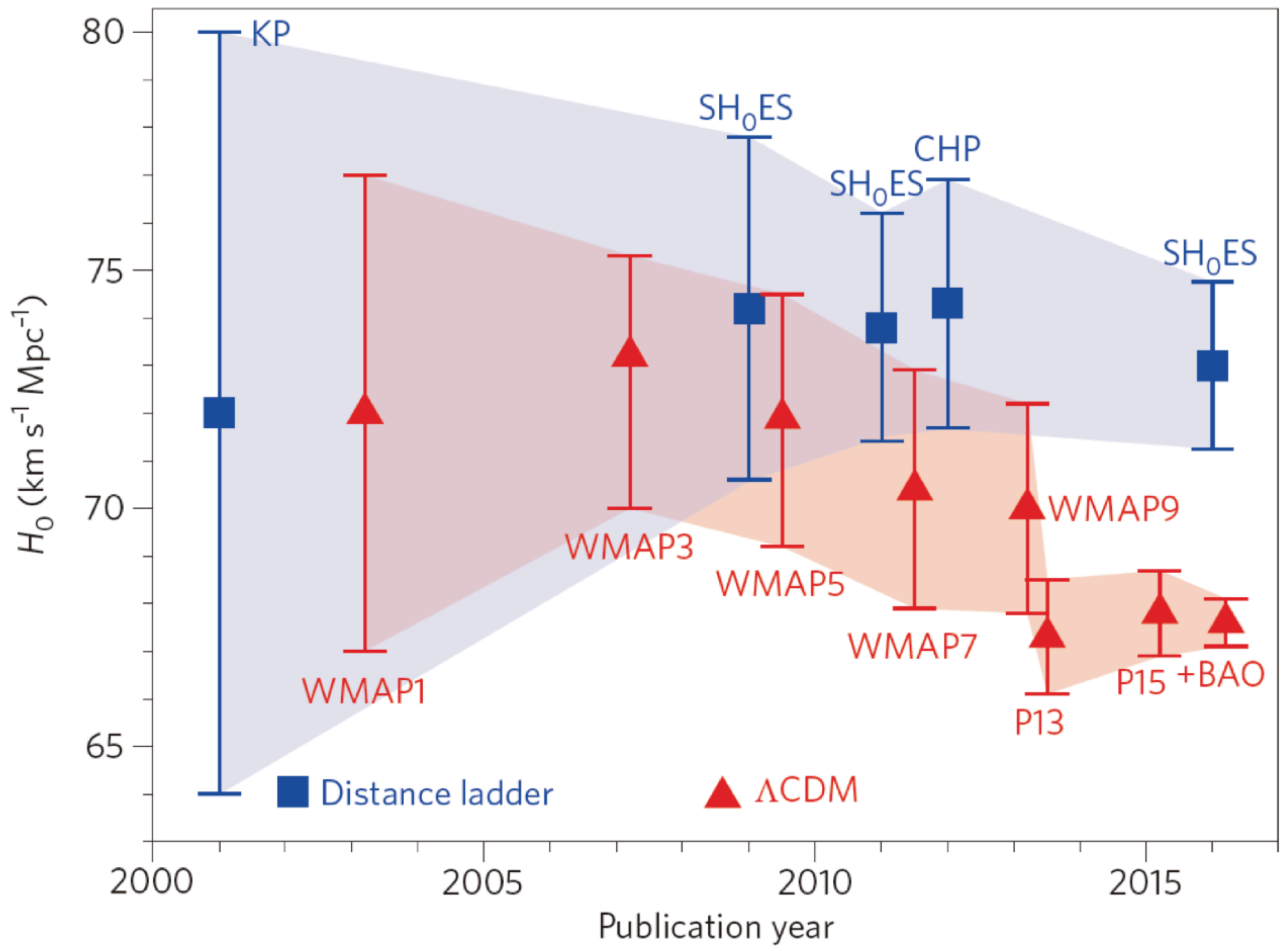
Calibrating Tension Statistics with Neural Ratio Estimators

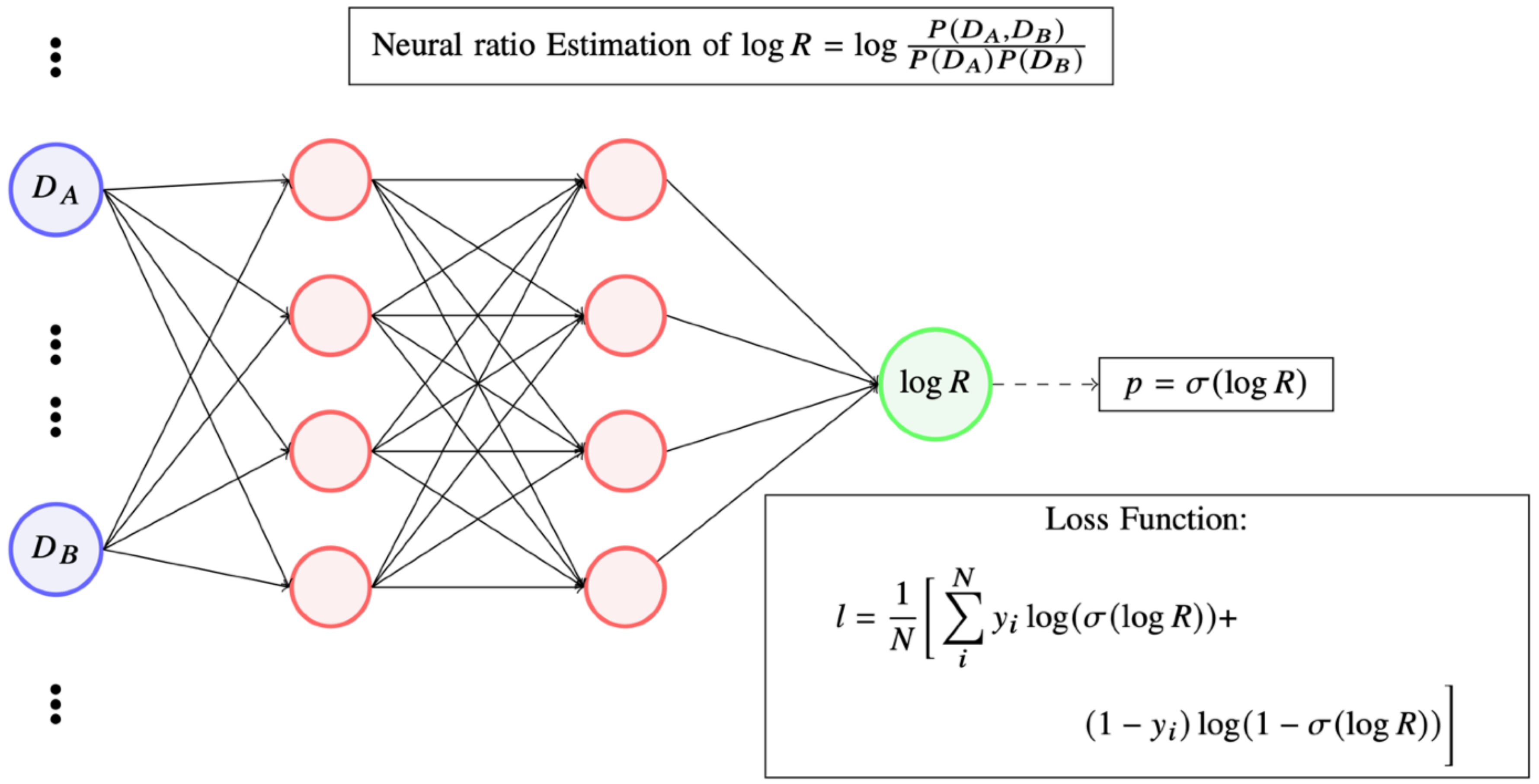
Harry Bevins
with Thomas Gessey-Jones and Will Handley
University of Cambridge

https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

Calibrating Tensions

1. Why are we interested and how do we measure tension?
2. Calibrating with Neural Ratio Estimation
3. Demonstrations



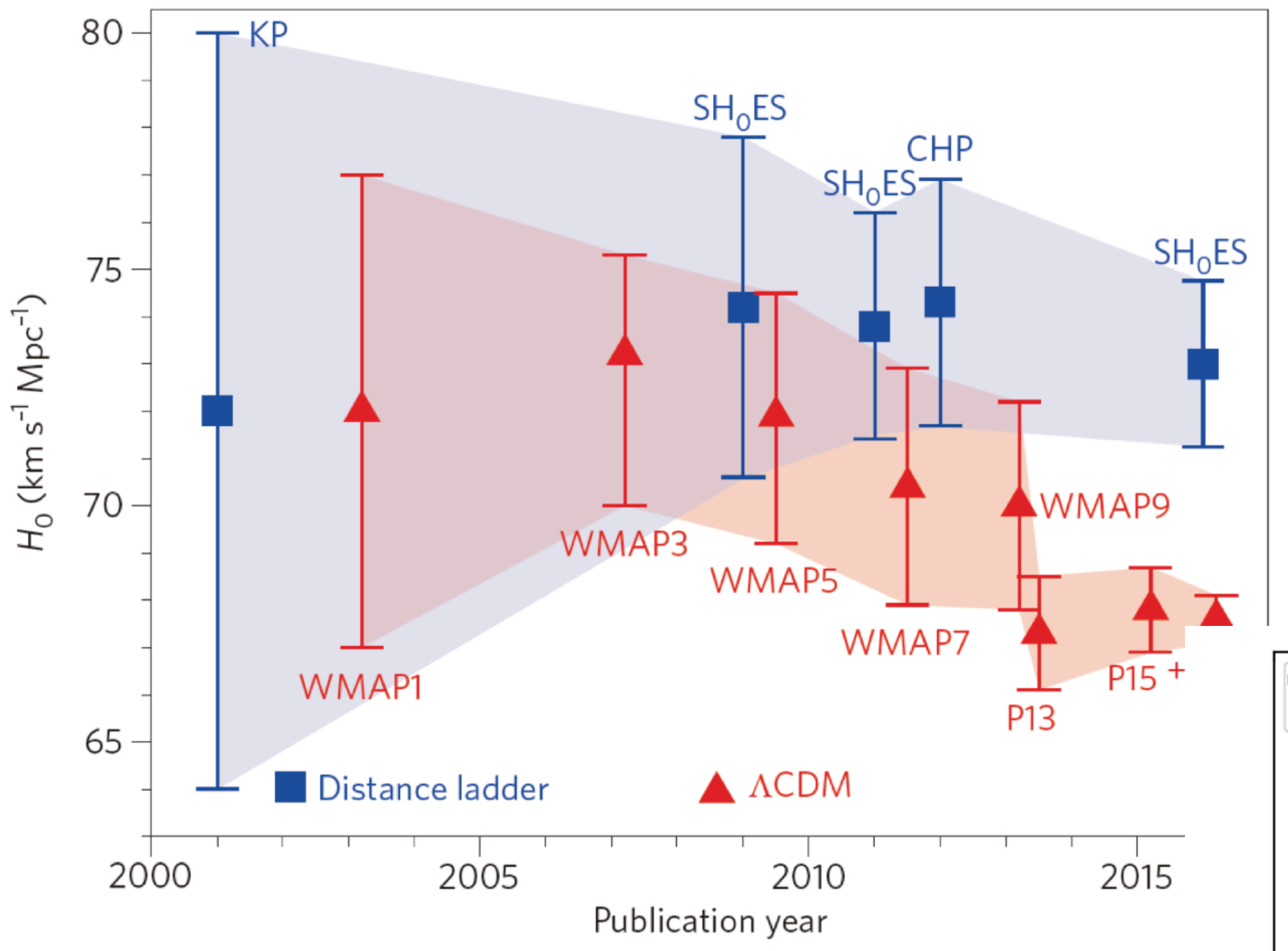


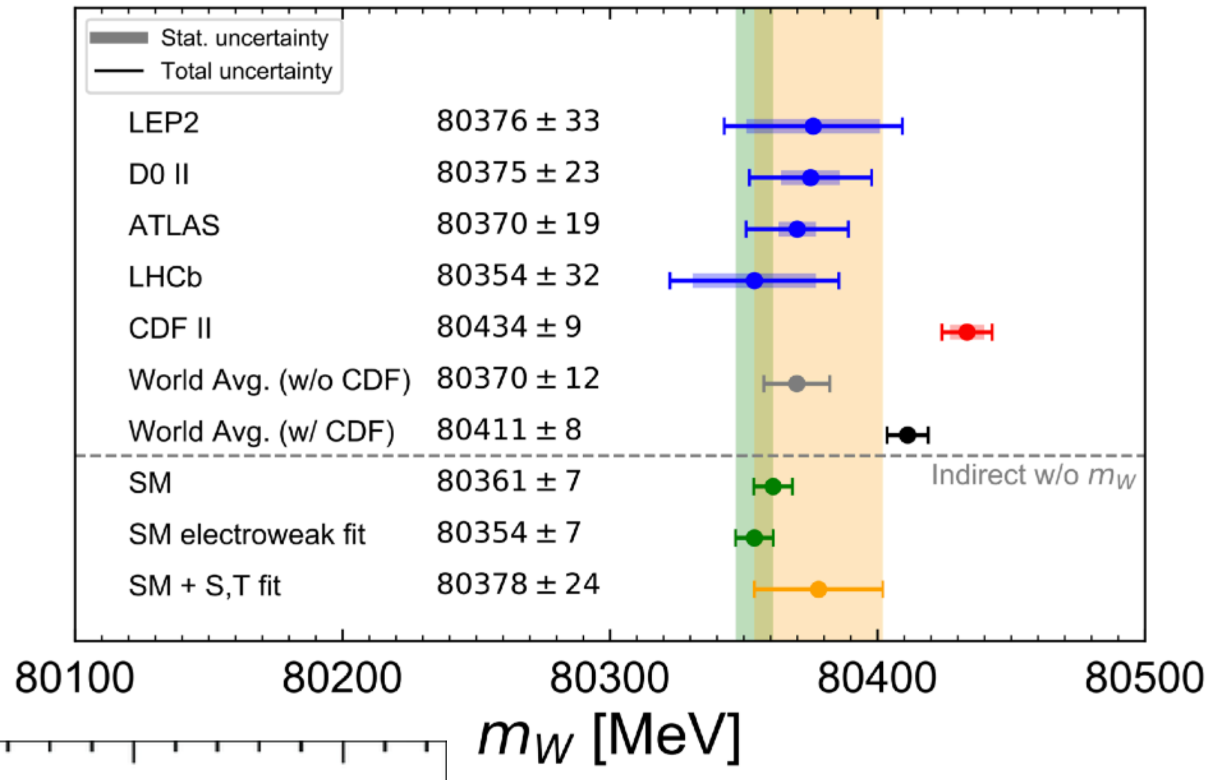
https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

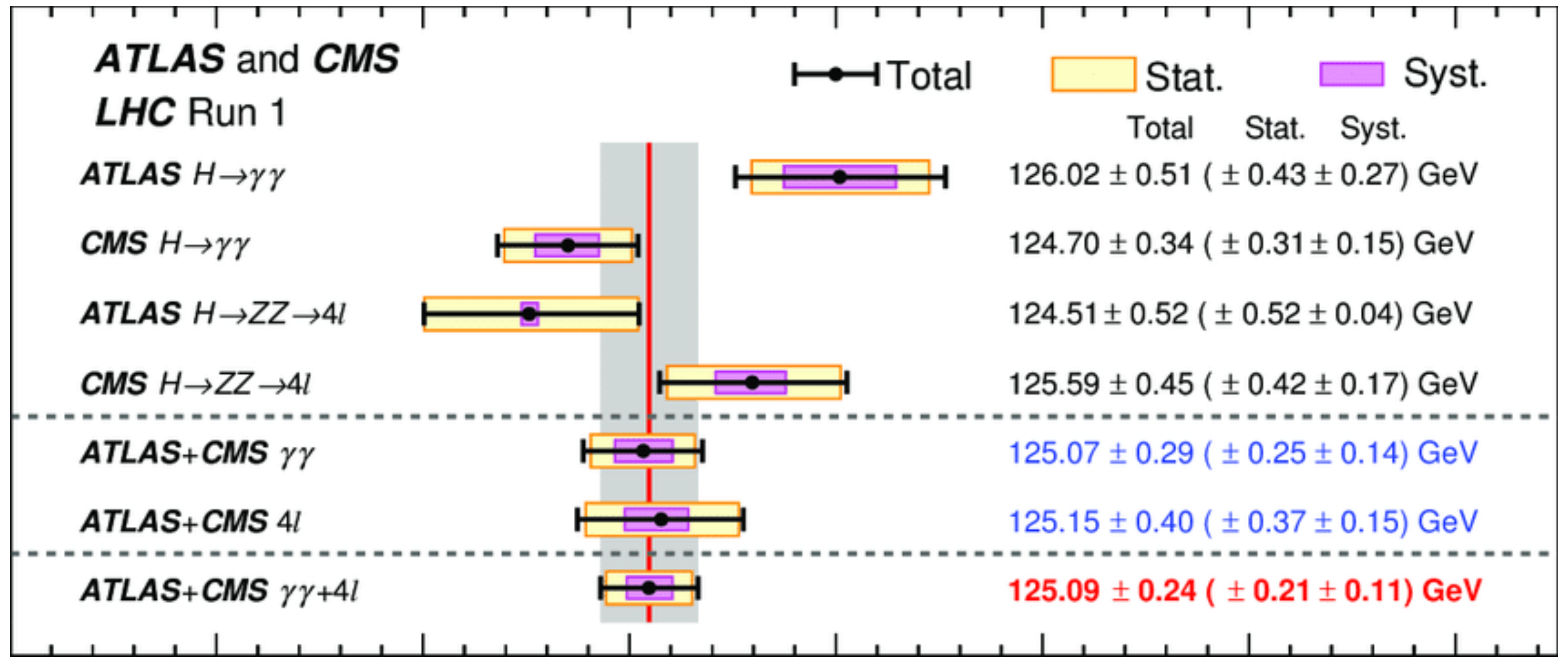
Why are we interested?

Why are we interested in tension?

- Important to be able to independently observe and confirm experimental results
- When two experiments give different results we call this a tension
- Understanding where tension comes from can lead us to new physics and a better understanding of our instruments







https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

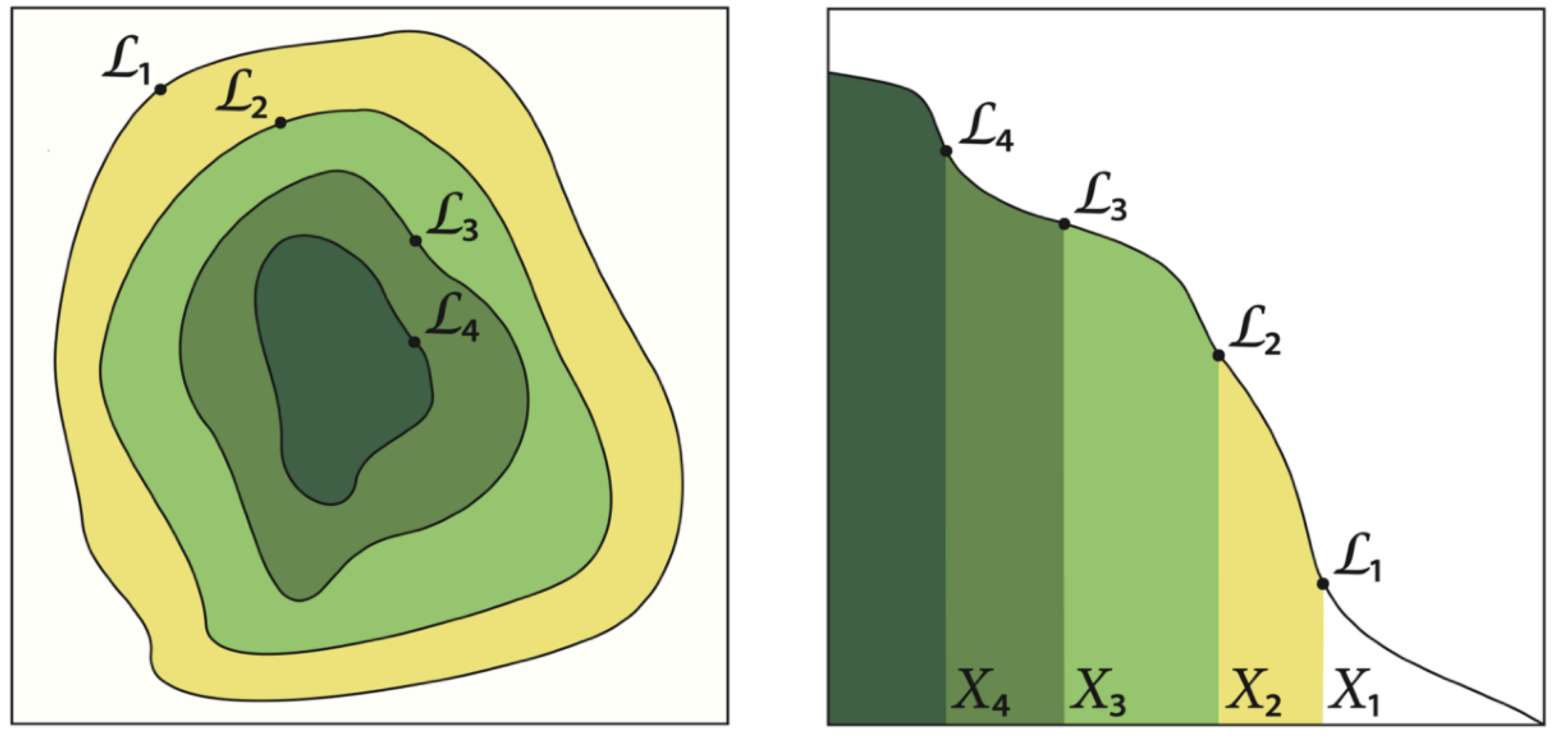
Measuring tension

- Parameter differences, goodness of fit degradation, suspiciousness (see 2012.09554 for a review)

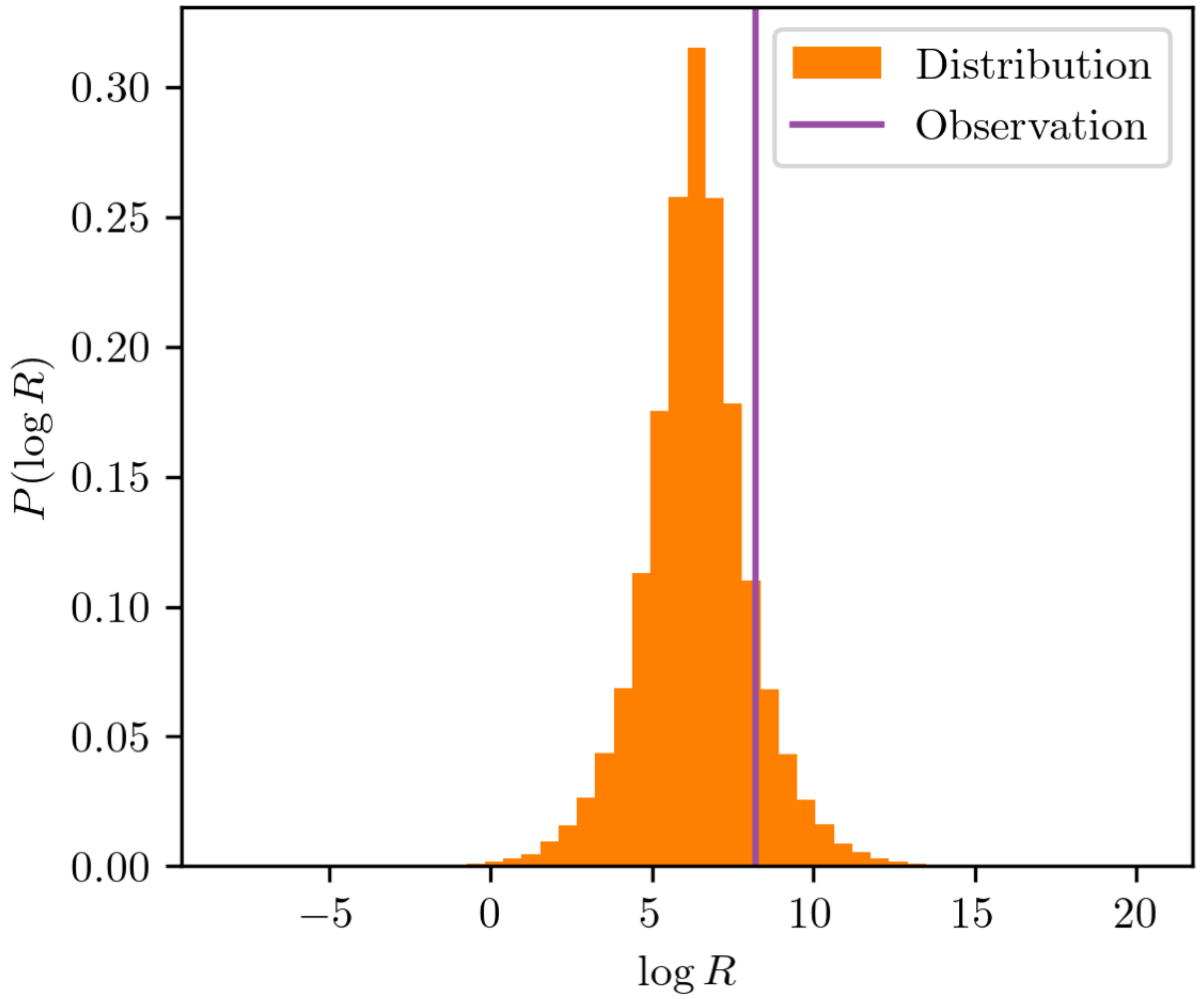
- Here, interested in evidence ratio

$$R = \frac{P(D_A, D_B)}{P(D_A)P(D_B)} = \frac{Z_{AB}}{Z_A Z_B}$$

- For any pair of experiments, model and prior there is a distribution of in concordance R values



(a) (b)



https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

Measuring tension

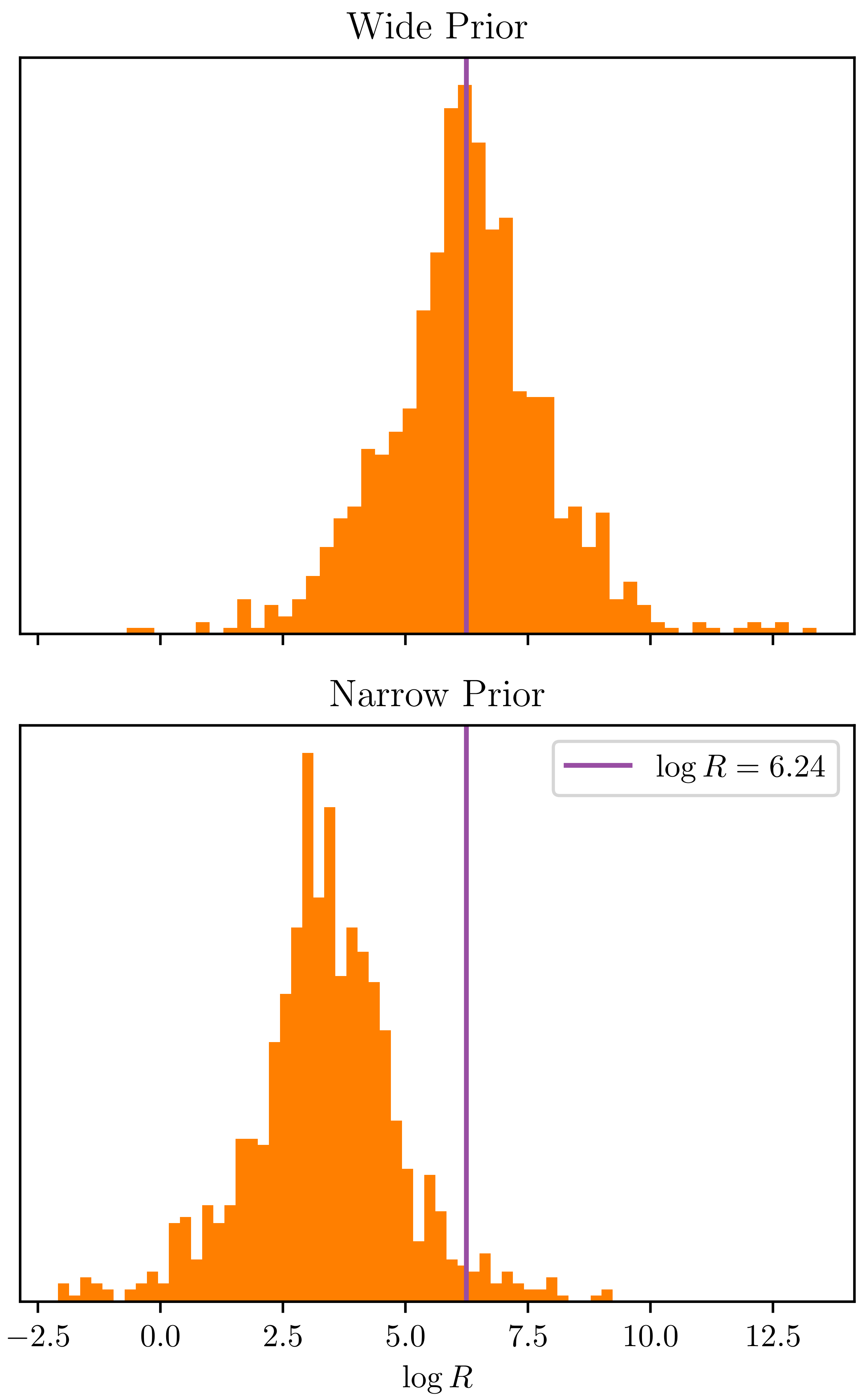
- The fractional increase in our confidence in one experiment given data from another

- Dimensionally consistent and parameterisation invariant

- But prior dependent and hard to interpret

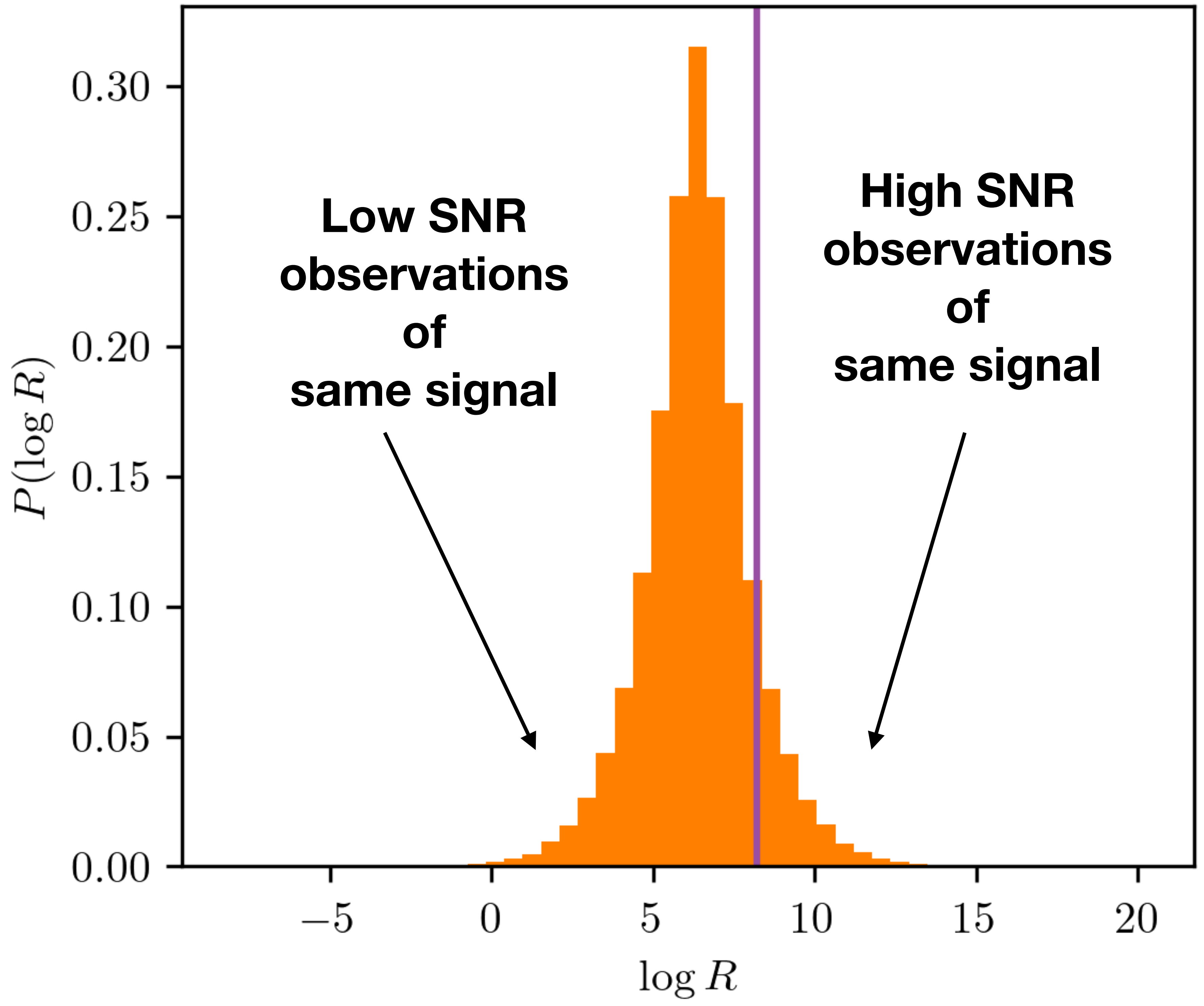
  - $R \gg 1 \rightarrow$ in concordance

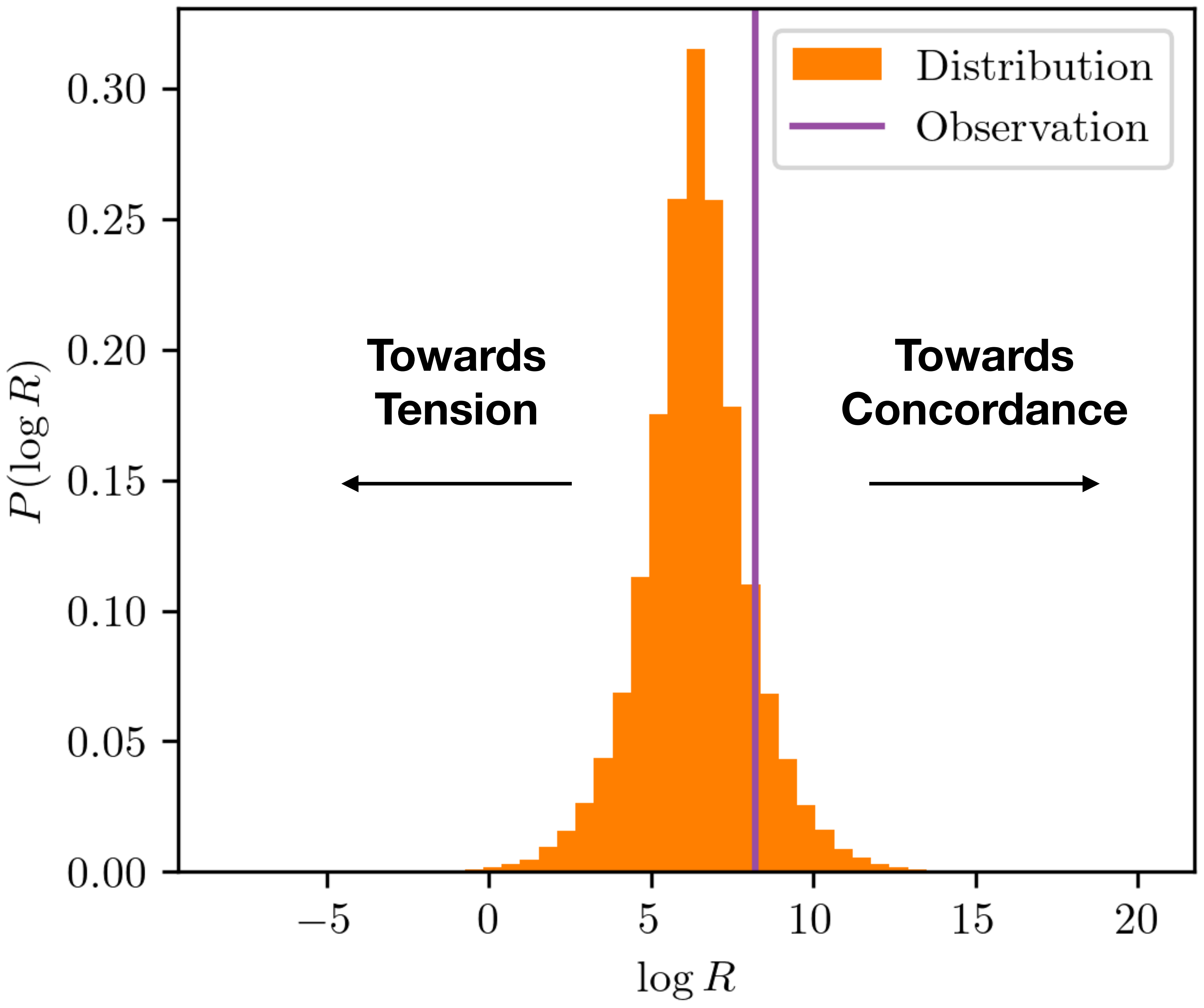
  - $R \ll 1 \rightarrow$ in tension



https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

Measuring tension





Calibrating with Neural Ratio Estimation

Neural Ratio Estimation

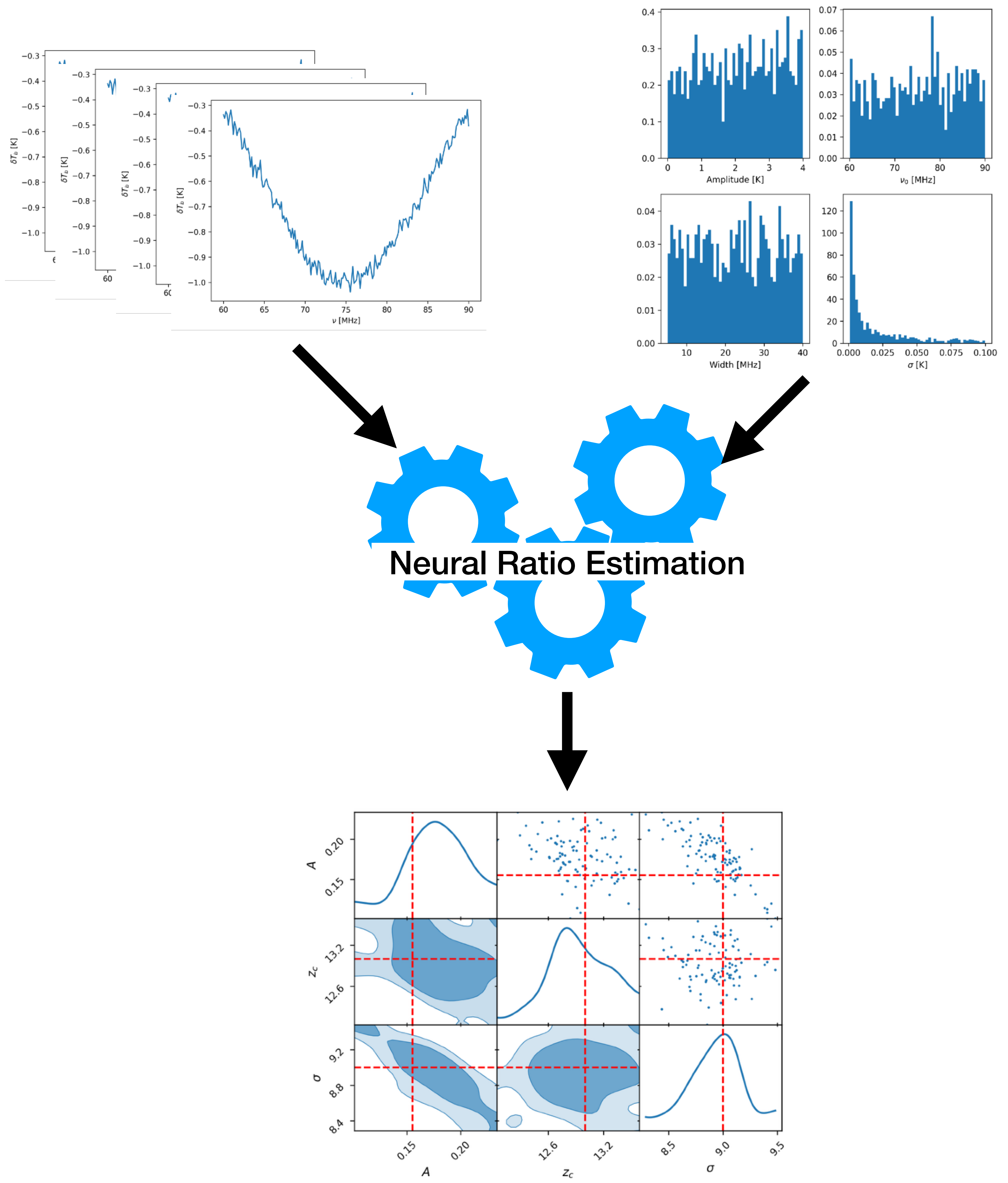
- Essentially just classifiers

- Take in two inputs A and B and estimate the probability that they are drawn from joint distribution vs disjoint

$$r = \frac{P(A, B)}{P(A)P(B)}$$

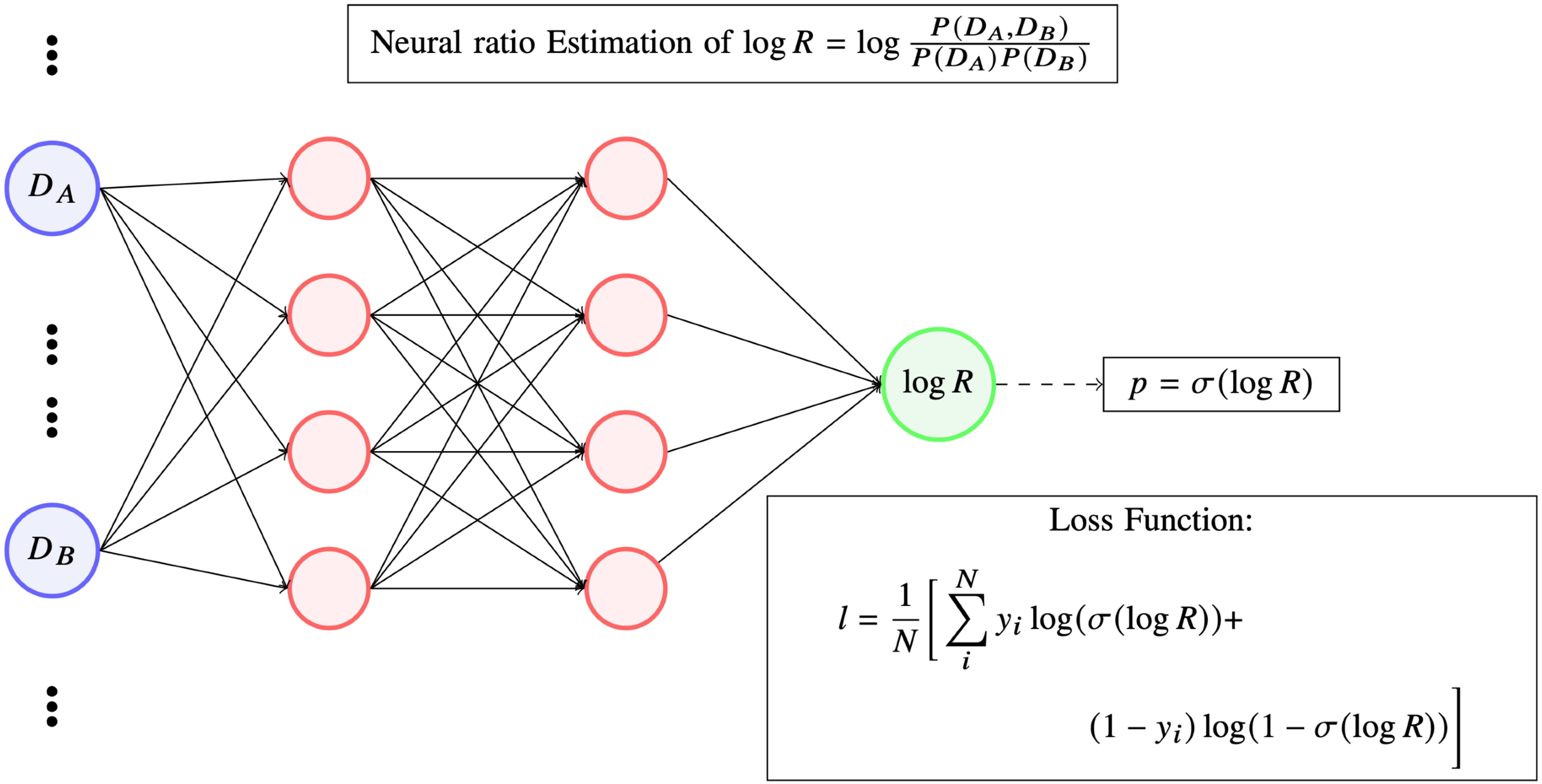
- Used for parameter inference

$$r = \frac{P(D, \theta)}{P(D)P(\theta)} = \frac{P(D\,|\,\theta)}{P(D)} = \frac{L(\theta)}{Z}$$



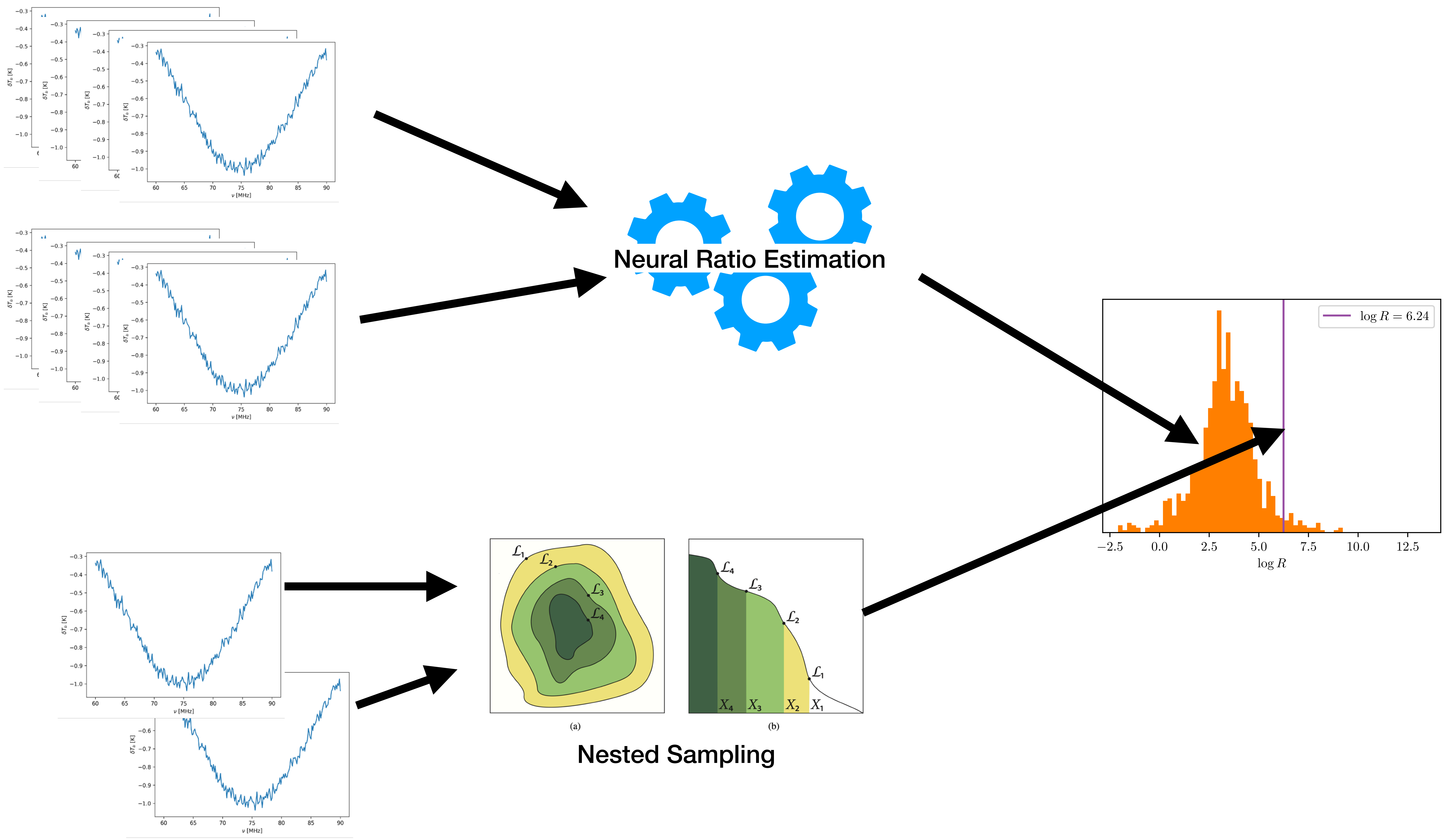
https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

R with NREs

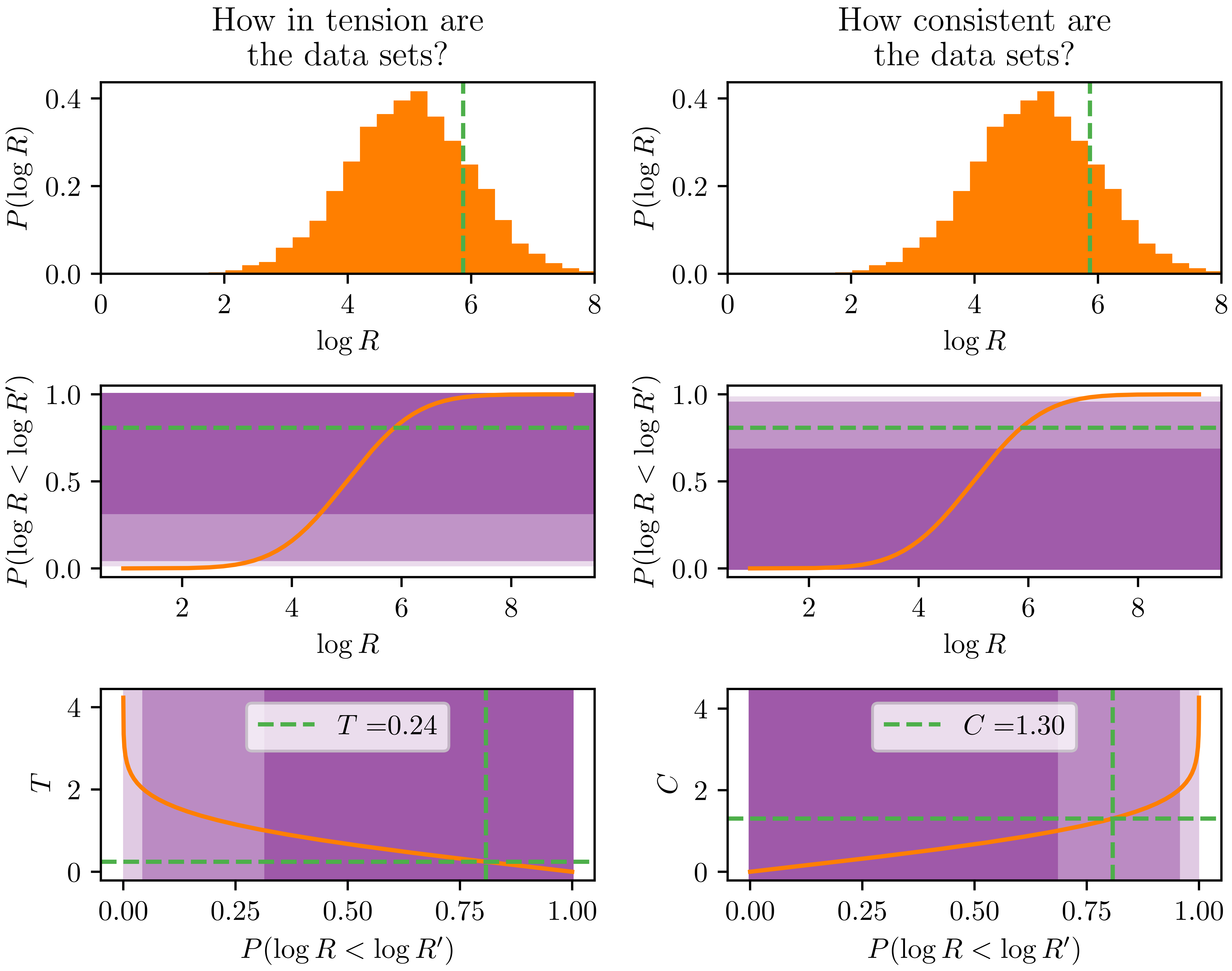


Neural ratio Estimation of $\log R = \log \frac{P(D_A,D_B)}{P(D_A)P(D_B)}$
D_A
D_B
$\log R$
$p = \sigma(\log R)$
Loss Function:
$l = \frac{1}{N}\left[\sum_i^N y_i \log(\sigma(\log R)) + (1-y_i)\log(1-\sigma(\log R))\right]$

Direct predictions or calibration?



Calibration of R



Examples

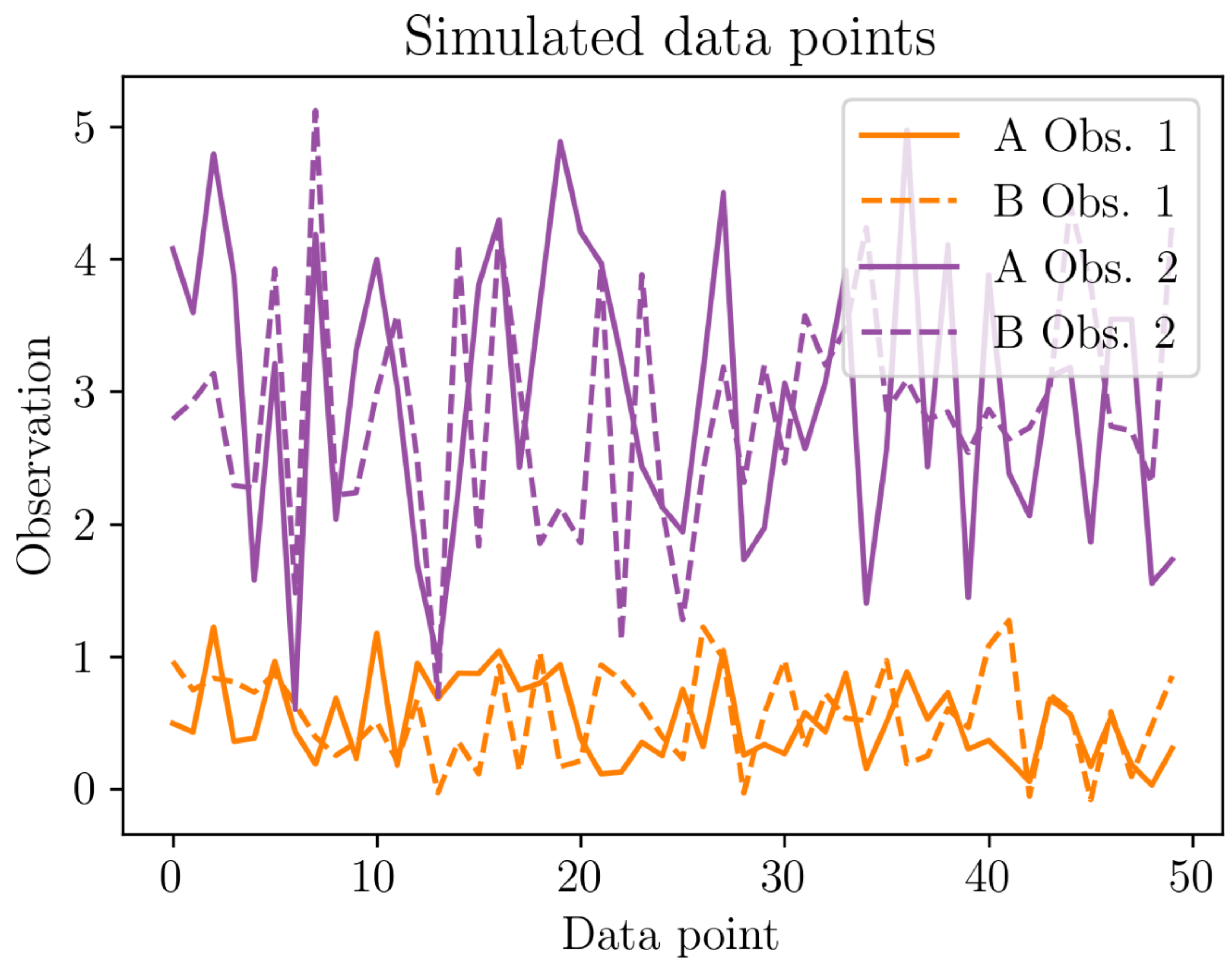
Analytic Example: Set Up

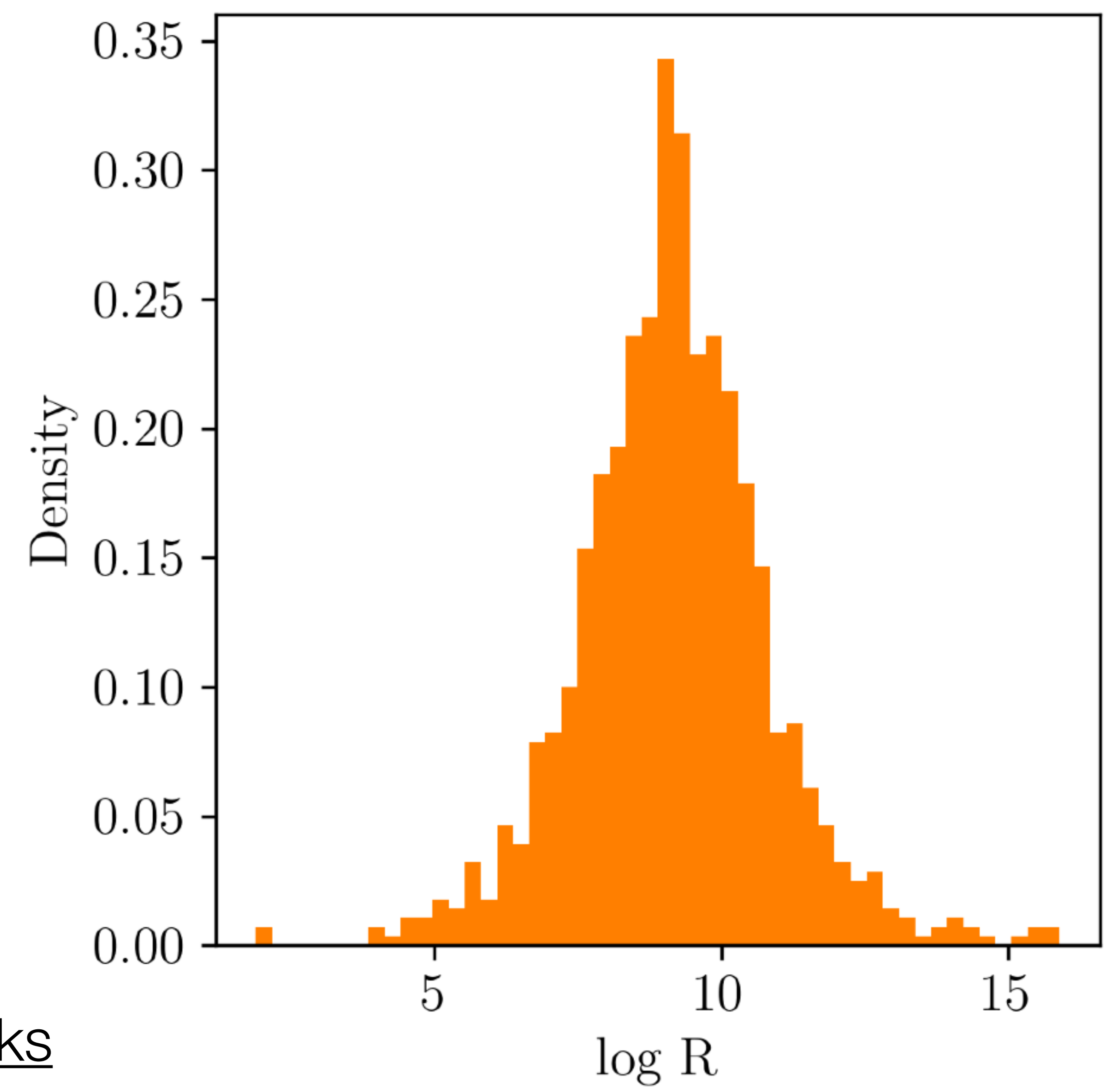
- Define a linear model

$$D_A = M_A\theta + m_A \pm \sqrt{C_A}$$

$$D_B = M_B\theta + m_B \pm \sqrt{C_B}$$

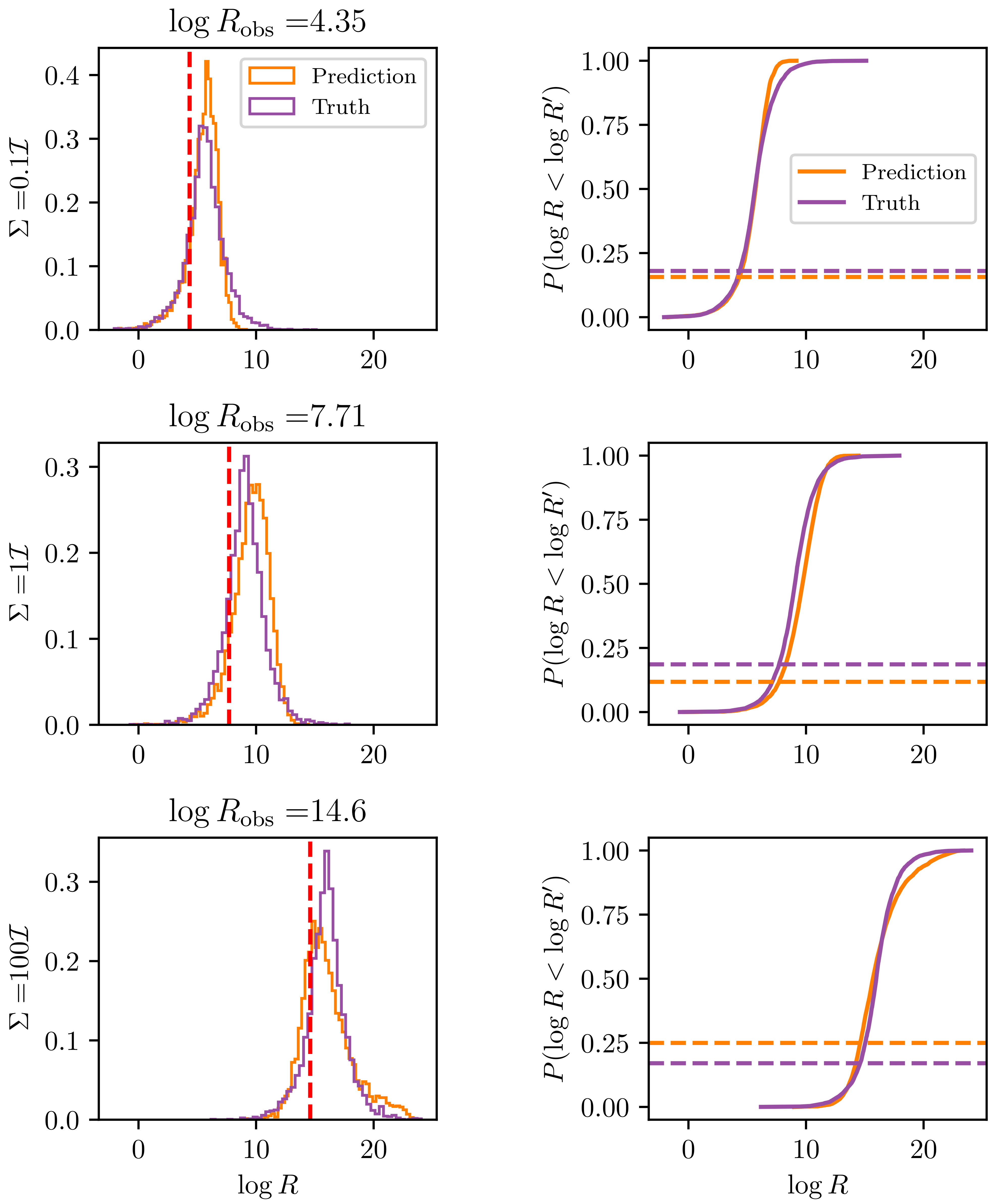
- $n_{dims} = 3, n_{data} = 50$
- Gaussian prior and likelihood
- Can analytically calculate $Z_A = P(D_A), Z_B$ and Z_{AB} and therefore get $\log R$
- Using lsbi package (https://github.com/handley-lab/lsbi)





https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

Analytic Example: Prior Dependence



	T	C
Truth	1.340	0.228
TENSIONNET	1.416	0.198

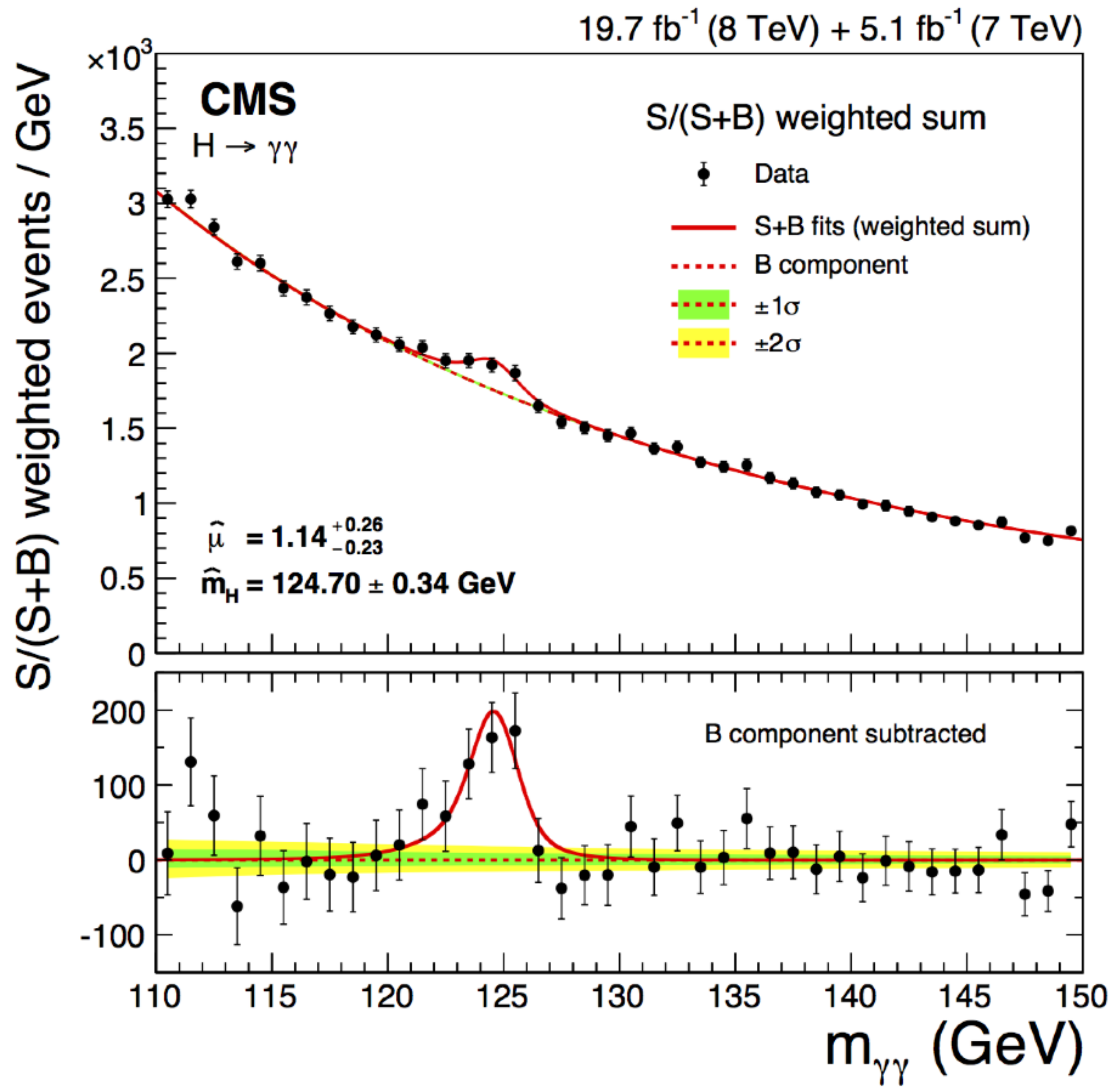
	T	C
Truth	1.323	0.235
TENSIONNET	1.563	0.148

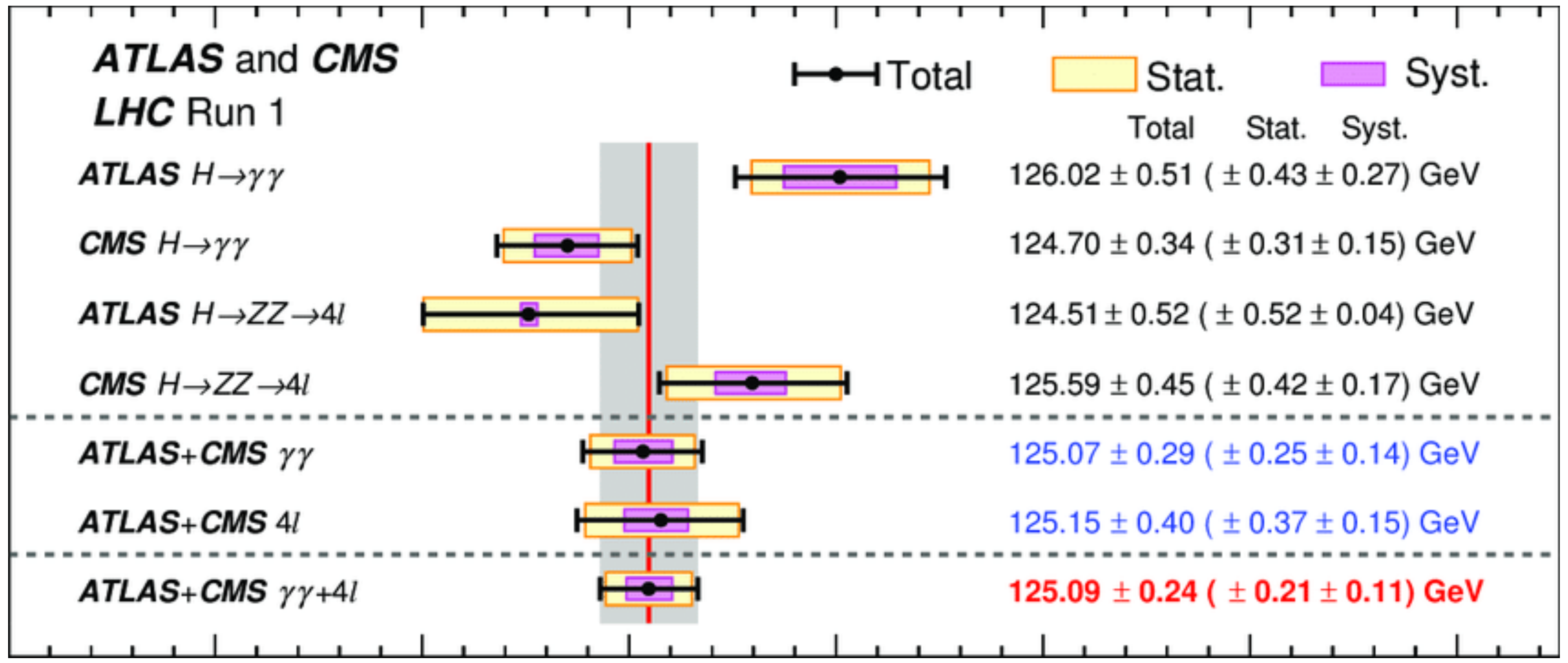
	T	C
Truth	1.371	0.215
TENSIONNET	1.151	0.318

https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

Bump Hunting

- Imagine two experiments recording excess events via some channel at a similar mass
- Obvious that the experiments are observing the same signal
- Want to quantify how well they agree
- For example run 1 measurements of Higgs Boson mass at ATLAS and CMS





Bump Hunting

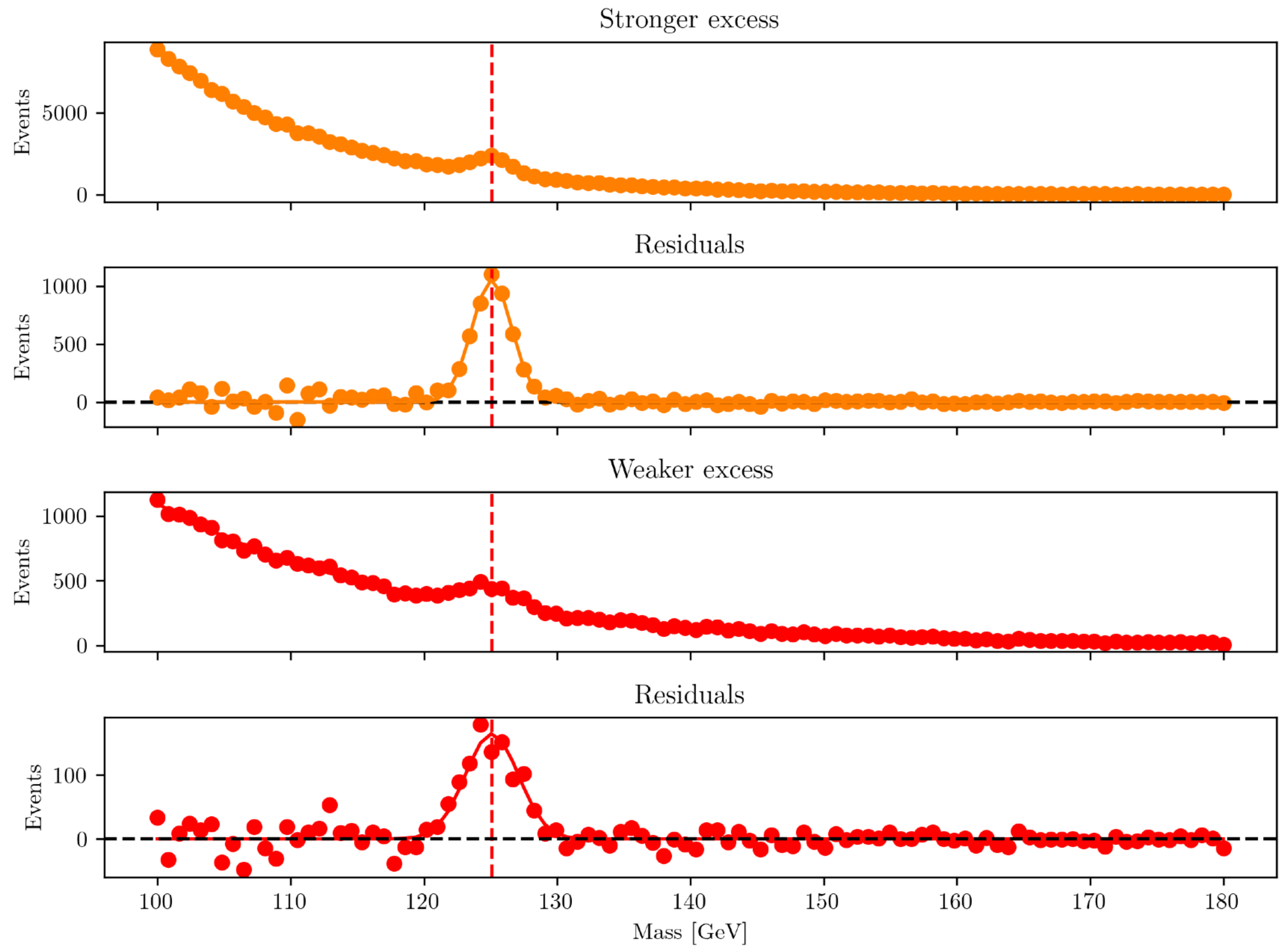
- Modelling background with

$$B(E) = \sum_{i-0}^{N=2} \theta_{1,i} \exp(-\theta_{2,i} E)$$

- And the excess as a gaussian centred around a mass of 125 GeV

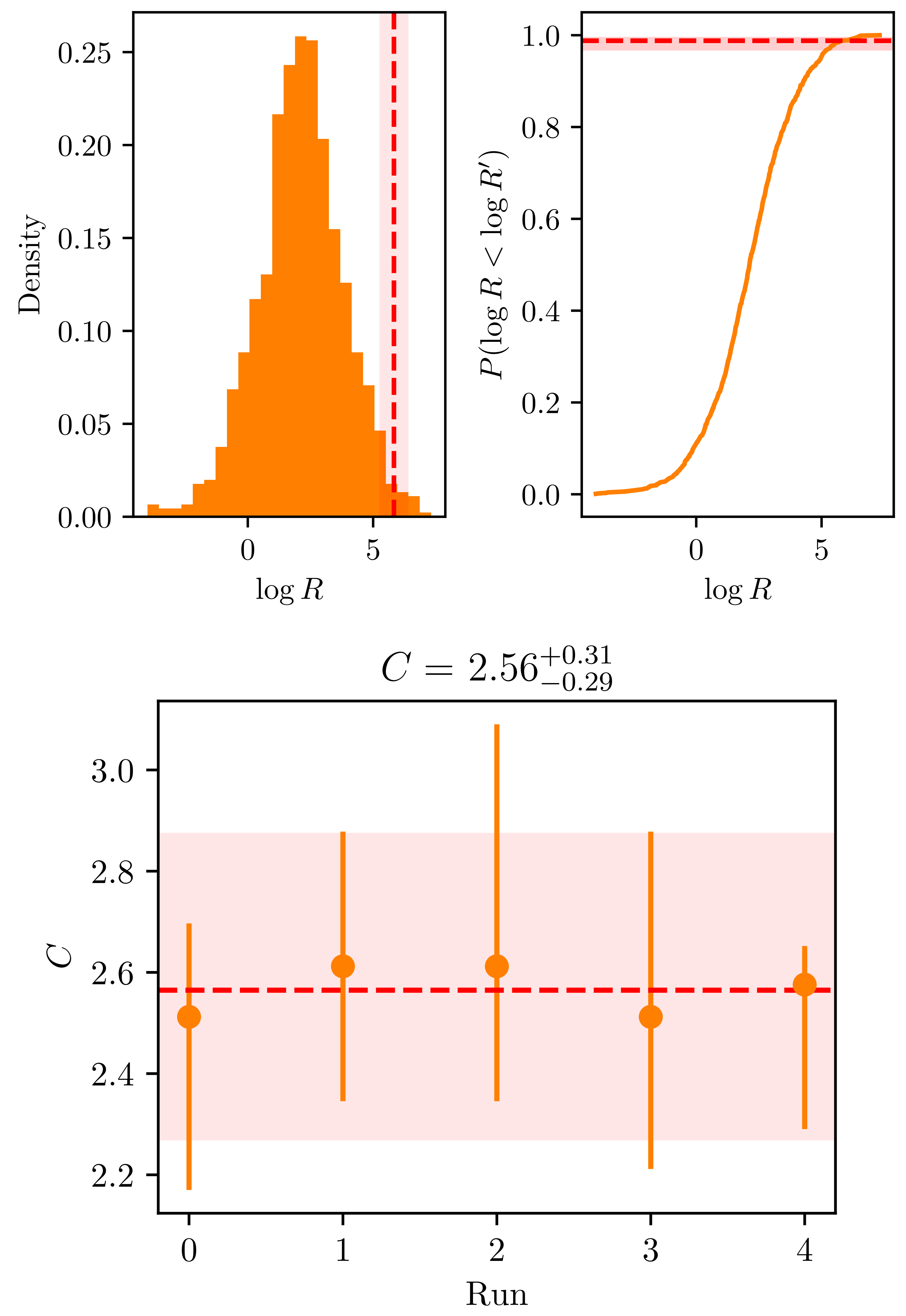
- Imagining two collider experiments observing excesses

- One with more events hence less noise and a greater confidence



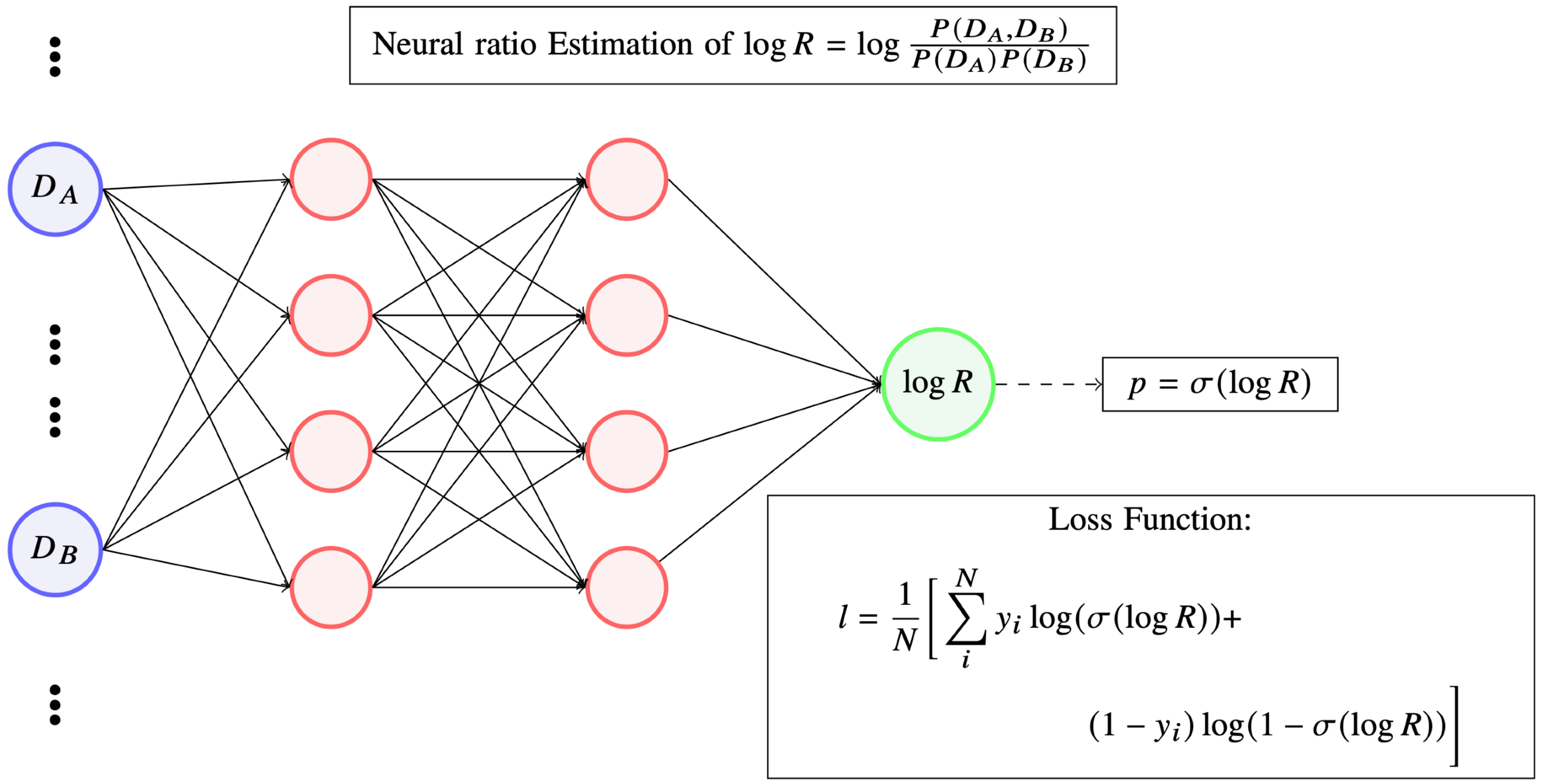
Bump Hunting

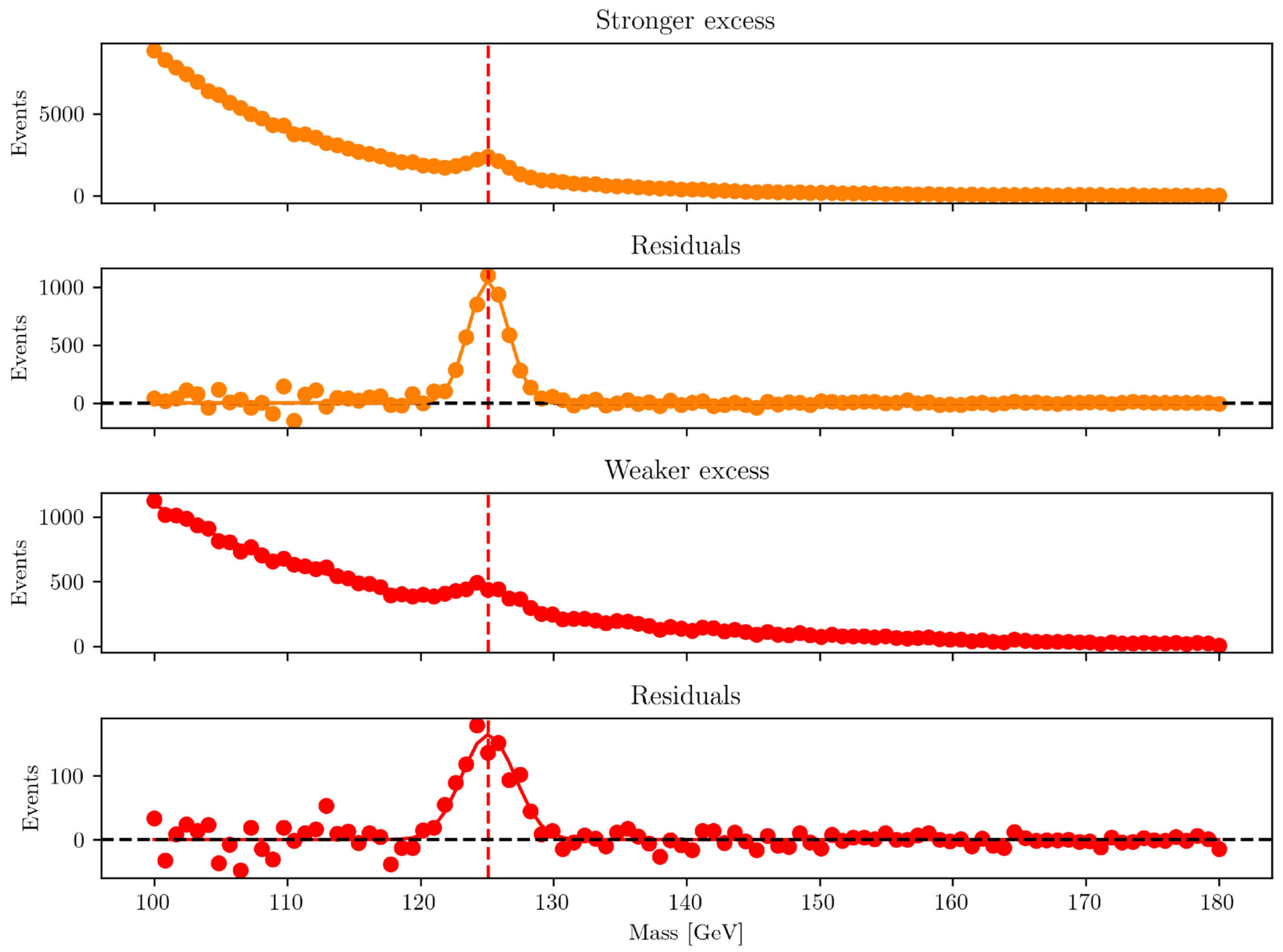
- Since I have an analytic model for the background and signal I can generate a range of simulations from a wide prior to train the NRE
- To calculate R_{obs} I use a product of Poisson distributions for my likelihood and the nested sampling algorithm
- Translating R_{obs} into units of σ concordance gives $C = 2.56^{+0.31}_{-0.29}$ averaged over five training runs



Conclusions

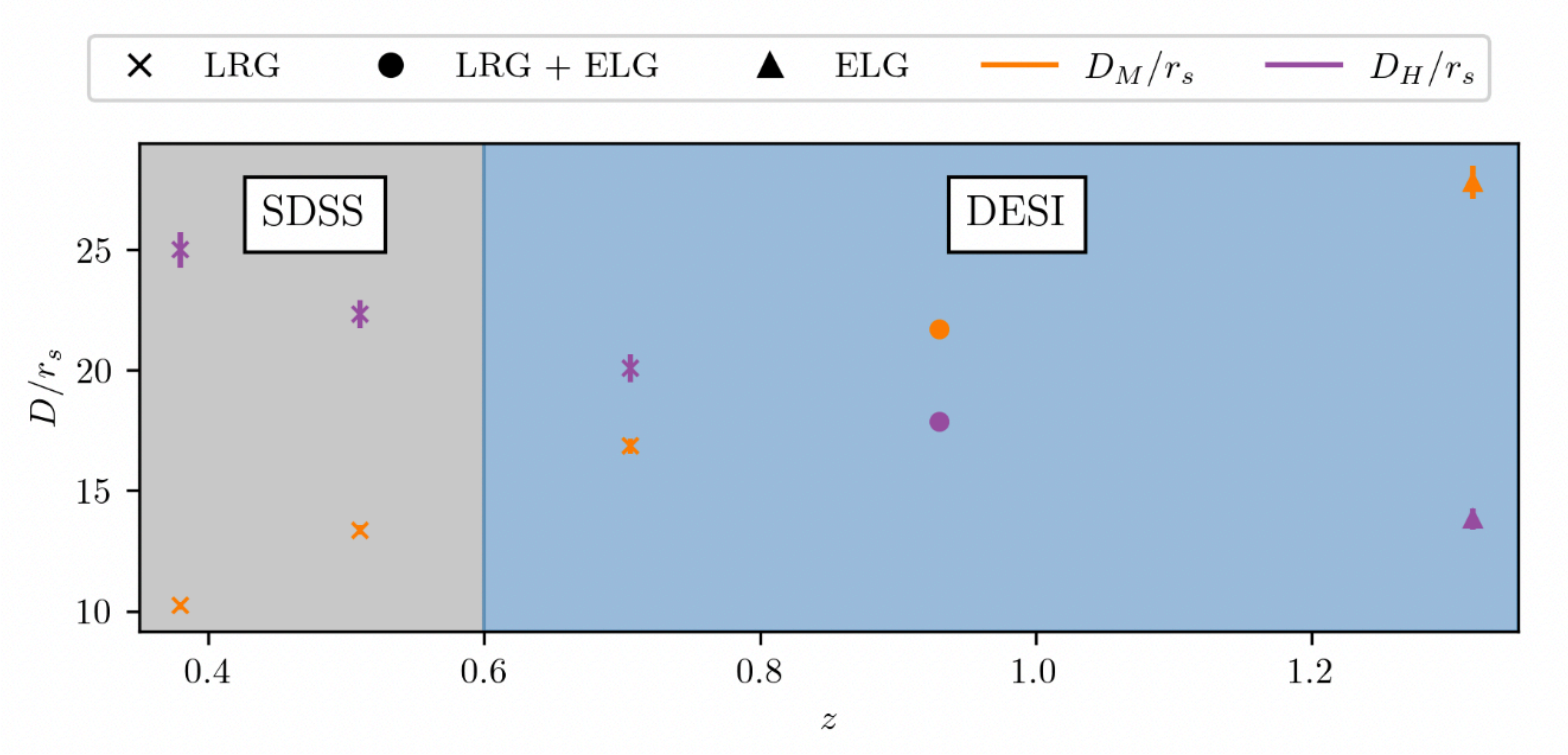
- Understanding tensions can help us identify new physics or instrumental systematics
- R statistic is an appropriately Bayesian choice
- We can use Neural Ratio Estimation to help us interpret the tension between different experiments
- Paper: arXiv:2407.15478
- Github: https://github.com/htjb/tension-networks





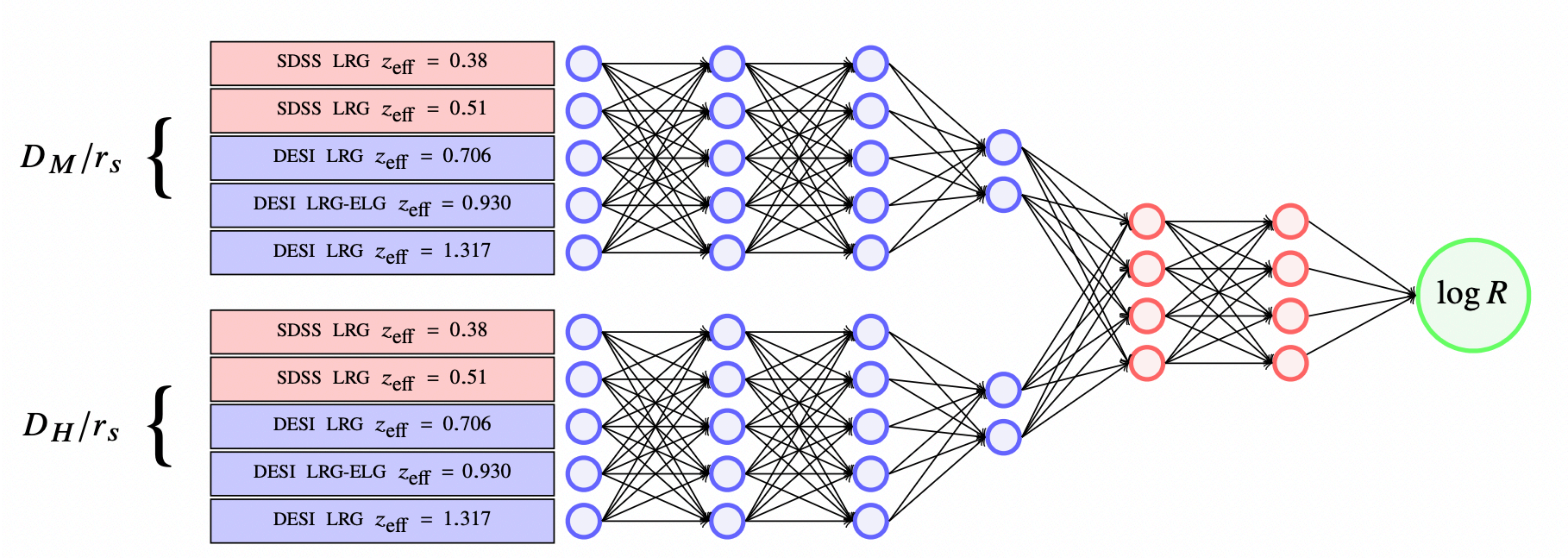
DESI + SDSS: Joint Data Set

- No existing correlated likelihood to evaluate a true R_{obs} with Nested Sampling
- Select different measurements from each survey to maximise the effective volume [e.g. 2404.03002]
- Focusing on LRG and ELG
- Add Quasars and Lyα in the future



https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

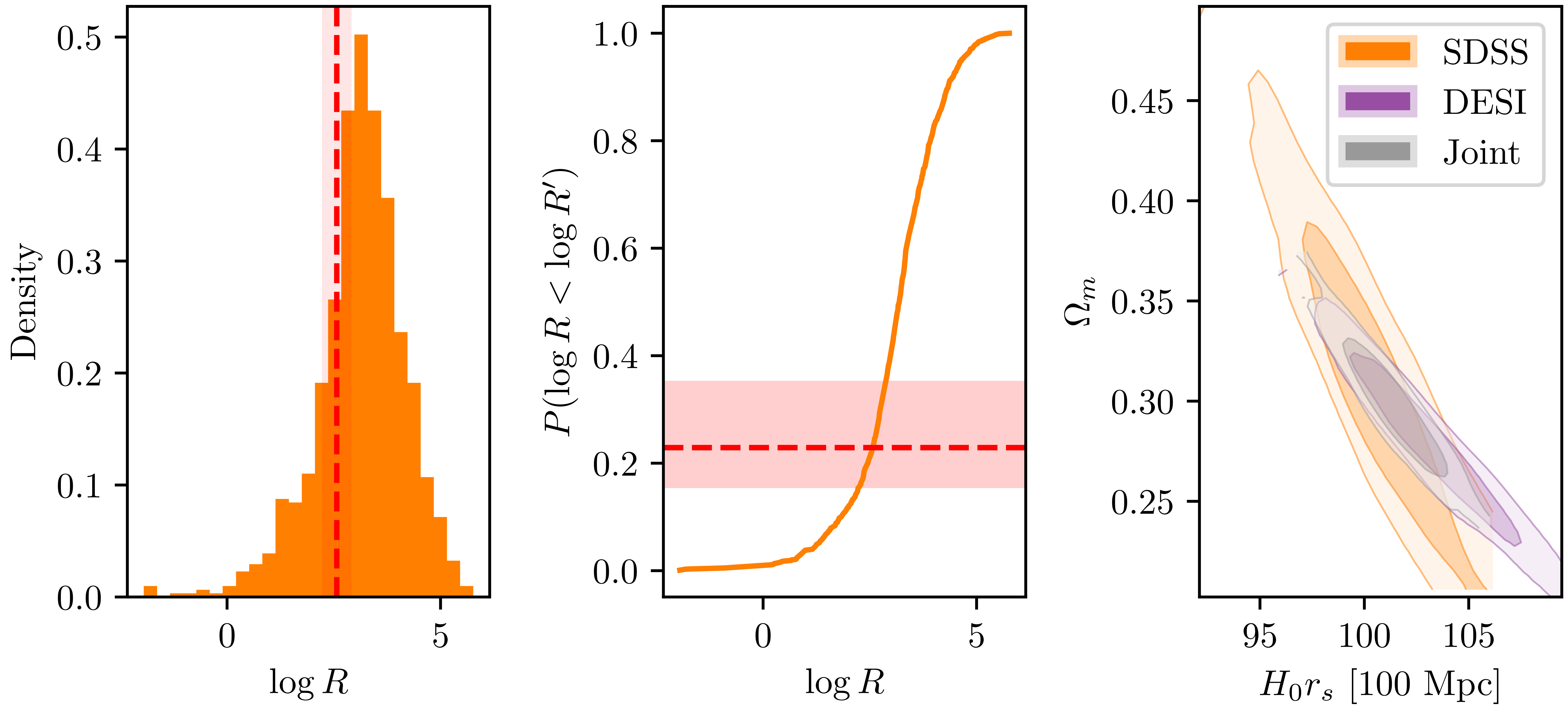
DESI + SDSS: NRE Set Up



https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478

DESI + SDSS: Results

- We find $T = 1.22 \pm 0.20$



https://github.com/htjb/tension-networks
https://arxiv.org/abs/2407.15478