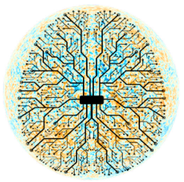
On the accuracy of posterior recovery with neural networks emulators

Harry Bevins
Kavli Fellow @ Kavli Institute for Cosmology, Cambridge
with
Thomas Gessey-Jones, Will Handley


The Importance of emulators

Inference in Cosmology

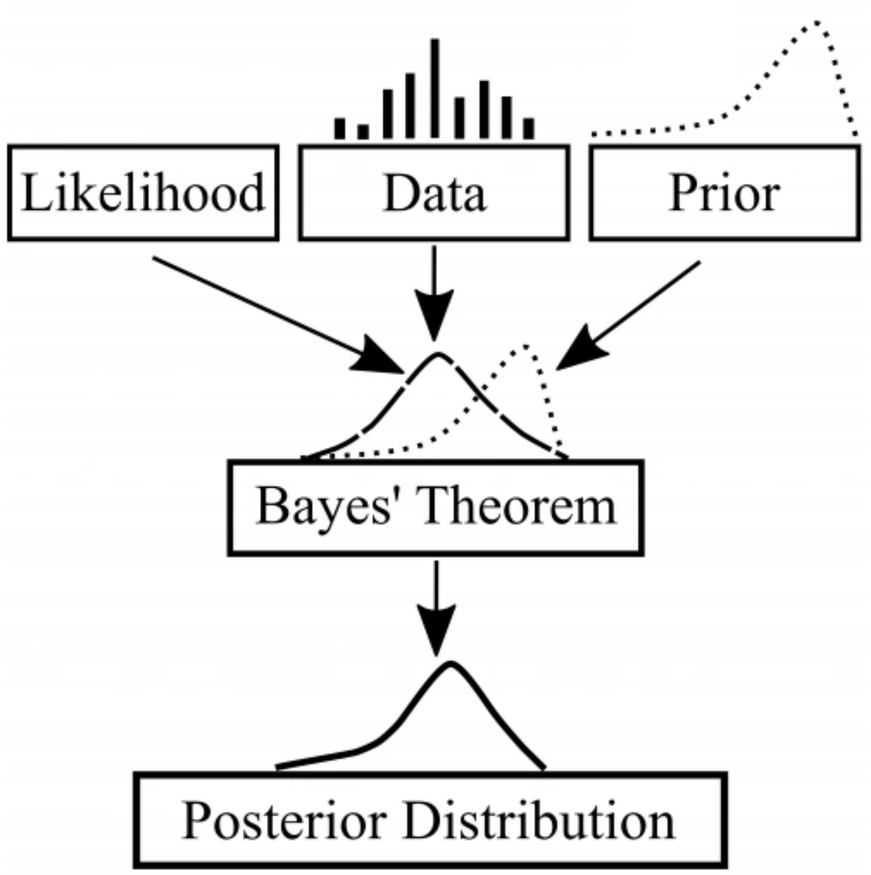
- Want to understand what our data tells us about a given parameterisation of the physics

- To fully understand modelling uncertainties and degeneracy between model parameters we use Bayesian inference

$$P(\theta \,|\, D, M) P(D \,|\, M) = P(D \,|\, \theta, M) P(\theta \,|\, M)$$

$$P\ Z = L\ \pi$$

where D is our data, θ are our model parameters and M is our model



Strong dependence on model runtime

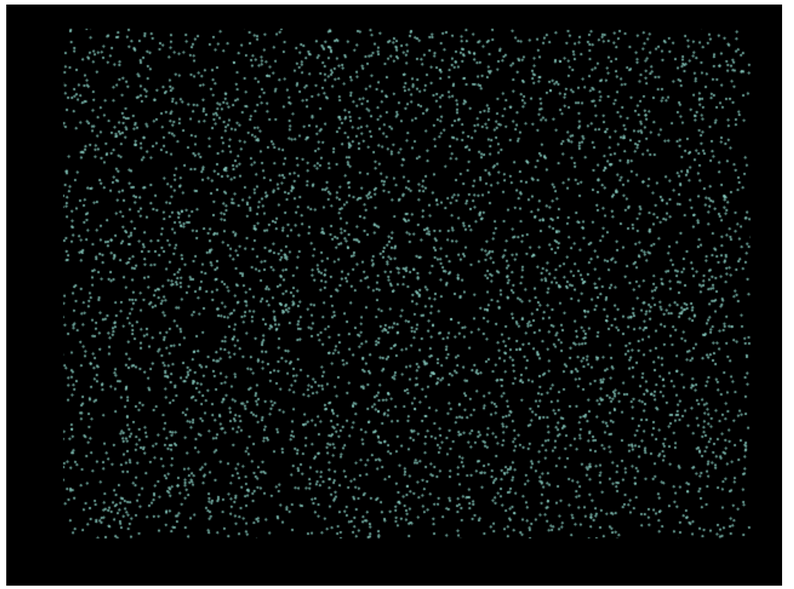
Inference runtime

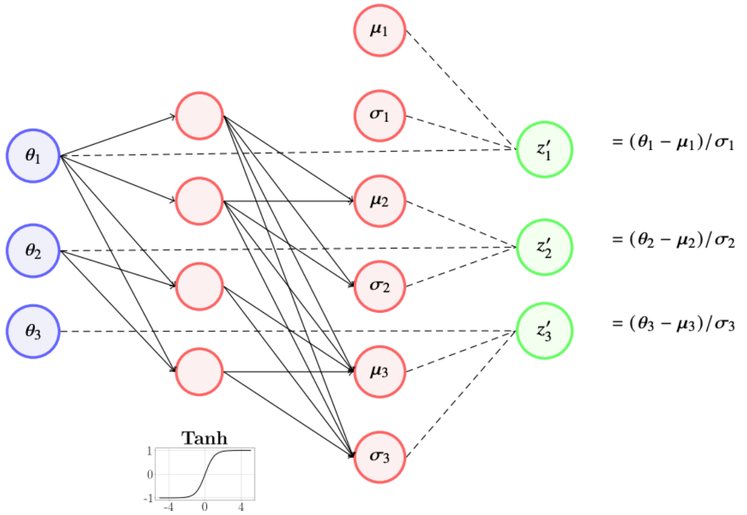
$$T \propto n_{\text{explorers}} \times \langle T\{L(M)\}\rangle \times \langle T\{\text{impl}\}\rangle \times D_{\text{KL}}(P\,||\,\pi)$$

$\langle T\{\text{impl}\}\rangle \rightarrow$ **BlackJAX NS (Yallup et al. 2025)**

$D_{KL}(P\,||\,\pi) \rightarrow$ **Better priors with normalising flows using margarine (Bevins et al 2022, 2023)**

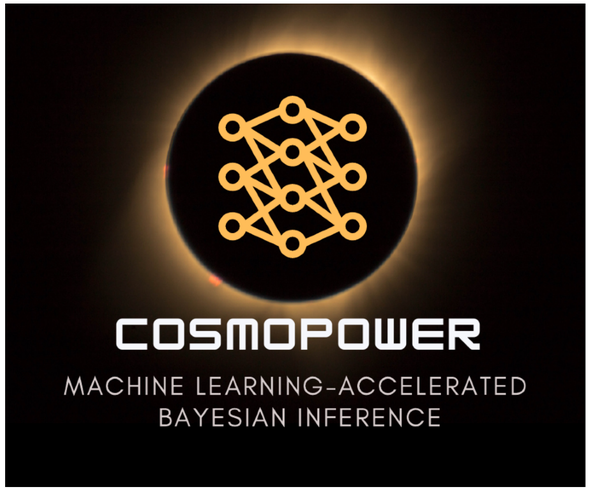
$\langle T\{L(M)\}\rangle \rightarrow$ **Emulators**

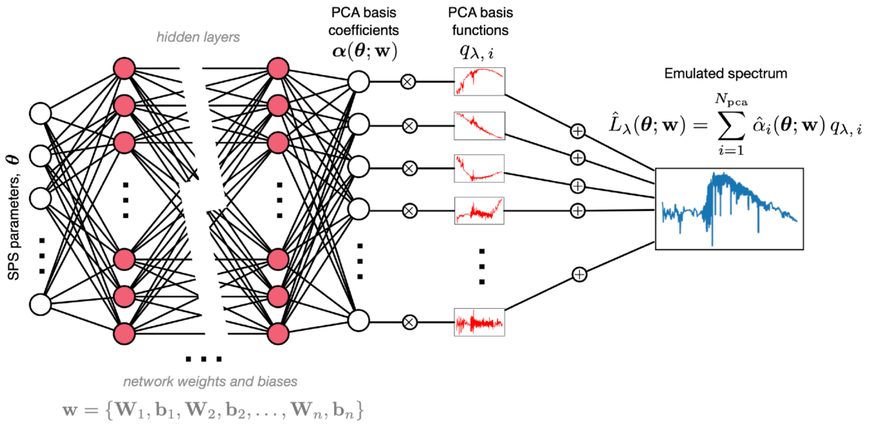




Emulators in Cosmology

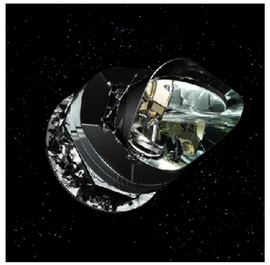
- Neural network emulators are really important in Cosmology and Astrophysics
- Modelling gravitational waves, CMB observables, galaxy SEDs, 21-cm signal etc is very complicated!
- We want to build an approximation of these models that we can rapidly evaluate
- $M_E \approx f_\phi(\theta)$ where ϕ are the ML Parameters
- Allow for fast inference on computationally expensive models that might not otherwise be possible

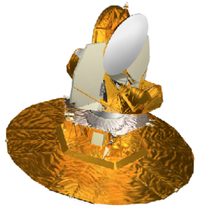


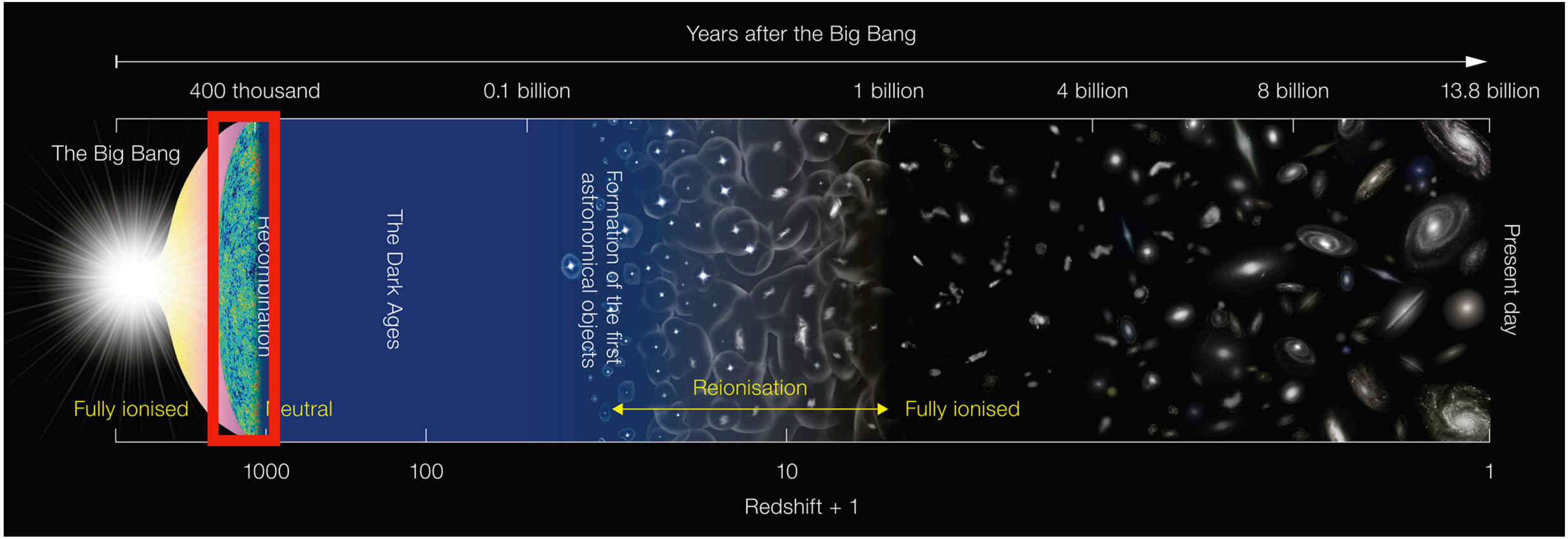


The Infant Universe and 21-cm Cosmology

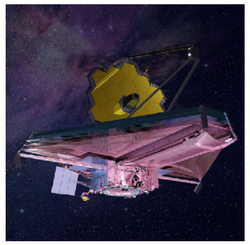
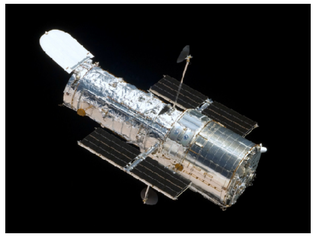
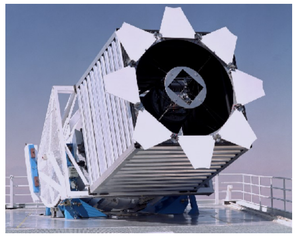
A brief history of the universe

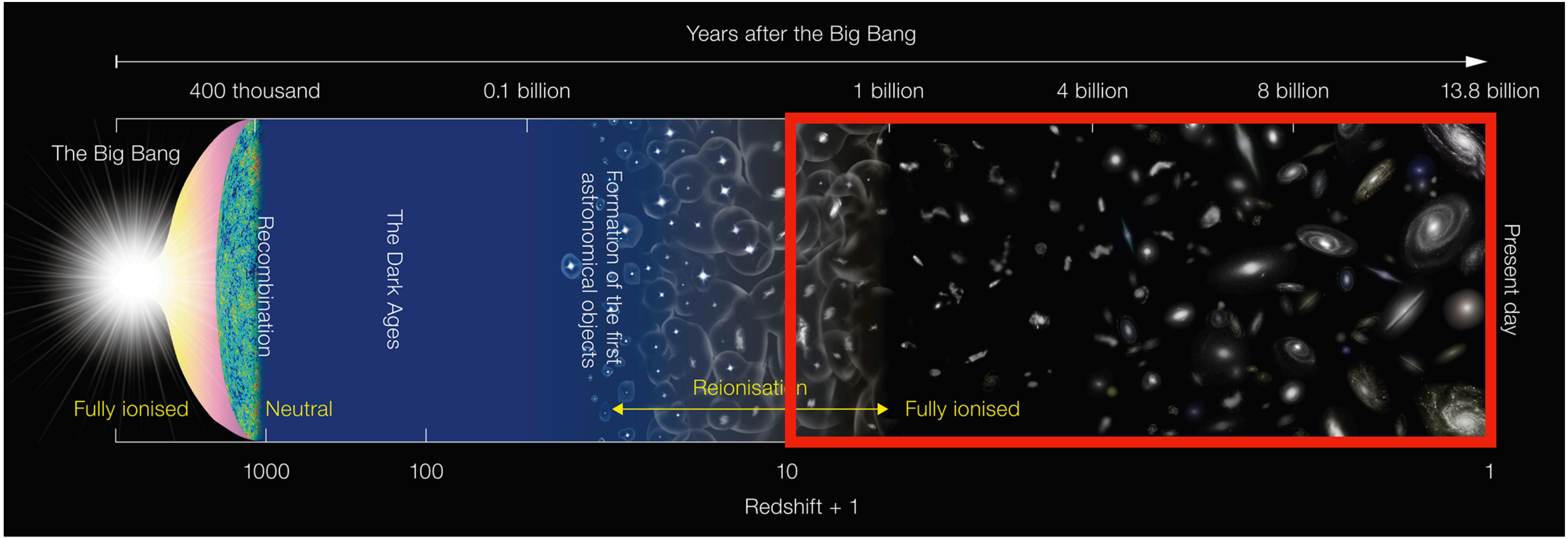




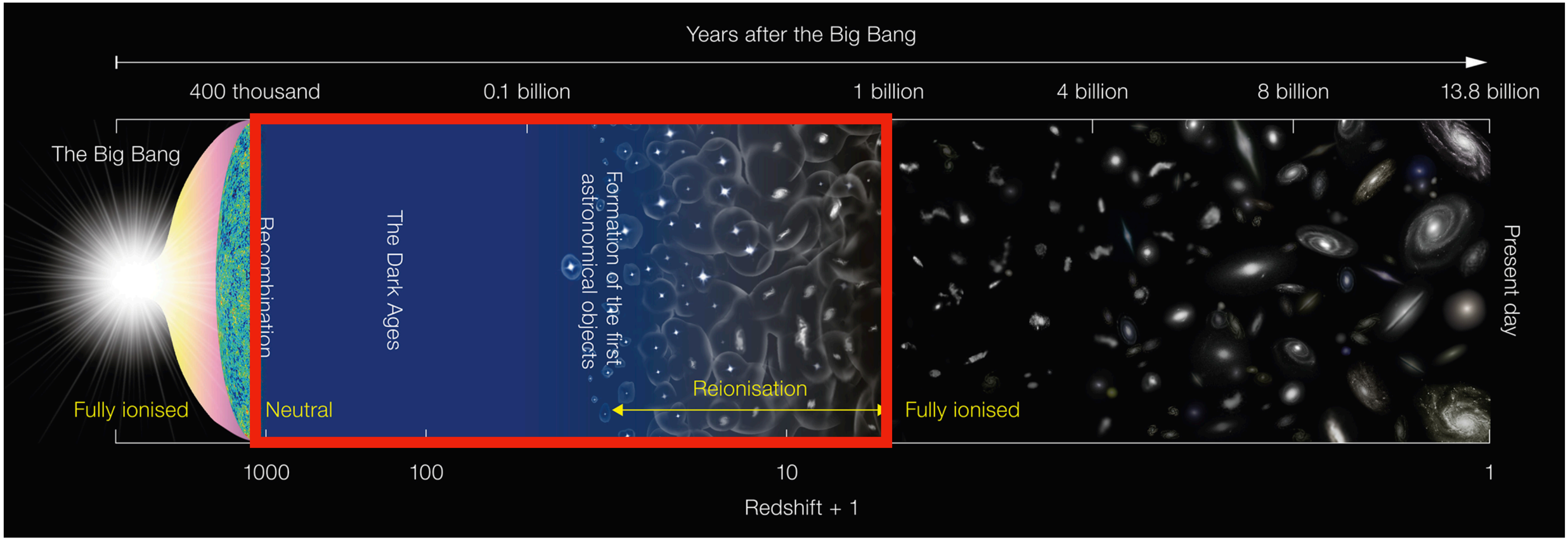


A brief history of the universe

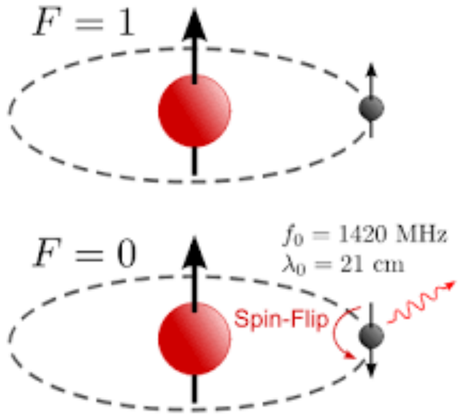


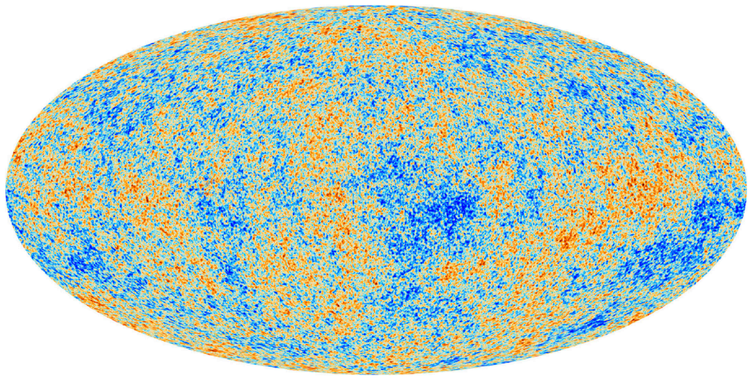
A brief history of the universe



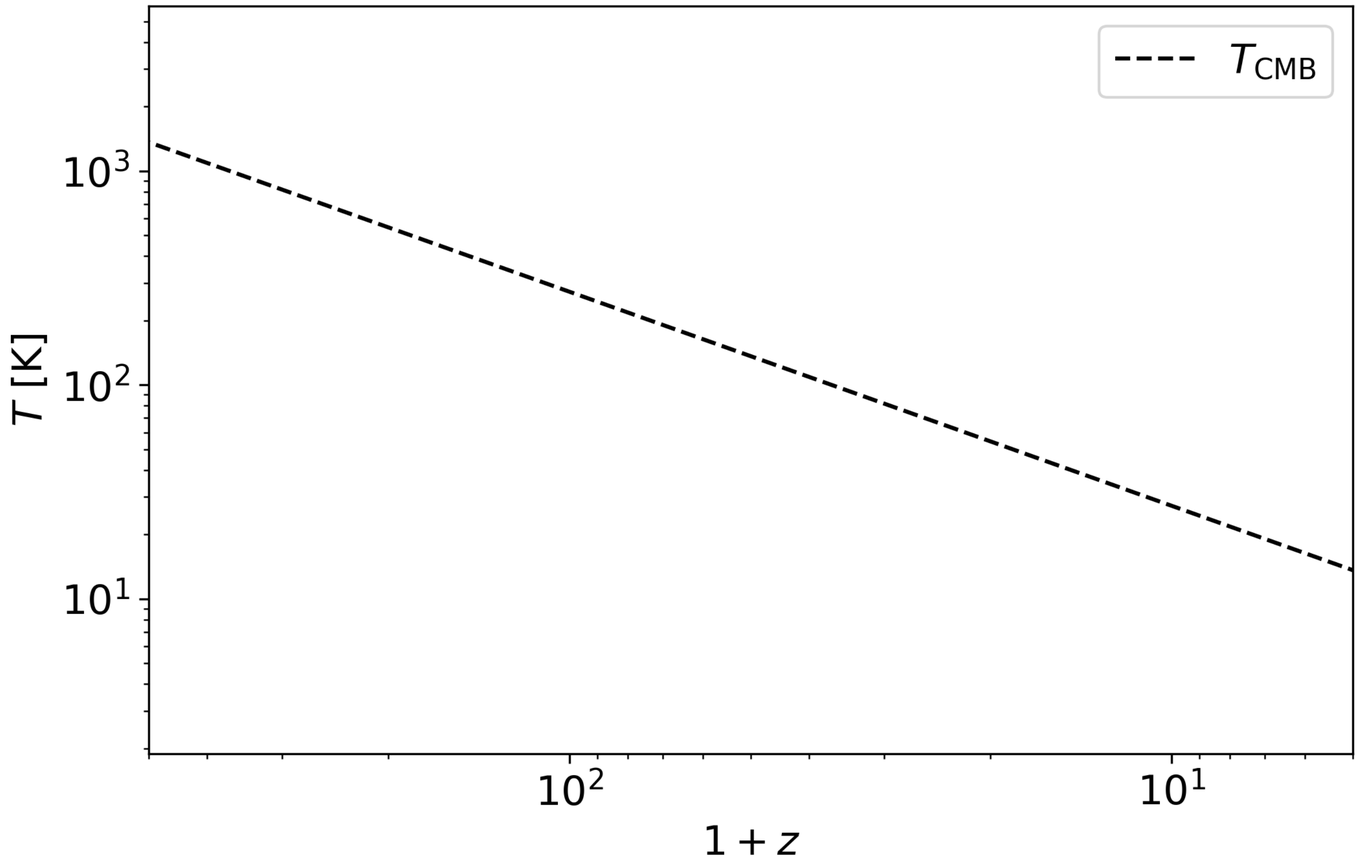
21-cm Cosmology

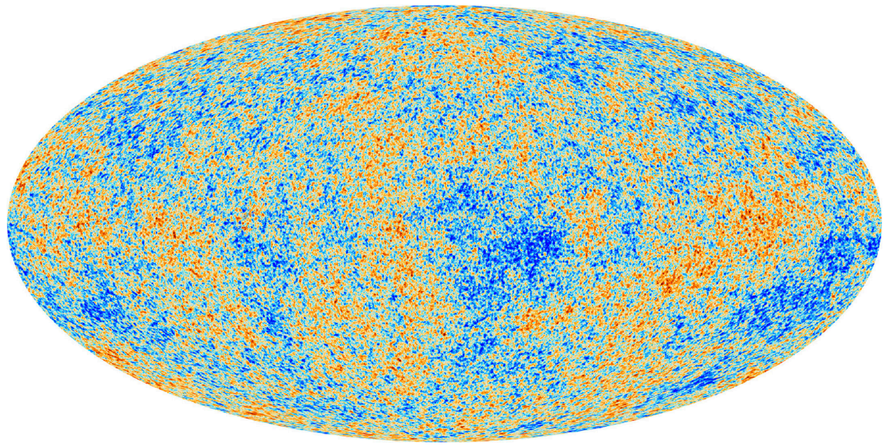
- Spin-flip transition in neutral hydrogen
- Forbidden transition that can't be seen in the lab
- Define the spin temperature
- Measure relative to the radio background
- To understand the importance of the spin temperature we look at the thermal history of the universe



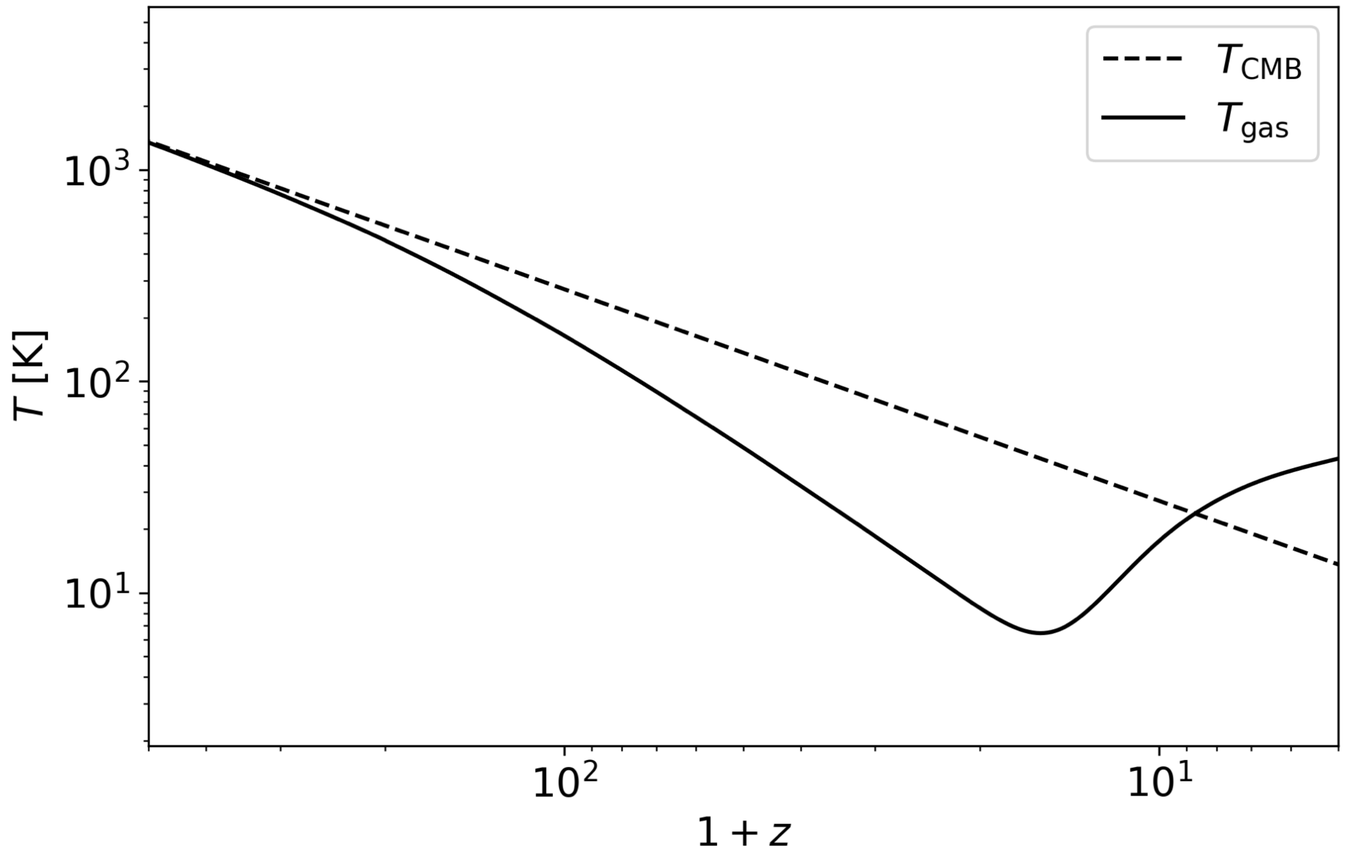


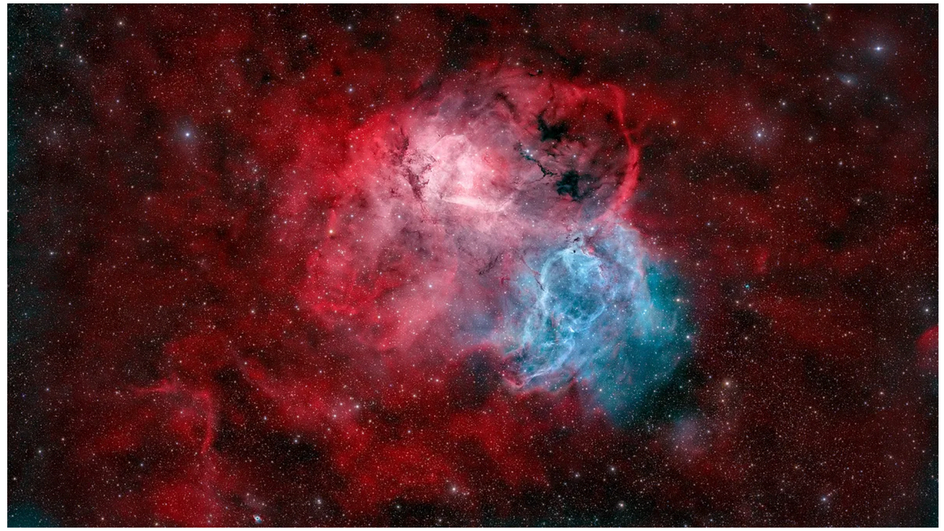
21-cm Cosmology



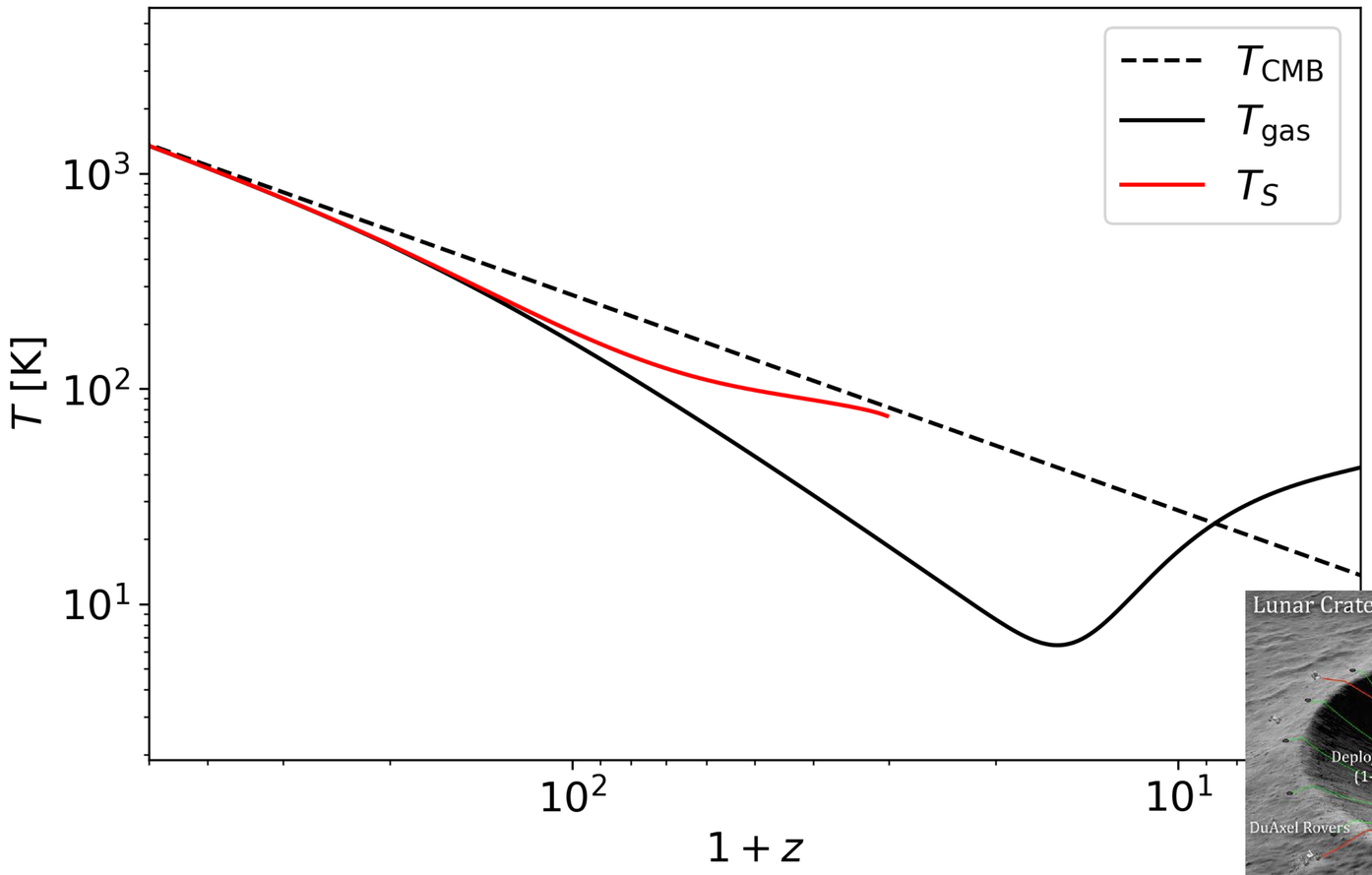


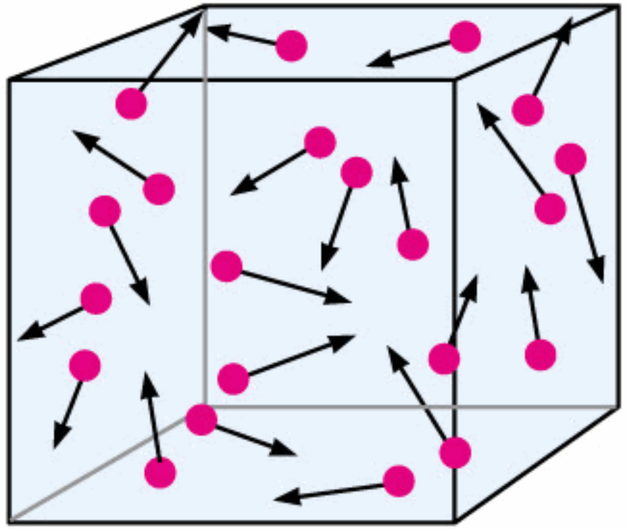
21-cm Cosmology

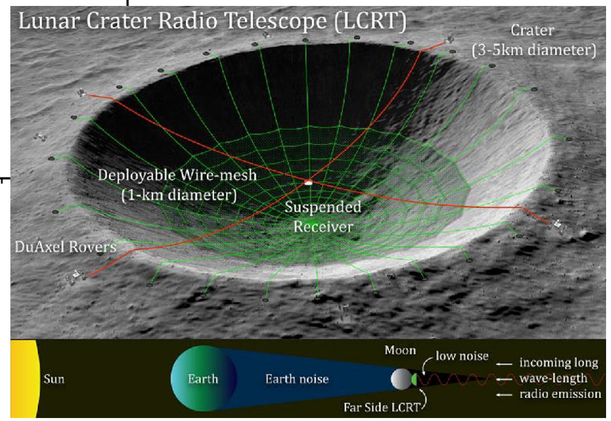


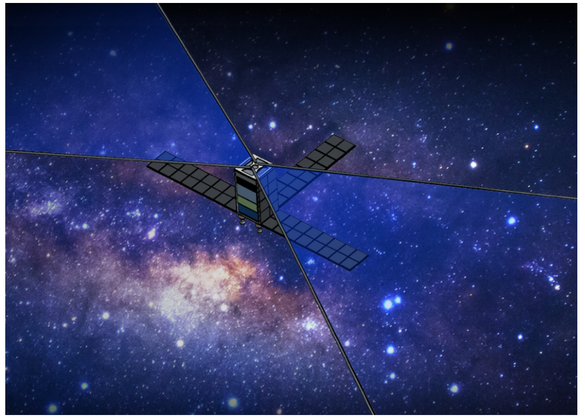


Dark Ages ($z \lesssim 30; t \lesssim 0.5$ Gyr)



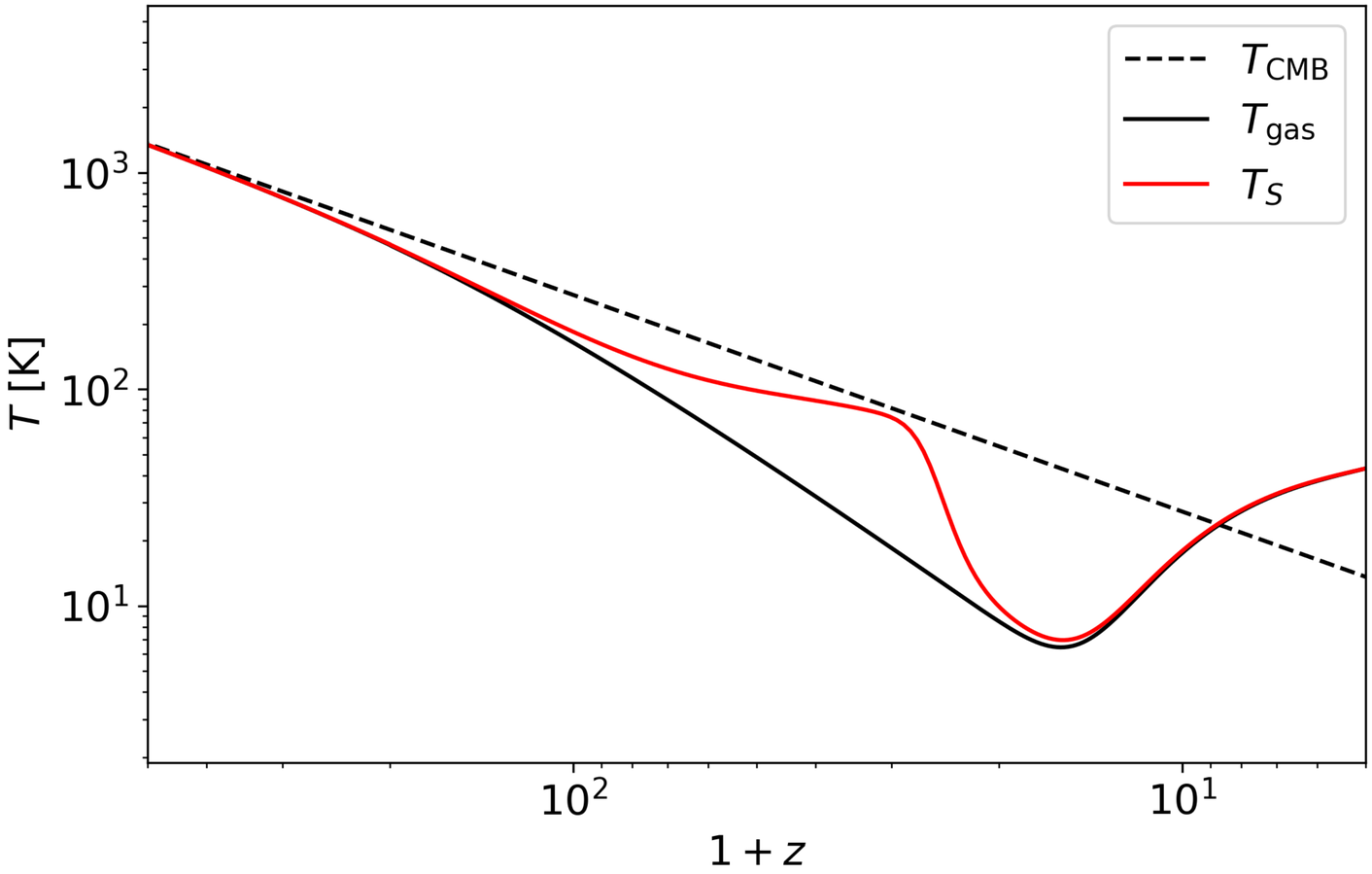


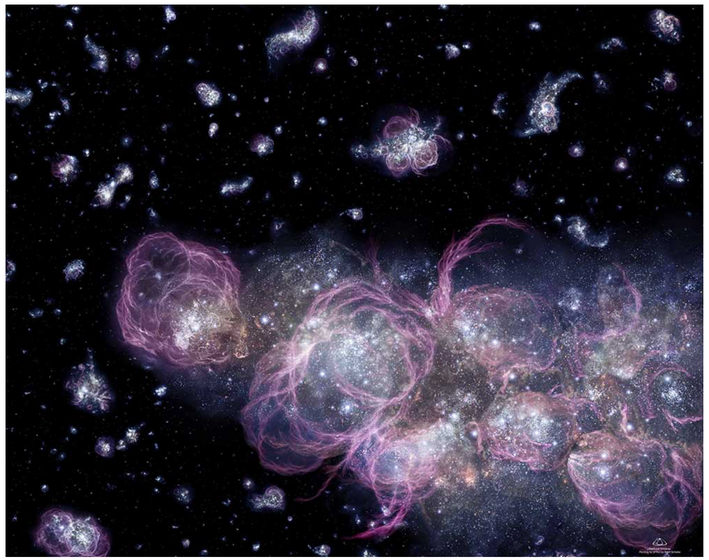


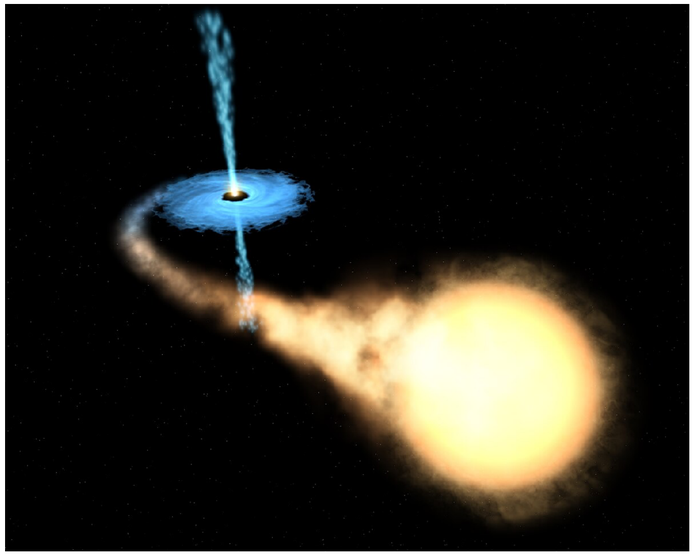


Cosmic Dawn and Reionisation

$(30 \lesssim z \lesssim 5; t = 0.5 - 12.5\ \mathrm{Gyr})$

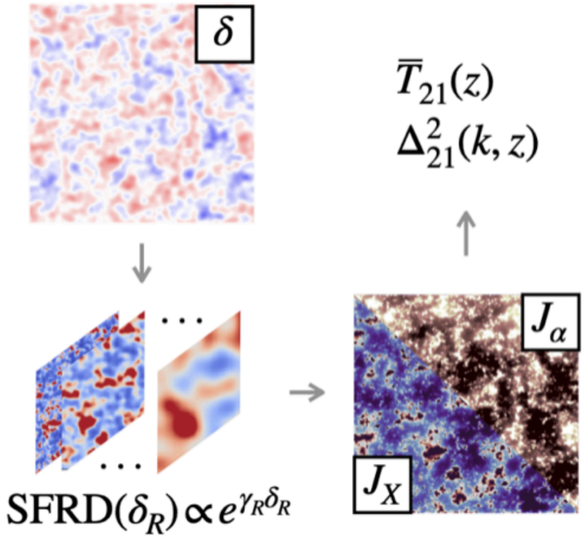


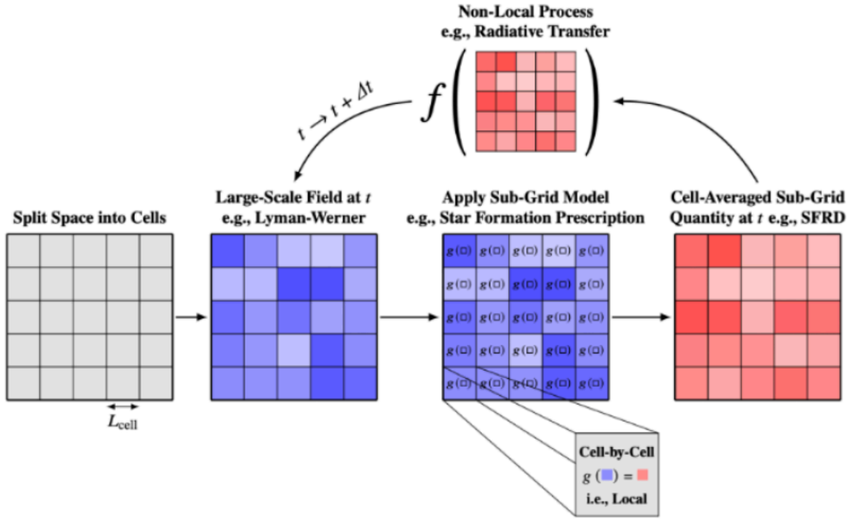




Simulating the 21-cm signal

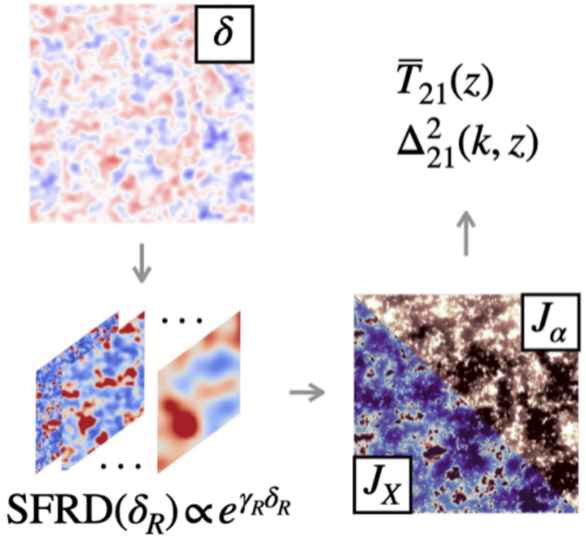
- Several classes of simulation
  - Analytic models (Zeus21, Echo21…)
    - Make approximation
    - Evaluate in seconds
  - Semi-numerical models (21cmSPACE, 21cmFAST…)
    - Detailed grid model populated with halos and galaxies
    - Evaluate in hours
  - Hydrodynamical codes (C2-RAY…)
    - Radiative transfer codes with hydrodynamics and feedback
    - Evaluate in days to weeks

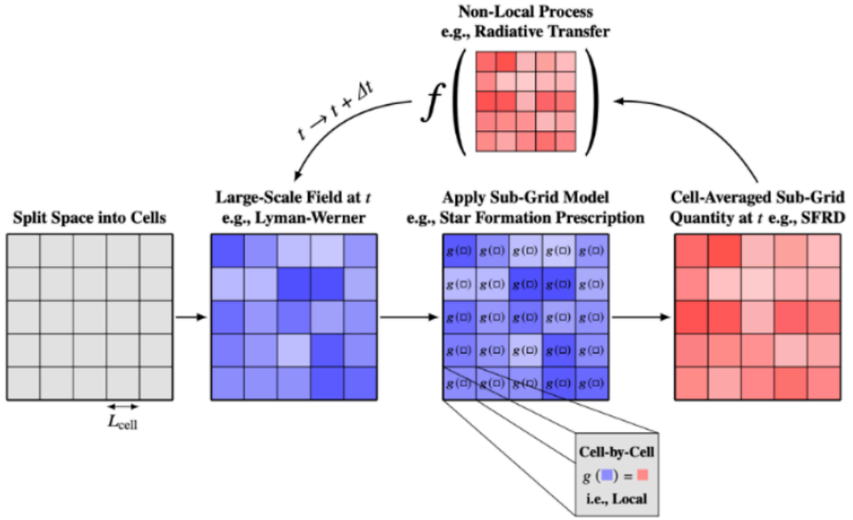




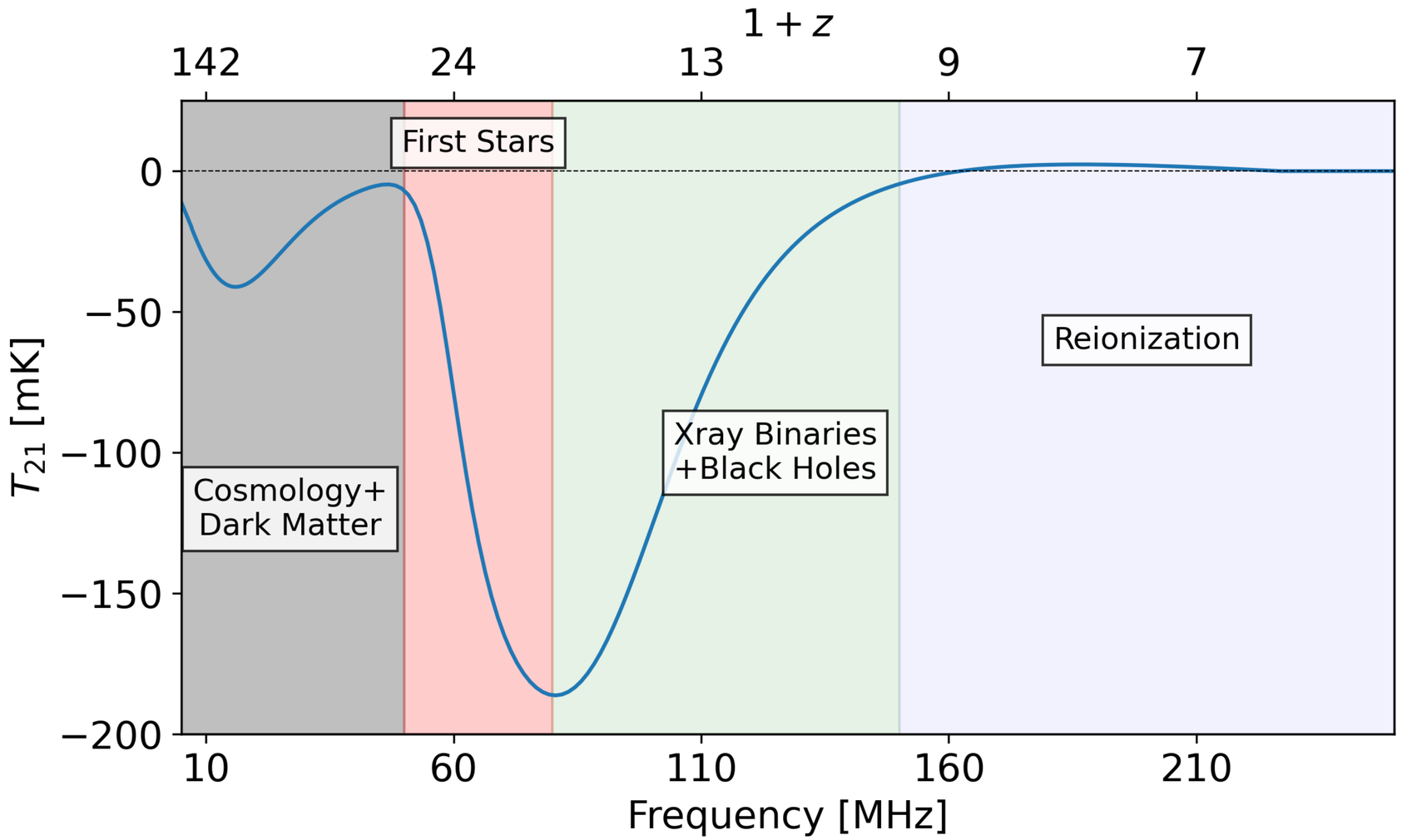
Simulating the 21-cm signal

- Several classes of simulation
  - Analytic models (Zeus21, Echo21…)
    - Make approximation
    - Evaluate in seconds
  - Semi-numerical models (21cmSPACE, 21cmFAST…)
    - Detailed grid model populated with halos and galaxies
    - Evaluate in hours
  - Hydrodynamical codes (C2-RAY…)
    - Radiative transfer codes with hydrodynamics and feedback
    - Evaluate in days to weeks



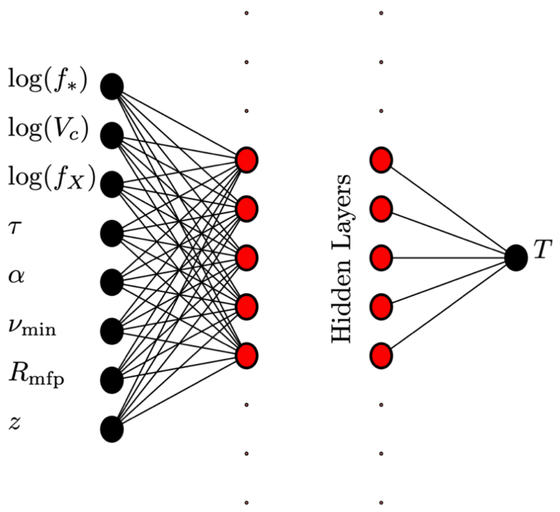


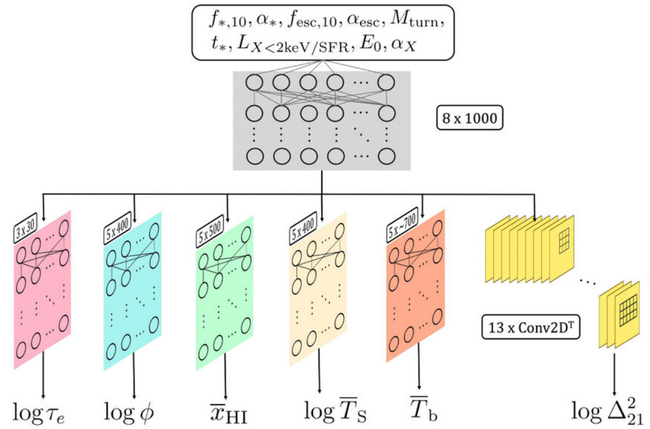
The sky-averaged 21-cm signal



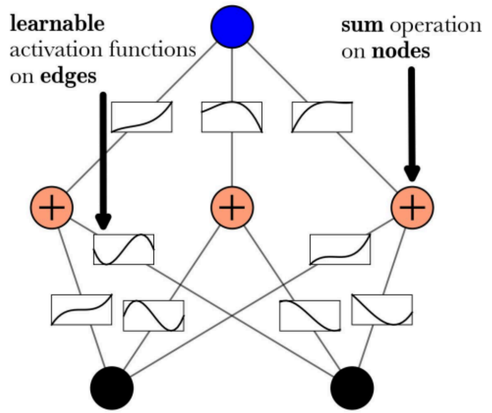
Emulators in 21cm Cosmology

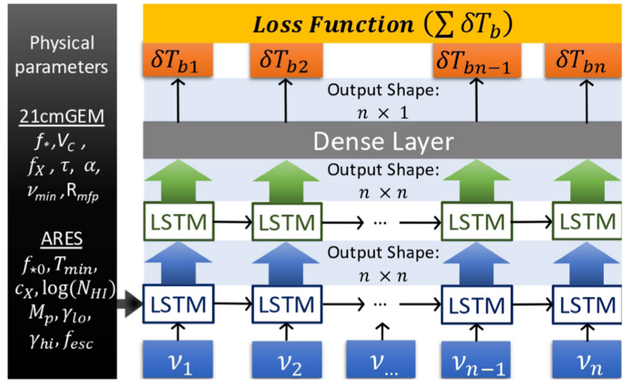
- globalemu [Bevins+ 2021]
  - Small and fast feedforward
- 21cmVAE [Bye+ 2021]
  - Variational Autoencoder
- 21cmEMU [Breitman+ 2023]
  - Lots of auxiliary tasks trained together
- 21cmLSTM [Dorigo Jones+2024]
  - Long and short term memory
- 21cmKAN [Dorigo Jones+2025]
  - Kolmogorov Arnold Networks



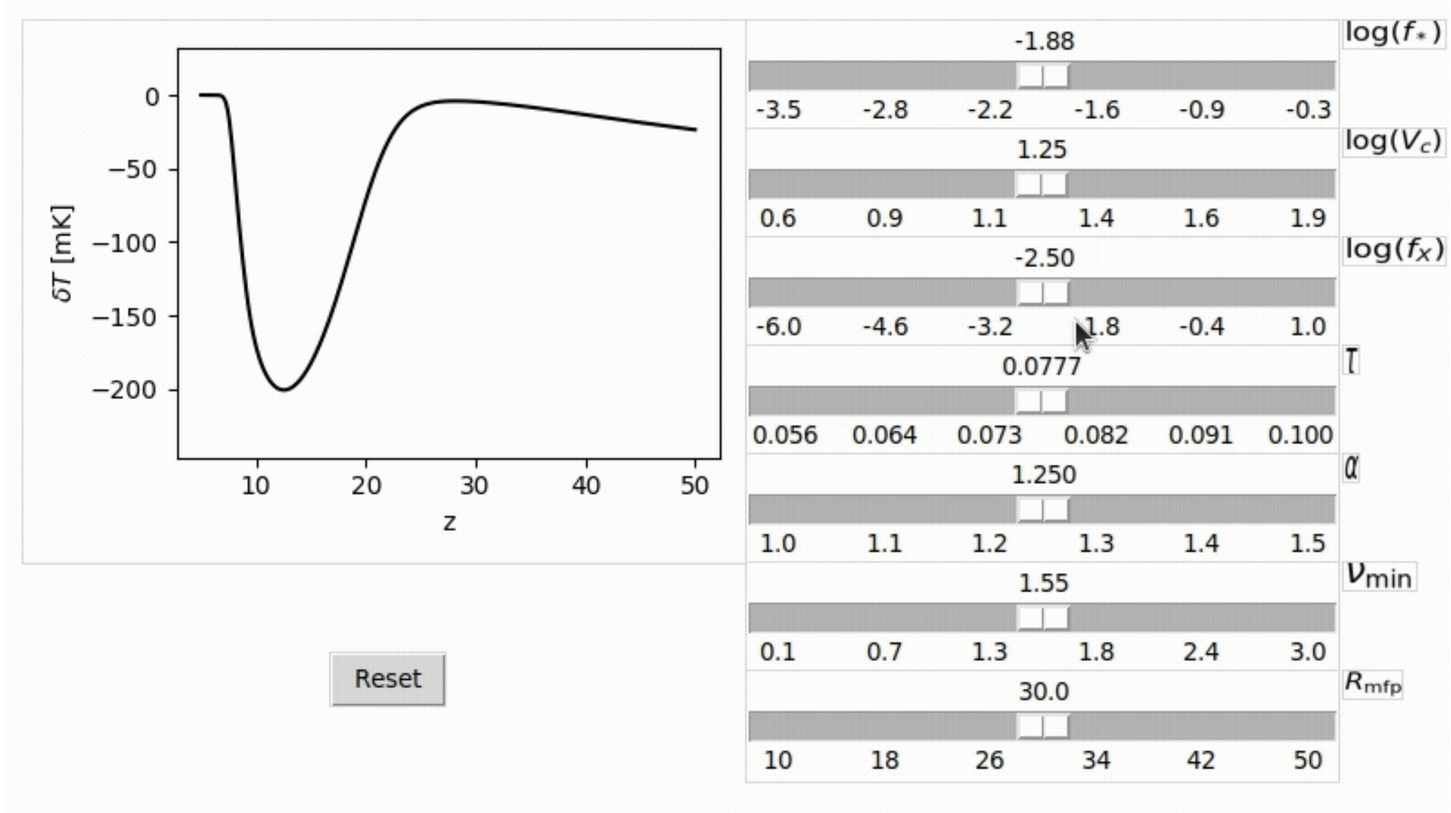


Kolmogorov-Arnold Network (KAN)





Emulators in 21cm Cosmology



Measuring accuracy…

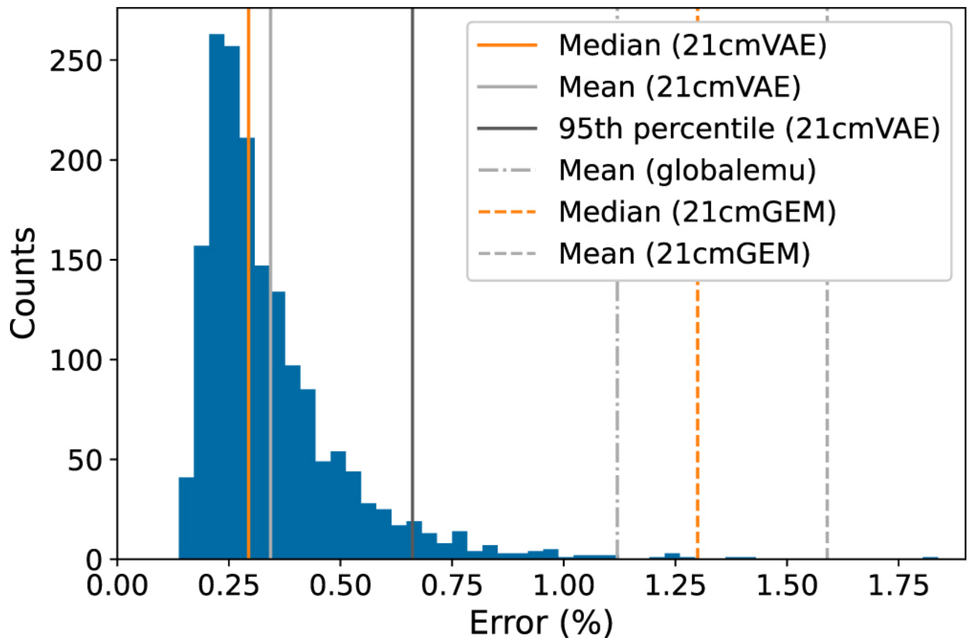
Defining required accuracy

- Typically we define accuracy with something like RMSE and a test data set

$$\epsilon = \sqrt{\frac{1}{N_\nu} \sum_i^{N_t} (T^{21}_{\rm true}(z) - T^{21}_{\rm pred}(t))^2}$$

- But what average value of ϵ over the test data is good enough for inference?

- Generally we work with “rules of thumb” e.g. globalemu paper suggested $\bar{\epsilon} \lesssim 0.1\sigma$

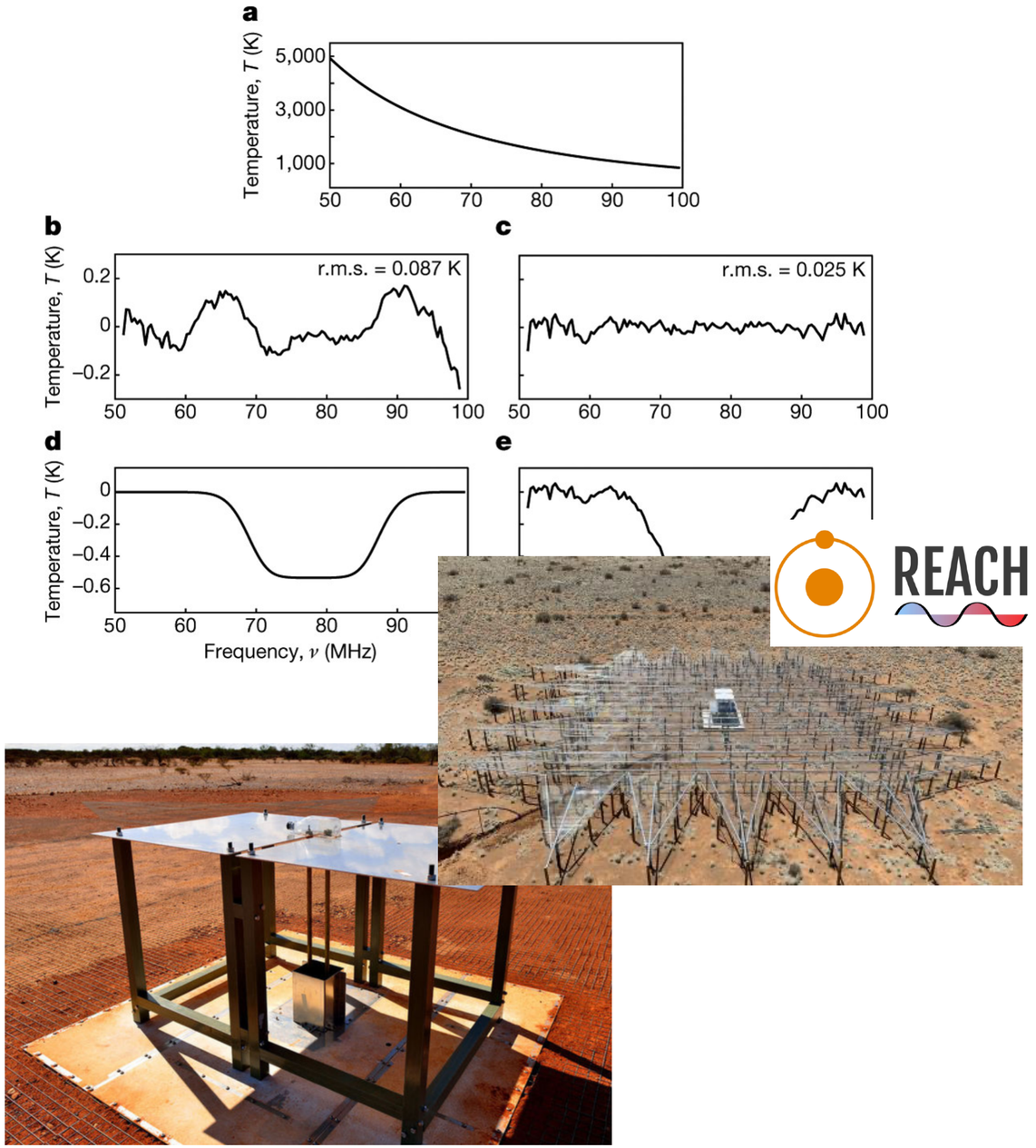


Why is accuracy important?

- Need 25 mK noise to confidently detect the sky-averaged 21cm signal
- Most emulators have
  $\bar{\epsilon} \approx 1\ \mathrm{mK} \approx 0.05 \times 25\mathrm{mK}$
- If we assume a Gaussian likelihood and

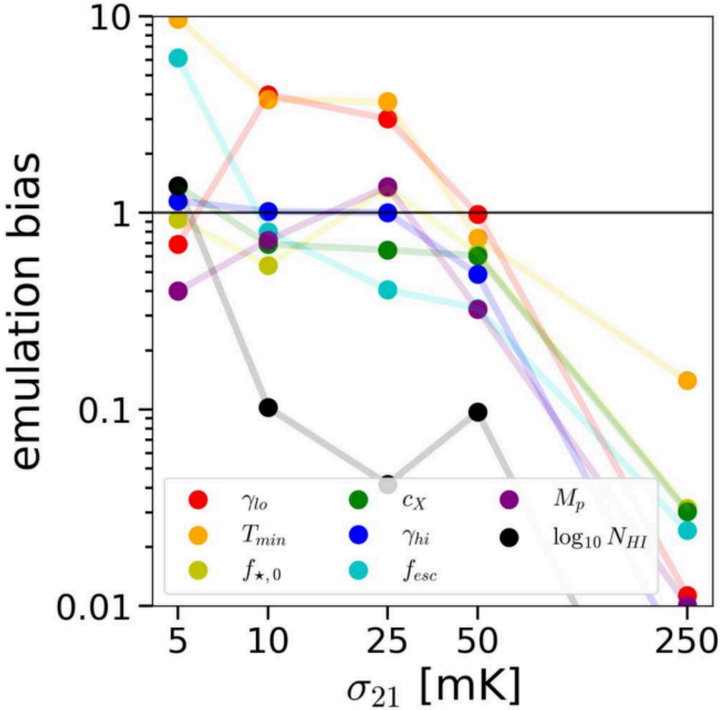
$$\sigma^2 = \sigma^2_{\mathrm{instrument}} + \bar{\epsilon}^2$$

we would expect the uncertainty from the instrument to dominate the posteriors but we really want a way to check this…



Defining required accuracy

- Dorigo Jones et al. 2023 started to ask and answer this question
- Making a direct comparison between $P(\theta \mid D, M) \leftrightarrow P(\theta \mid D, M_E)$
- We wanted to come up with something more predictive because we don't have $P(\theta \mid D, M)$ only $P(\theta \mid D, M_E)$
- "Given this error in our emulator and in our data how accurate do we expect our posteriors to be?"



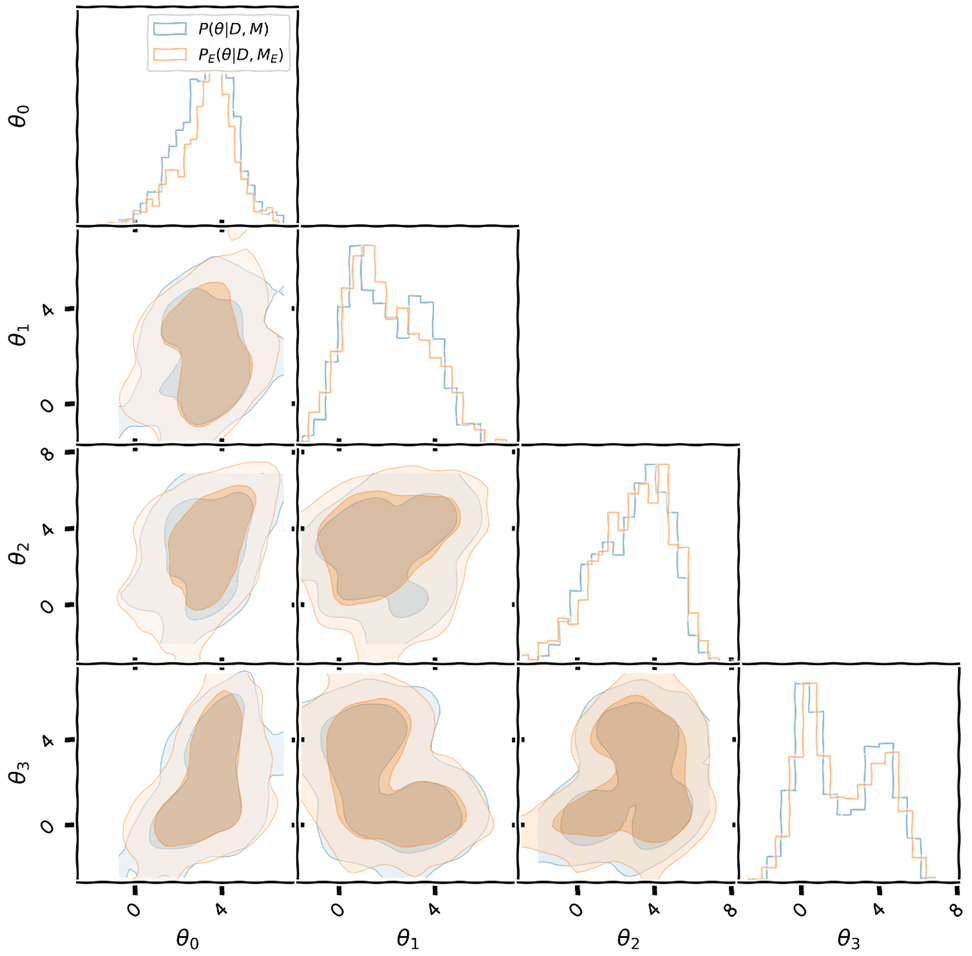
Impact on posterior recovery?

$$P\ Z = L\ \pi$$

- Likelihood function is probability of the data given the model $L = P(D\,|\,\theta, M)$

$$\log L(M) \rightarrow \log L(M_E) + \delta \log L(M_E)$$

$$P(\theta\,|\,D, M) = \frac{L\pi}{\int L\pi d\theta} \rightarrow P_E(\theta\,|\,D, M_E) = \frac{L\pi e^{\delta \log L}}{\int L\pi e^{\delta \log L} d\theta}$$



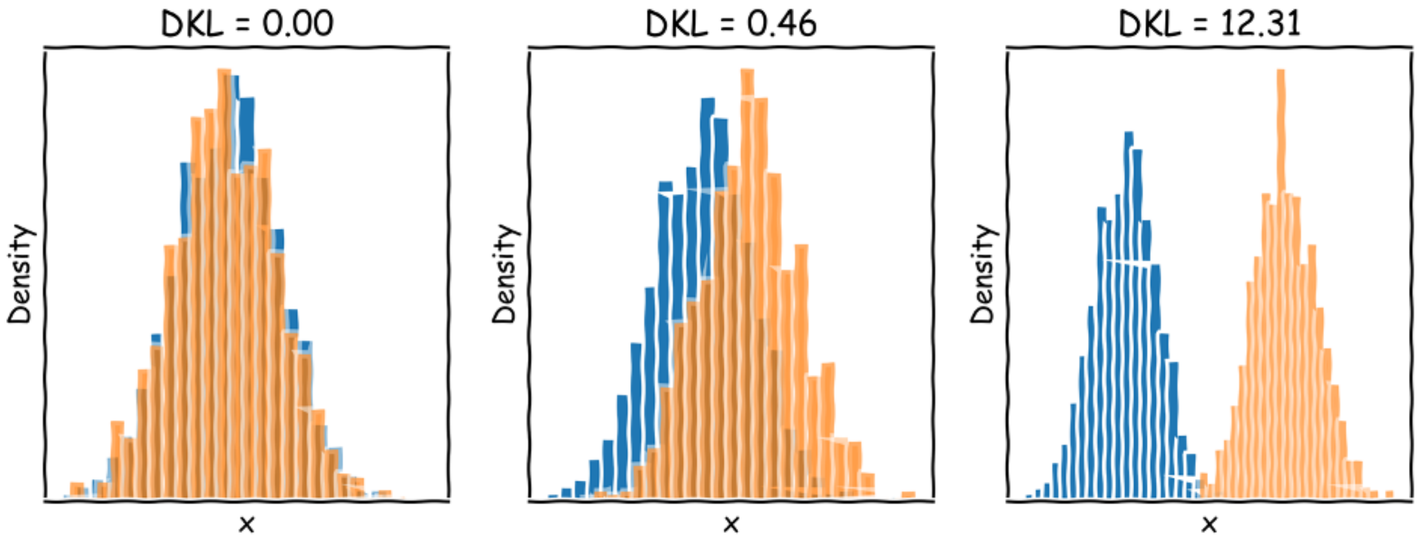
Measuring the impact of the emulator

- Comprehensive measure of the difference between the true and emulated posteriors is the Kullback-Leibler Divergence

$$D_{\mathrm{KL}} = \int P \log\left(\frac{P}{P_E}\right) d\theta$$

- Smaller this number the closer the distributions.

- However we don't have access to P...



Measuring the impact of the emulator

Thomas Gessey-Jones

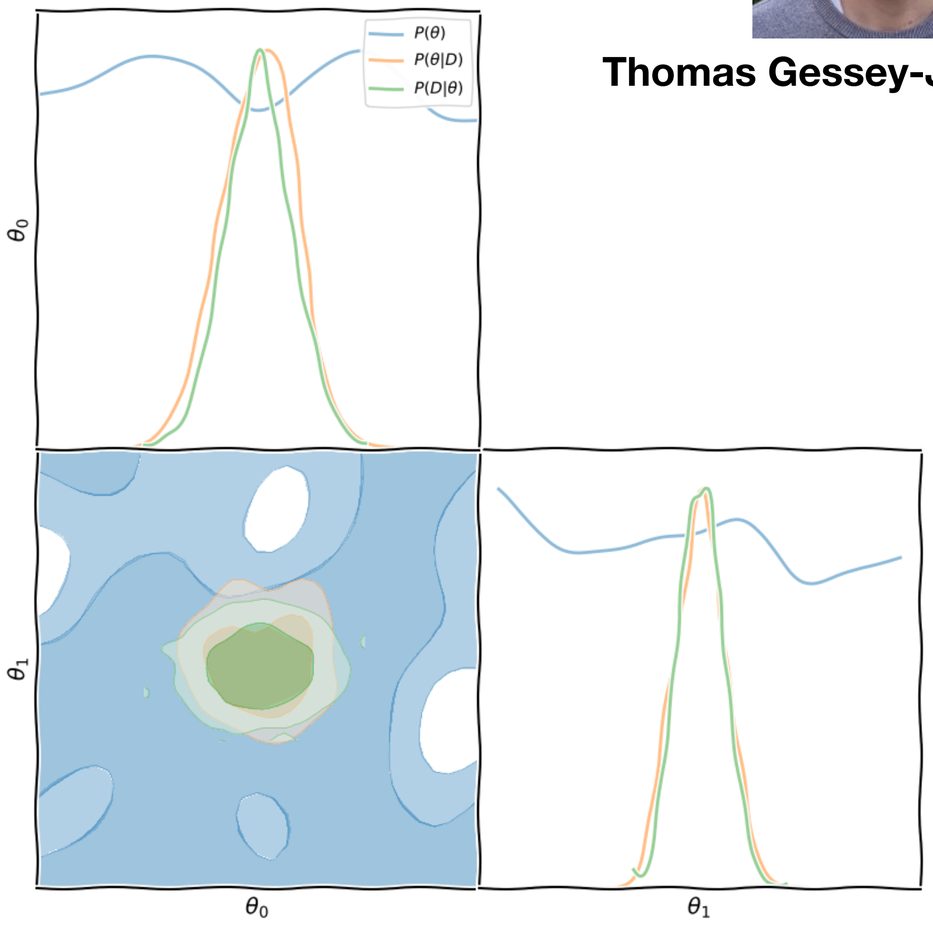
- If we make some approximations we can however define an upper limit on $D_{KL}(P\,||\,P_E)$

$$L = \mathcal{N}(D; \Sigma, M(\theta))$$
$$\pi = \mathcal{U}(\theta)$$
$$\rightarrow P = \mathcal{N}(\theta; C, \mu)$$

- P and P_E are Gaussian then the KL divergence between them is given by

$$D_{KL} = \frac{1}{2}\left[\log\left(\frac{|C_E|}{|C|}\right) - N_\theta + \mathrm{tr}(C_E^{-1}C) + (\mu_E - \mu)^T C^{-1}(\mu_E - \mu)\right]$$

- Where C, C_E, μ and μ_E are functions of M, M_E and Σ the noise in the data



Measuring the impact of the emulator

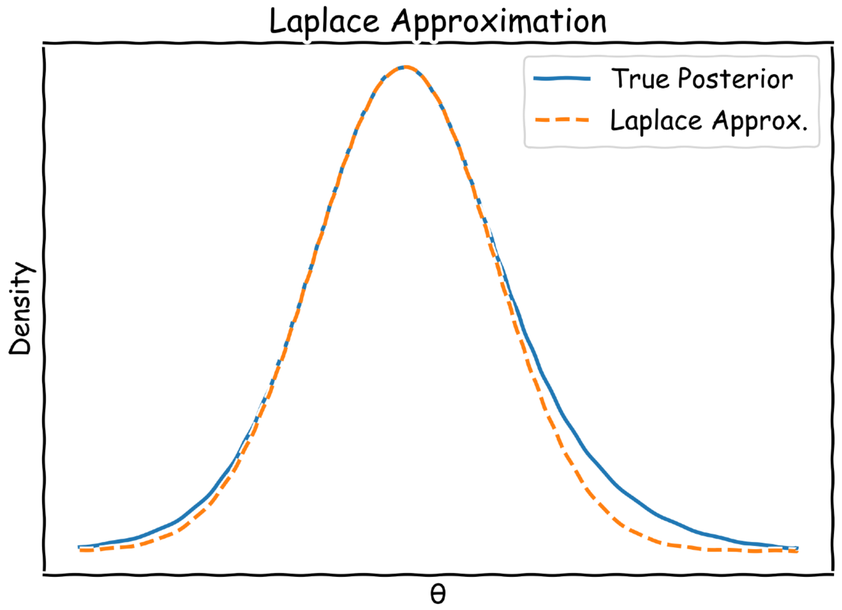
- Assume a linear model and linear emulator error

$$\mathscr{M}(\theta) \approx M\theta + m \text{ and } E(\theta) \approx E\theta + \epsilon$$

Such that $M_\epsilon(\theta) = (M + E)\theta + (m + \epsilon)$

- Comes from Taylor expansion of model around the MAP and the assumption that the posterior is sharply peaked so we can ignore higher order terms

$$M = \mathscr{J}(\theta_0)$$
$$m = M(\theta_0) - \mathscr{J}(\theta_0)\theta_0$$



Measuring the impact of the emulator

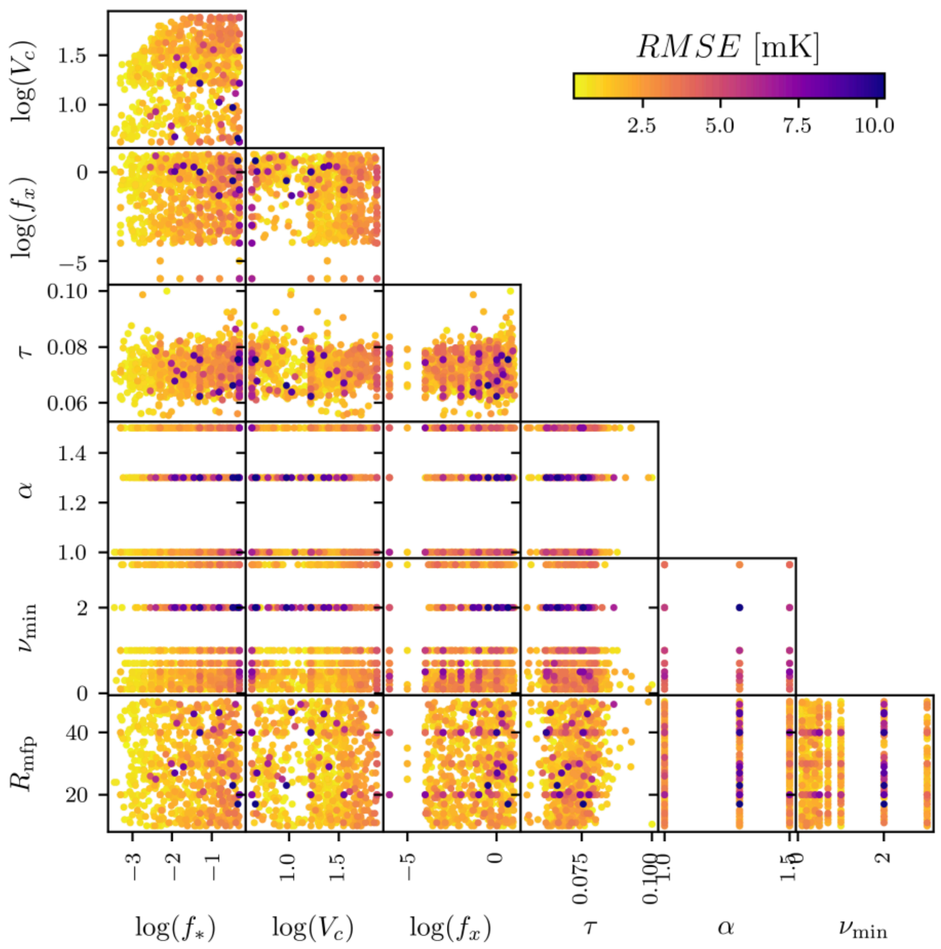
$$\mathscr{M}(\theta) \approx M\theta + m \text{ and } E(\theta) \approx E\theta + \epsilon$$

Such that $M_\epsilon(\theta) = (M + E)\theta + (m + \epsilon)$

- Then assume that the emulator error evolves slowly over the parameters space relative to the model then $E(\theta) \approx \epsilon$
- Substitute into the posteriors

$$P(\theta \,|\, D, M) \propto \exp\left(-\frac{1}{2}(D - m - M\theta)^T \Sigma^{-1} (D - m - M\theta) \right)$$

Then we can express C, C_E, μ and μ_E in terms of M, m and ϵ.

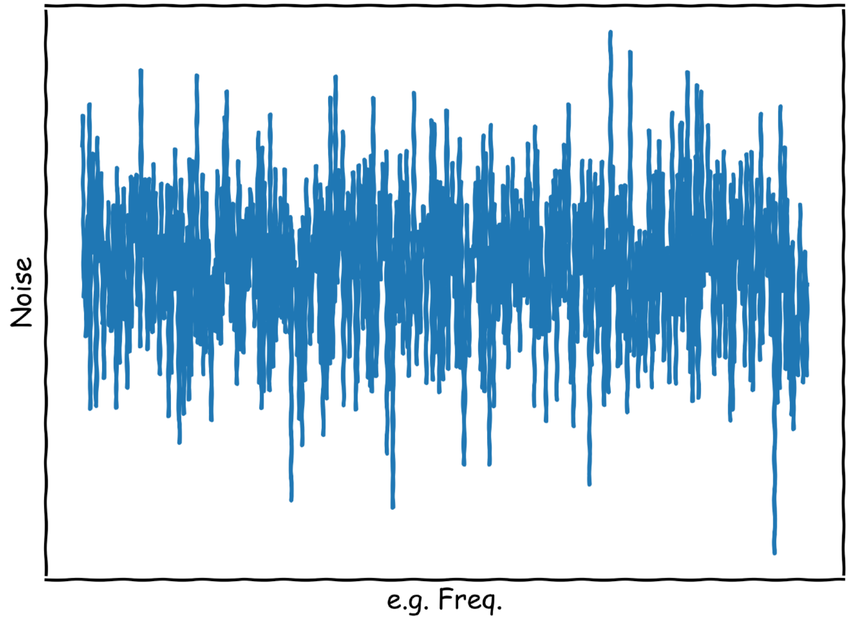


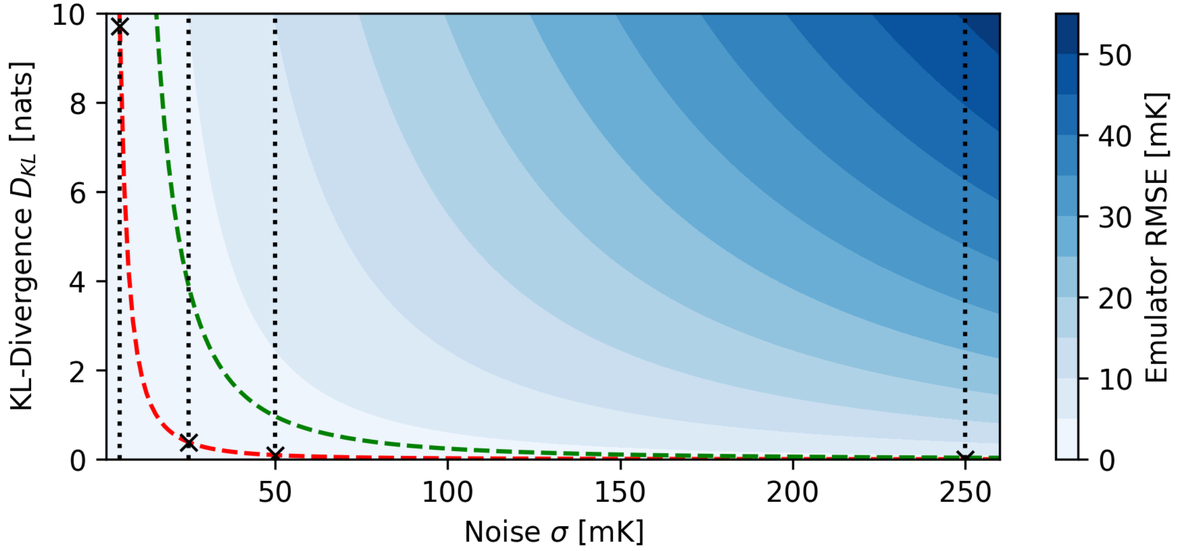
Measuring the impact of the emulator

We then assume the noise $\Sigma = \frac{1}{\sigma^2}\mathbf{1}_{N_d}$ then you can show...

$$D_{\mathrm{KL}}(P\,||\,P_E) \leq \frac{1}{2}\frac{1}{\sigma^2}||\epsilon||^2$$

$$D_{\mathrm{KL}}(P\,||\,P_E) \leq \frac{N_d}{2}\left(\frac{\mathrm{RMSE}}{\sigma}\right)^2$$



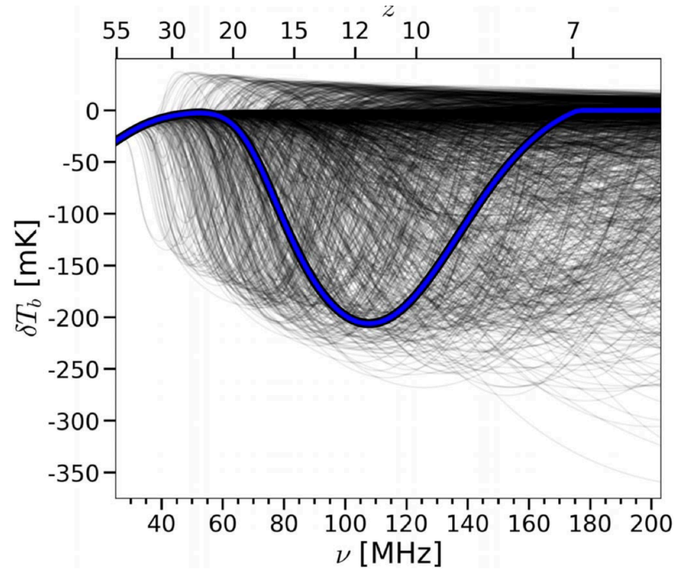


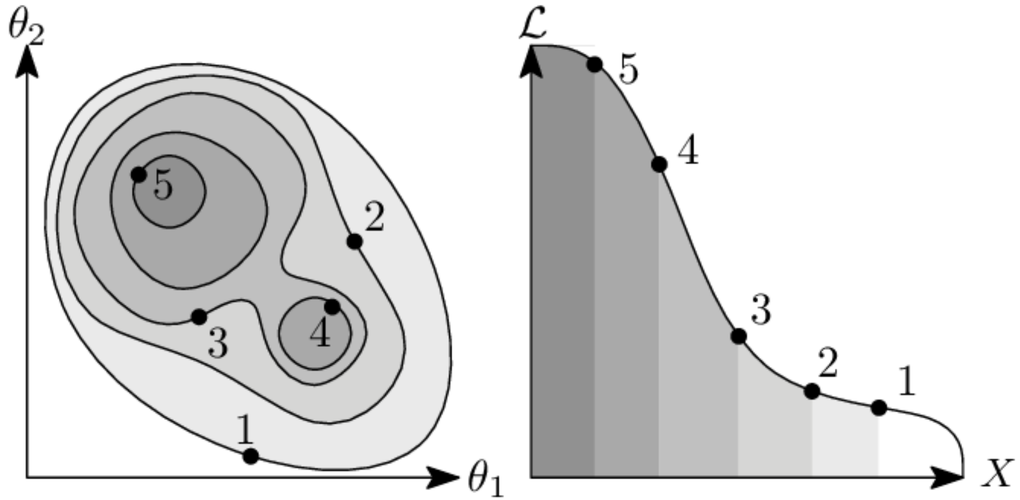
Measuring the impact of the emulator

$$\epsilon = \sqrt{\frac{1}{N_\nu} \sum_i^{N_t} (S_{\text{true}}(t) - S_{\text{pred}}(t))^2} \longrightarrow D_{\text{KL}}(P \,||\, P_E) \leq \frac{N_d}{2} \left(\frac{\text{RMSE}}{\sigma} \right)^2$$

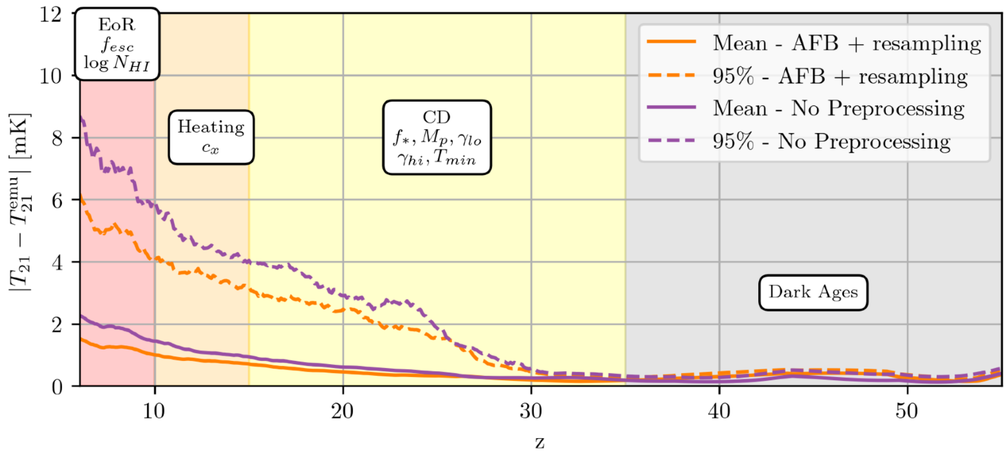
Testing on a 21cm Cosmology problem

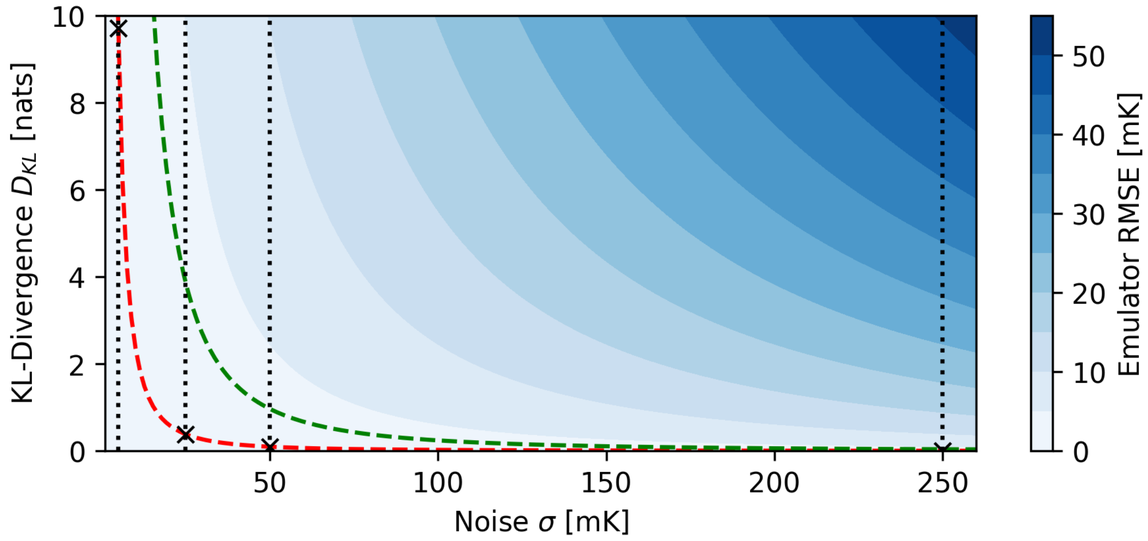
- Assuming the data comprises of signal plus noise
- Using the ARES 1D radiative transfer code with 8 parameters
- Using Polychord to perform inference with a gaussian likelihood
- Assuming absolute knowledge of the level of noise in the data
- Running for 5, 25, 50 and 250 mK noise





globalemu performance and ARES modelling



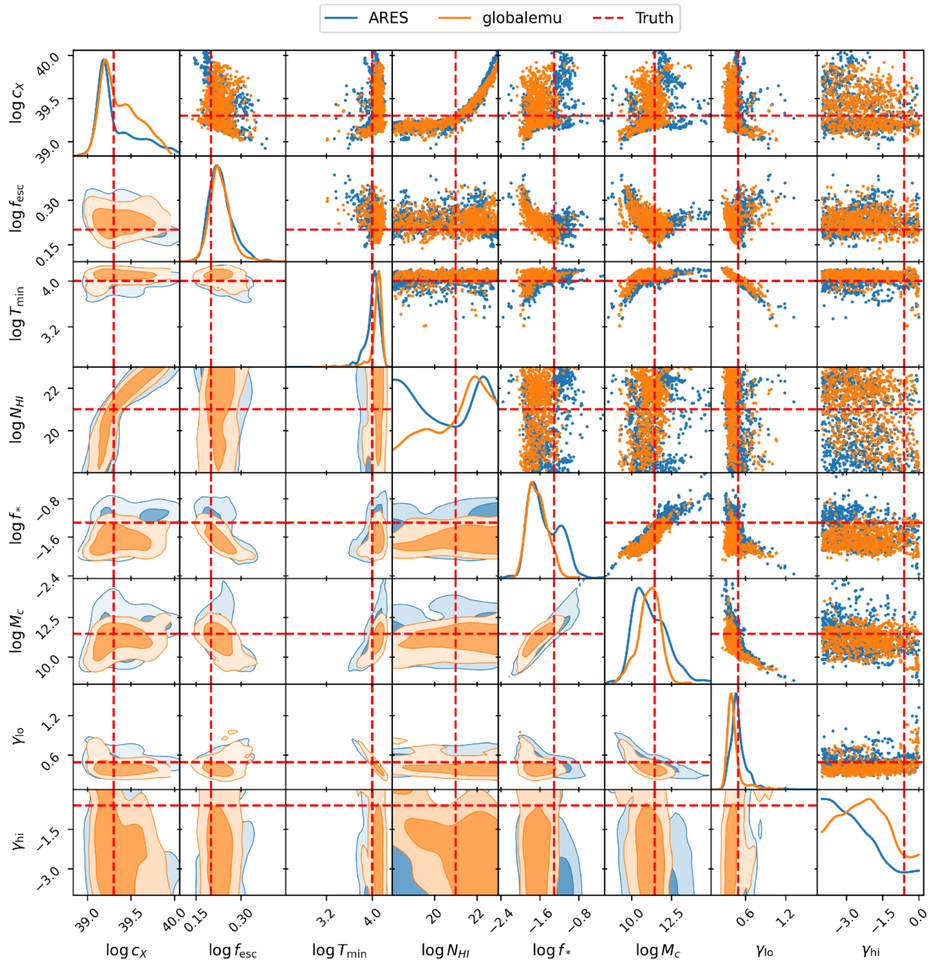


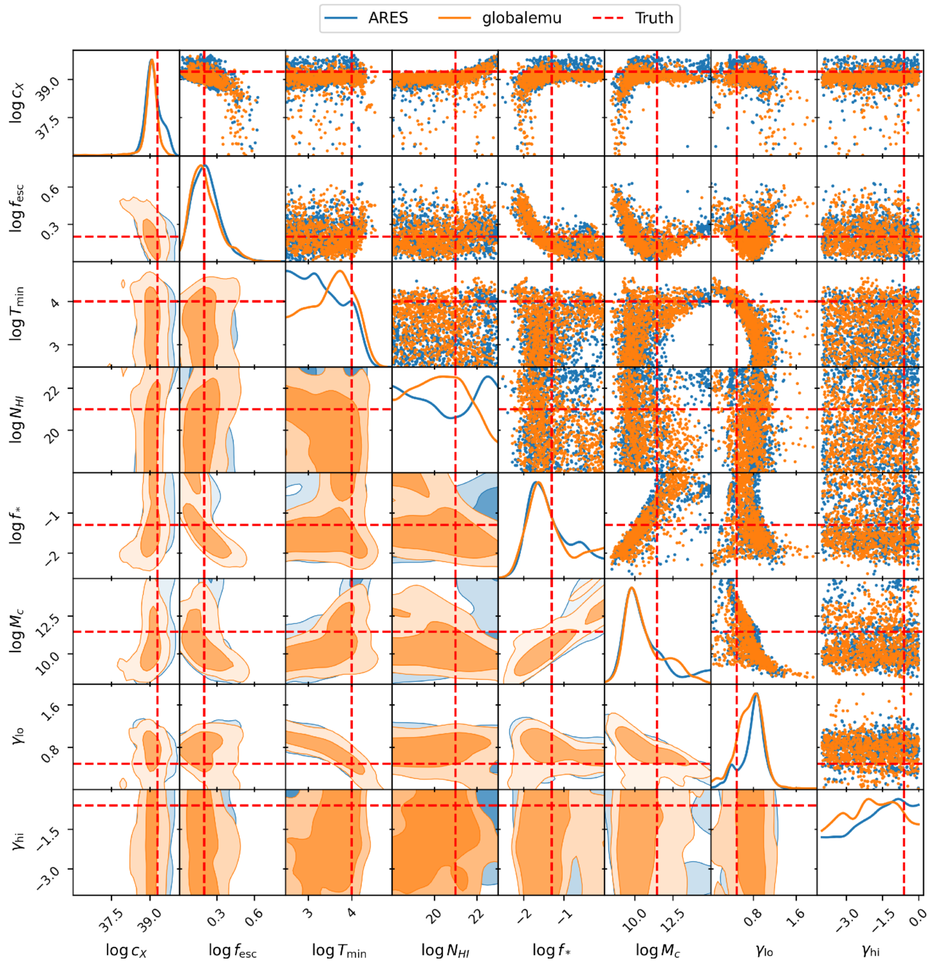
Metric	With AFB, with resampling	Resampling only	AFB only	No AFB, no resampling	Emulator Used (With AFB, with resampling)	DJ23 No AFB, no resampling
Mean	1.23 ± 0.16	1.61 ± 0.16	1.41 ± 0.12	1.36 ± 0.08	0.99	1.25
95th Percentile	3.96 ± 0.72	3.98 ± 0.57	4.59 ± 0.53	4.48 ± 0.33	3.14	—
Worst	31.04 ± 7.09	31.58 ± 9.48	33.91 ± 6.13	33.89 ± 4.70	25.97	18.5

Noise Level [mK]	Estimated $\mathcal{D}_{\mathrm{KL}} \leq$	
	Mean RMSE	95th Percentile
5	9.60	96.62
25	0.38	3.86
50	0.10	0.97
250	0.004	0.039

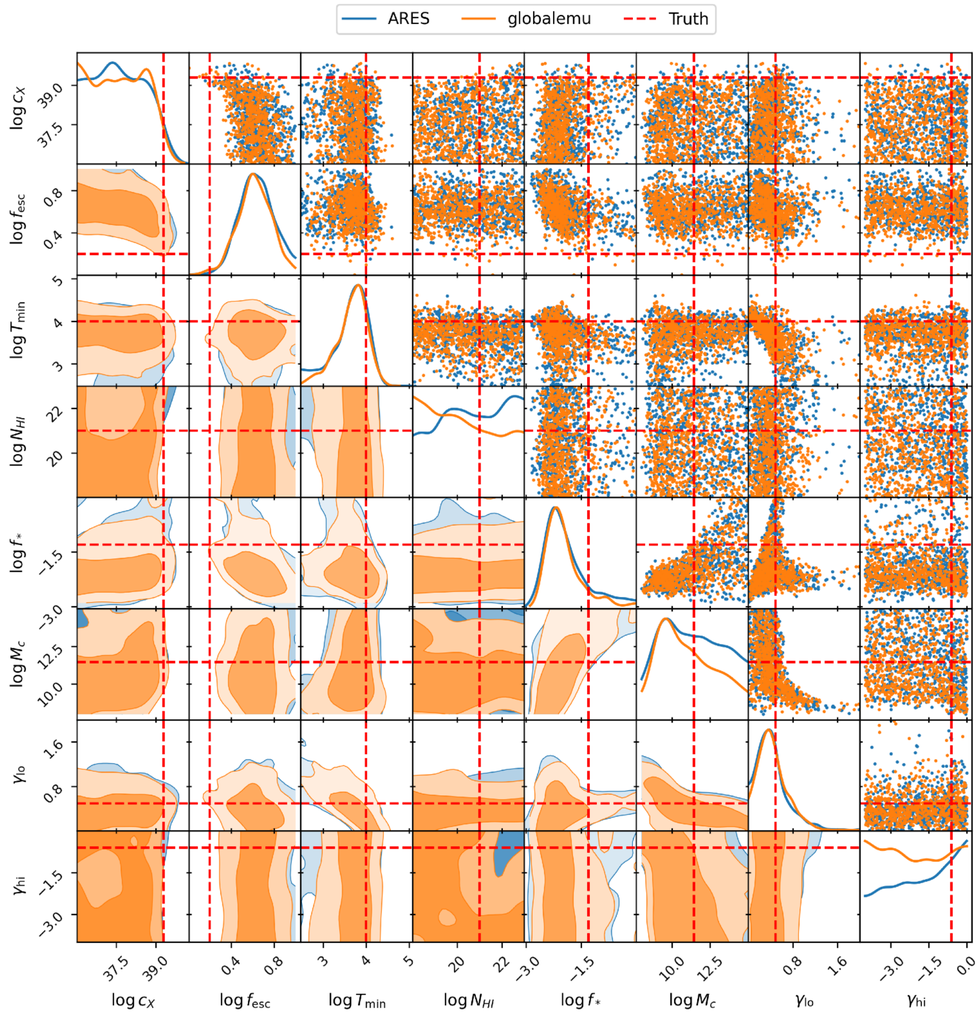
Running the analysis - 5 mK and 25 mK

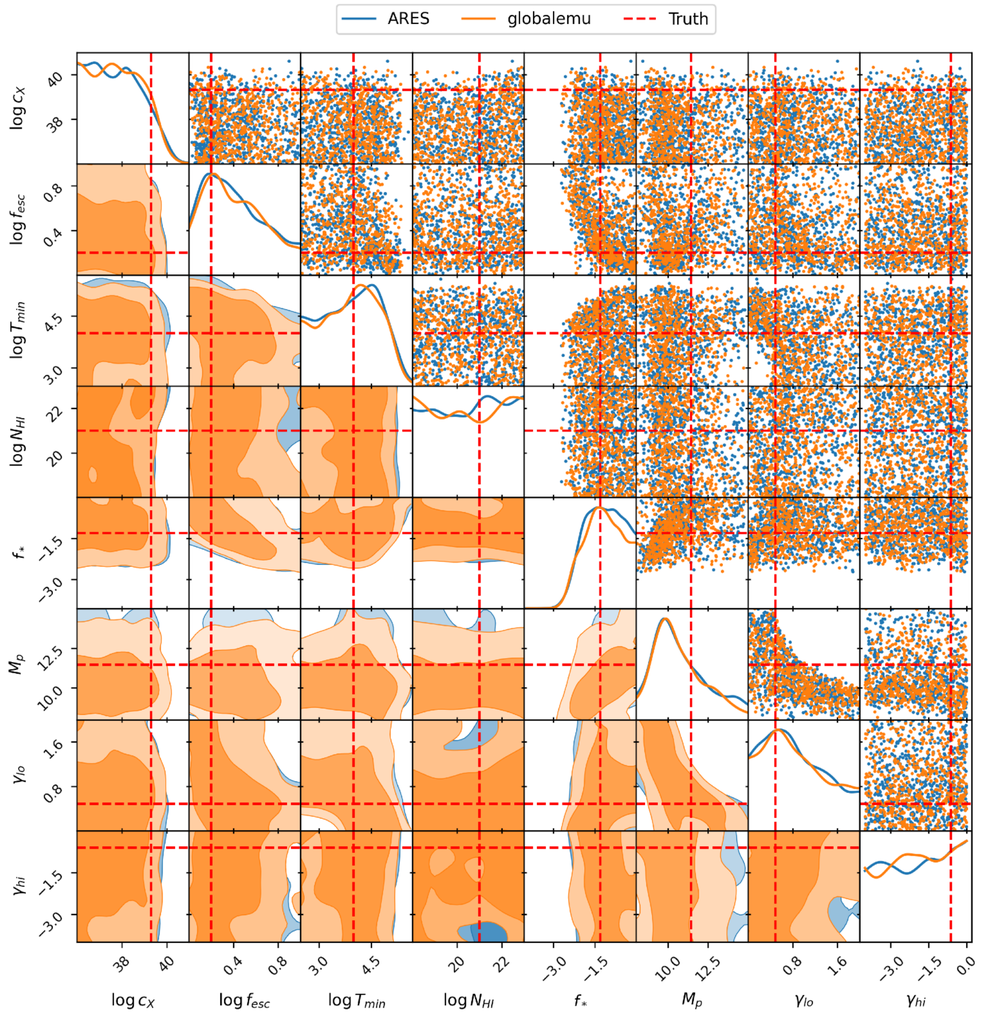
- Mean emulator error $= 1.23 \pm 0.16$ mK





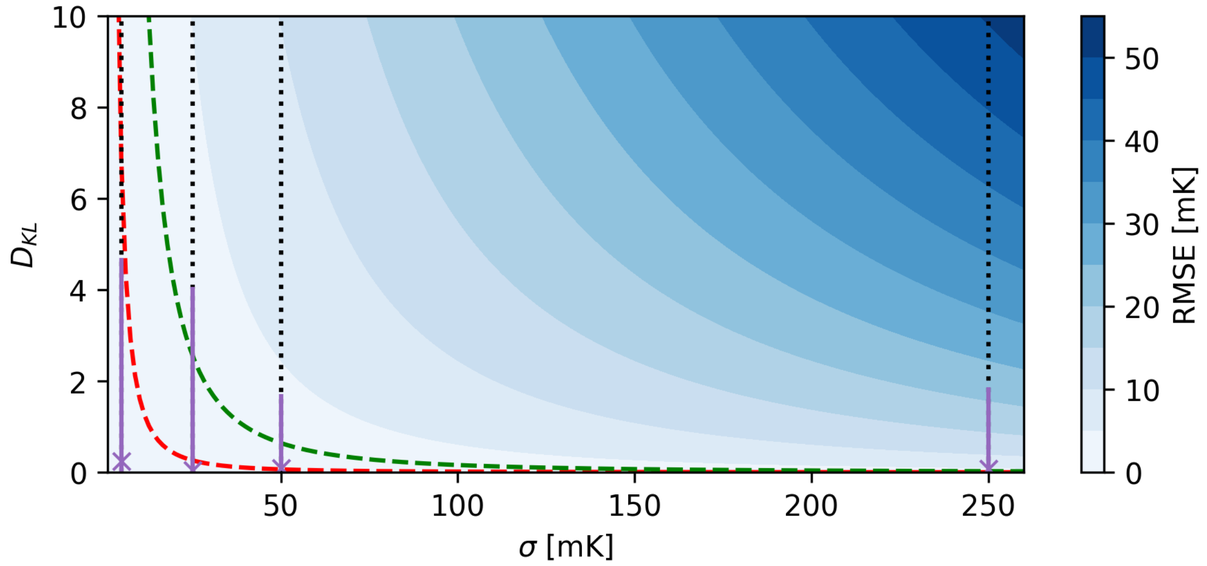
Running the analysis - 50 and 250 mK





arXiv:2503.13263

How about the D_{KL}?



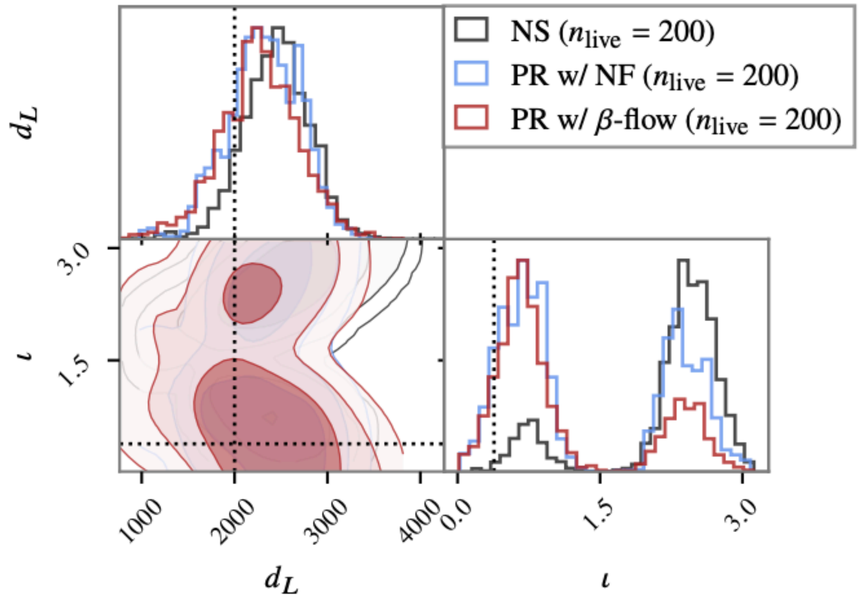
Noise Level [mK]	Estimated $\mathcal{D}_{\mathrm{KL}} \leq$		Actual $\mathcal{D}_{\mathrm{KL}}$
	Mean RMSE	95th Percentile	
5	9.60	96.62	$0.25^{+4.45}_{-0.25}$
25	0.38	3.86	$0.05^{+4.02}_{-0.52}$
50	0.10	0.97	$0.09^{+1.62}_{-0.03}$
250	0.004	0.039	$0.08^{+1.78}_{-0.02}$

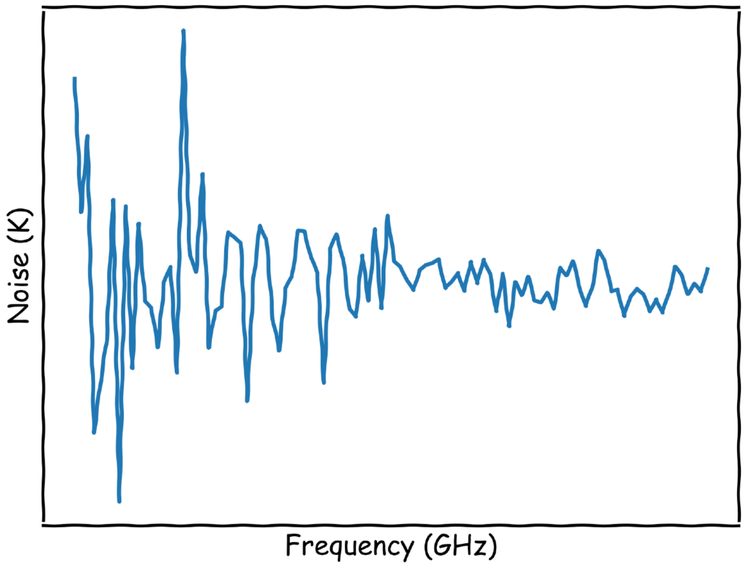
- Use normalising flows implemented with ***margarine*** to estimate the true KL [see Bevins et al 2022, 2023, arXiv:2207.11457, arXiv:2205.12841]

Limitations of the approximation

- The approximation assumes linearity around the peak of the posterior which might not hold in higher dimensions
- Posteriors become curved or multi modal
- Assuming a Gaussian likelihood and posterior
- Assumes uncorrelated noise in the data
- Assumes noise is constant across the data

Prathaban, Bevins, Handley. 2024

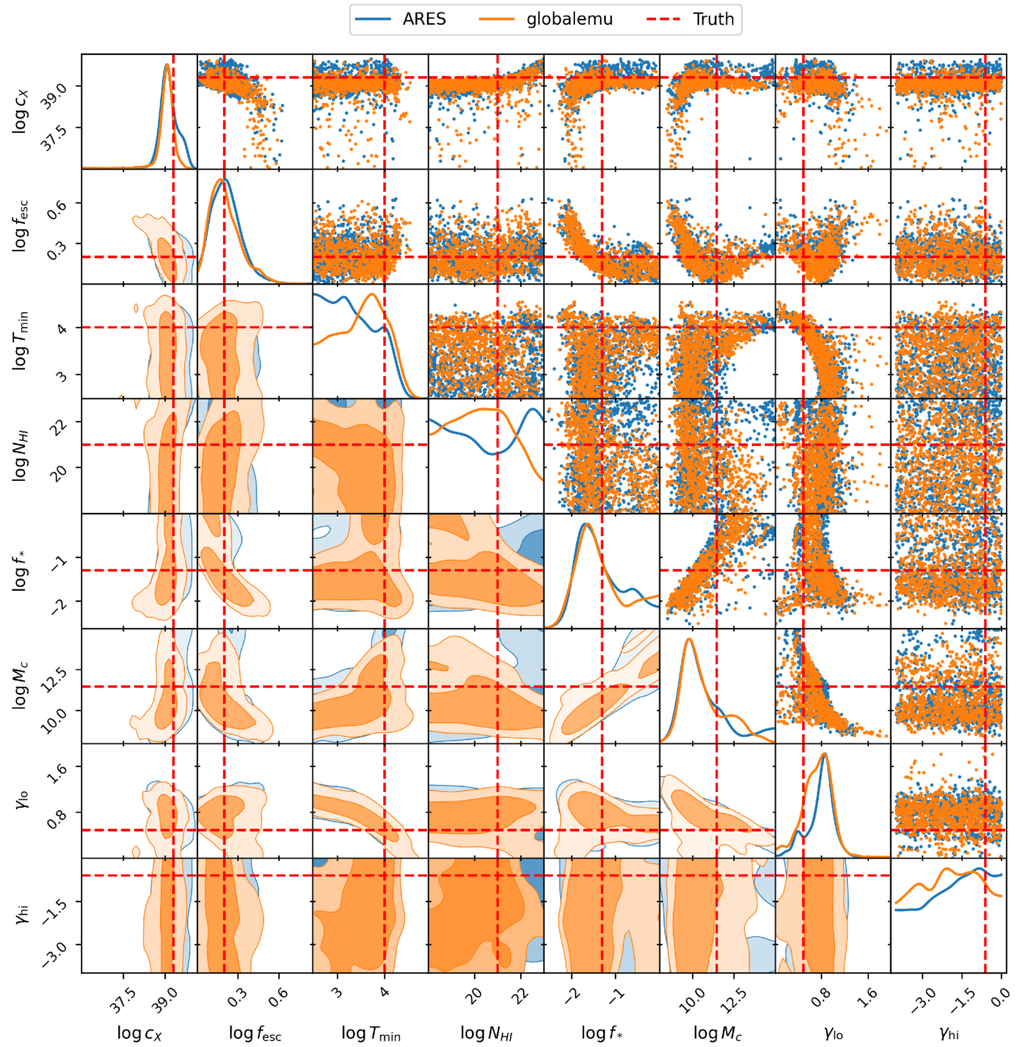




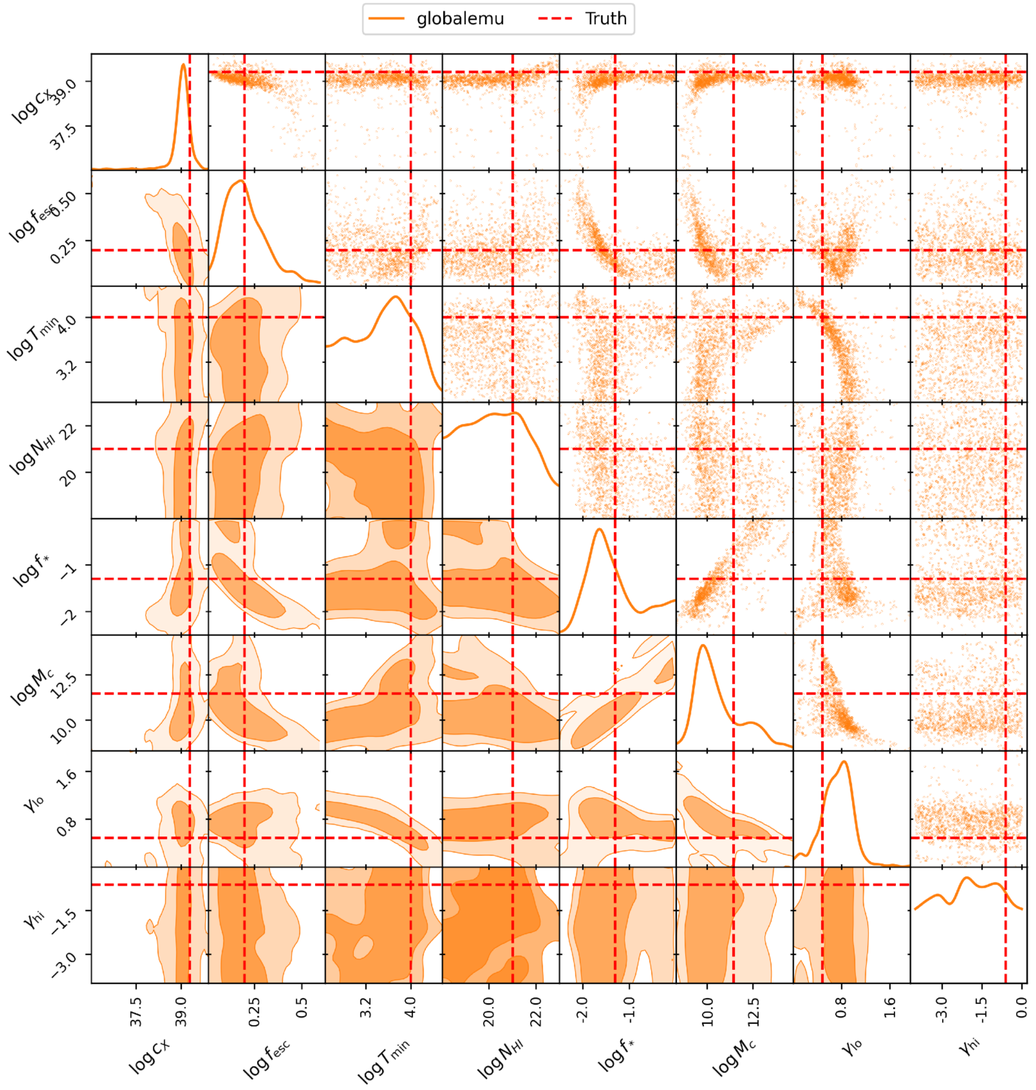
arXiv:2503.13263

How to use this?

Build Confidence in inference



Build Confidence in inference



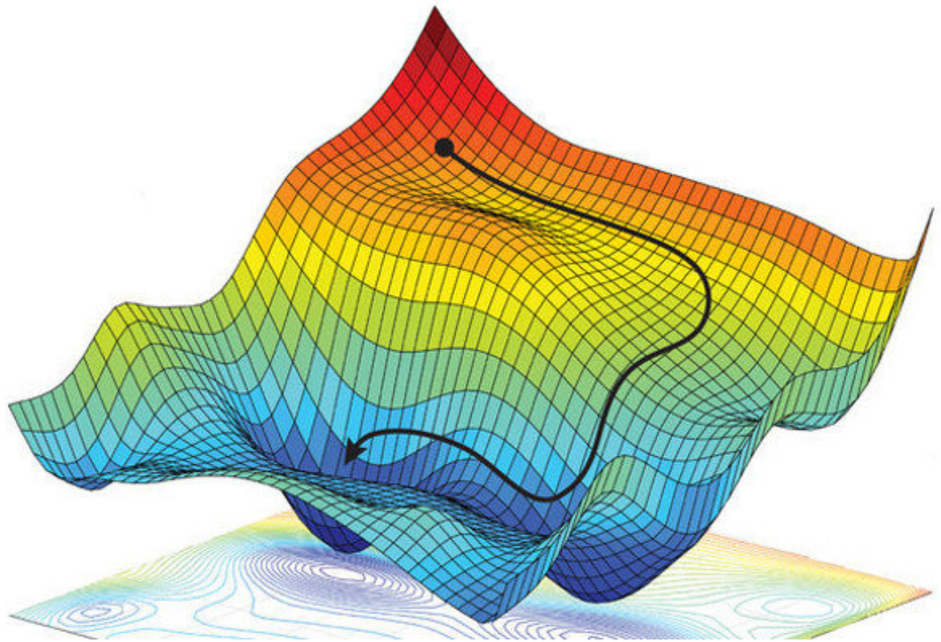
As a Loss Function and for hyperparameter tuning?

- If we know how noisy our data is we can use

$$D_{KL} \leq \frac{N_d}{2}\left(\frac{\mathrm{RMSE}}{\sigma}\right)^2$$

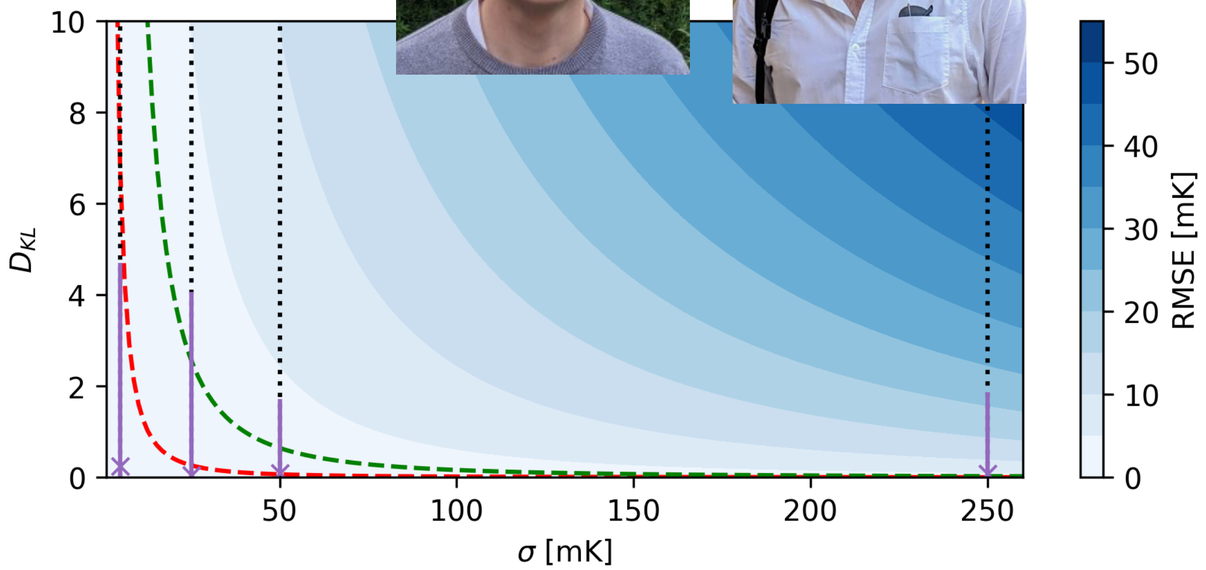
as a loss function in our emulator training

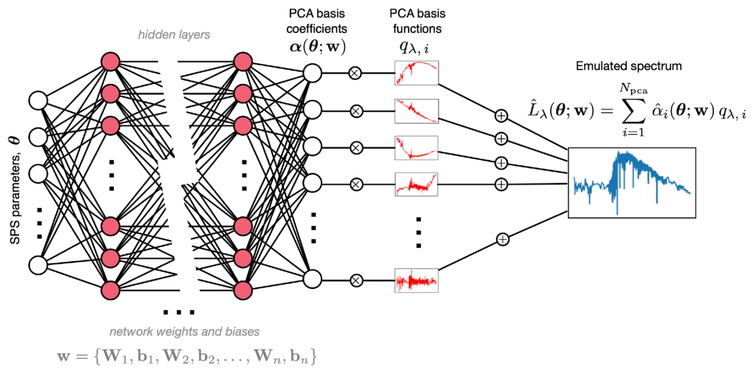
- Similarly we can use it as our objective in hyperparameter tuning with frameworks like optuna

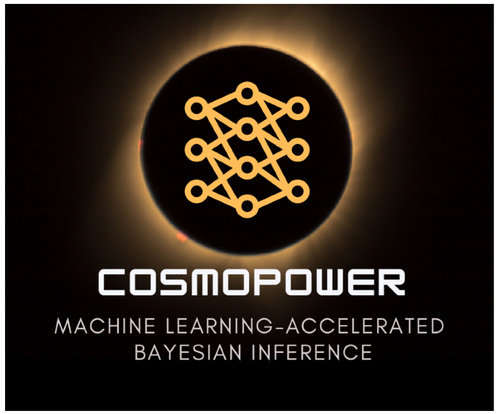


Conclusions

- A useful upper bound on the incurred information loss from using emulators in inference
- Broadly applicable beyond 21cm
- Can use this as a loss function or for hyperparameter tuning
- Accepted in MNRAS [arXiv:2503.13263]
- https://github.com/htjb/validating_posteriors







Additional Slides

Measuring the D_{KL} with normalising flows

- Need to be able to evaluate the probability on each distribution for the same samples
- Use normalising flows implemented with ***margarine*** [see Bevins et al 2022, 2023, arXiv:2207.11457, arXiv:2205.12841]