right\}, \quad t_D = T(\theta_D) \quad (12)$$

where $\sum_i^{N_D} w_D^{(i)} = 1$. To numerically calculate R_t, we need to swap out the integral in Equation 11 for a sum of binned weights obtained via a histogram.

Let us begin by defining a typical histogram as a probability density estimator. For an arbitrary weighted dataset $\{(X^{(i)}, w^{(i)}), i = 1, \ldots, N, \sum_i^N w^{(i)} = 1\}$ of range $[x_0, x_1]$, bins $b^{(j)}$ and constant bin width Δ, the interval of $b^{(j)}$ is defined as $[x_0 + j\Delta, x_0 + (j+1)\Delta]$. Let the centre of $b^{(j)}$ be denoted by $c^{(j)}$. The probability estimate of values in the jth bin is then given by

$$\hat{p}^{(j)}(x) = \frac{1}{\Delta} \sum_{i=1}^{N} w^{(i)} F(X^{(i)} - c^{(j)}) \quad (13)$$

where $x \in b^{(j)}$, and F is a top-hat function of width Δ centred at zero.

With this definition of a histogram, the marginalised Bayes factor R_t can be approximated as

$$R_t \approx \sum_i^{N_\pi} \frac{\left(\sum_{c_A^{(j)} \in b_\pi^{(i)}} \Delta_A \hat{p}_A^{(j)}\right)\left(\sum_{c_B^{(j)} \in b_\pi^{(i)}} \Delta_B \hat{p}_B^{(j)}\right)}{\Delta_\pi \hat{p}_\pi^{(i)}}, \quad (14)$$

where we have two datasets A and B with a shared prior π.

However, we have a big problem. The optimisation of the neural network uses a gradient-based method, which requires our computation of R_t to be smooth and differentiable everywhere. The top-hat function F in Equation 13 is not smooth. Fortunately, we can replace F with a smooth function with non-extreme derivatives to approximate a histogram. We choose a Gaussian to be the replacement of F [23], such that

$$F(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (15)$$

with $\sigma = \Delta/2$. We find using such a Gaussian envelope is able to approximate a top-hat histogram quite well. [Give chi-squared?]

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 ineffective as $R_t \to 0$. On the other hand, $\log R_t$ with its range of $(-\infty, \infty)$ allows gradient descent to make more confident steps towards the local minimum.

Note that instead of using a histogram as a probability density estimator, an alternative method is to use a Gaussian kernel density estimator (KDE) [24]. However, we find that the histogram method provided more stability and computational efficiency during gradient descent, hence the decision to stick with histograms using a Gaussian envelope function.

C. Toy Examples

Algorithm 1 takes us through the training of our neural network using PyTorch-esque langauge. Note that the learning rate of the optimisation method and number of epochs can be tweaked in order to reach a stable minimum of the Bayes Factor R_t.

Algorithm 1 Training of Neural Network

```
X_A ← Dataset A; w_A ← Weights A
X_B ← Dataset B; w_B ← Weights B
X_p ← Prior; w_p ← Prior Weights

net ← NeuralNetwork(in=6, hidden=4096, out=1)
optim ← AdamOptimizer(net.parameters(), learning_rate=0.001)
loss ← LogBayesFactor()
epochs ← 1000

for i ← 0 to epochs do
    t_A ← net(X_A)                    ▷ Tension coordinates
    t_B ← net(X_B)
    t_p ← net(X_p)

    R_t ← loss(t_A, t_B, t_p, 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 coordinate 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

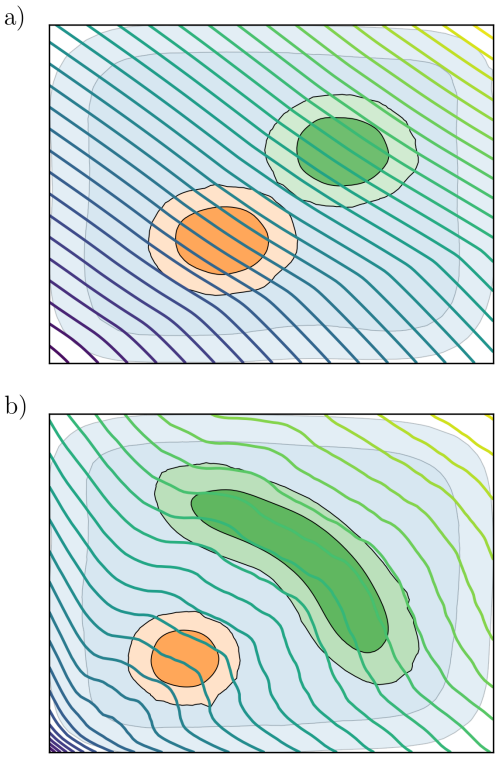

FIG. 3. These contour plots illustrate the hypersurfaces shaped by the neural network after training for 500 epochs. a) shows two Gaussian distributions and b) shows a Gaussian distribution accompanied by a 'banana-shaped' distribution. The faded blue distribution in the background is the prior.

contour lines beginning to bend away from the curve of the distribution. Our neural network is working well.

D. Cosmological Dataset

In this paper, we work solely with cosmological measurements made by the DES and the *Planck* satellite. We choose these two datasets because of the well reported tensions between them in recent years [11, 25]. 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 well-established discrepancies allows us to verify and reinforce the existing tension in them.

We obtain the DES Y1 and *Planck* datasets from [26]. 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 [27] to extract weighted samples from the nested sampling chains.

E. Identifying Source(s) of Tension

IV. RESULTS AND DISCUSSION

V. CONCLUSIONS

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] 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.

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

[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).