globalemu: Novel and robust emulation of 21-cm signals from the Epoch of Reionization

Harry T. J. Bevins

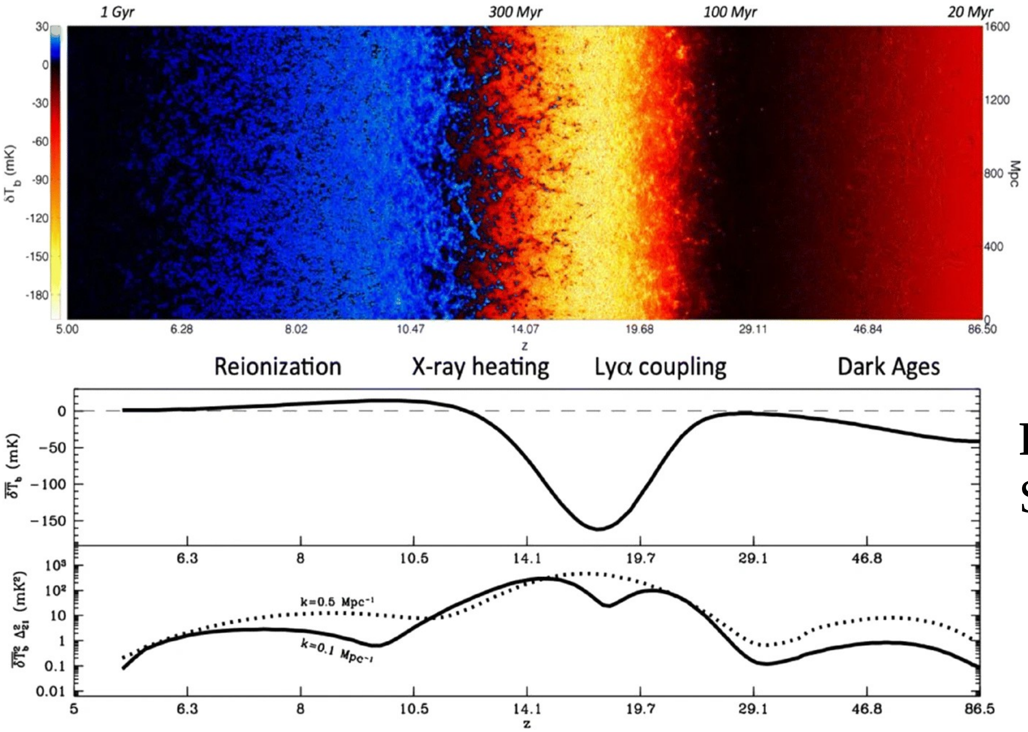
Cavendish Asrophysics, University of Cambridge

In collaboration with Will Handley, Anastasia Fialkov, Eloy de Lera Acedo and Kamran Javid

EUROPEAN ASTRONOMICAL SOCIETY ANNUAL MEETING

eas 2022 Valencia

21-cm Cosmology and the SKA



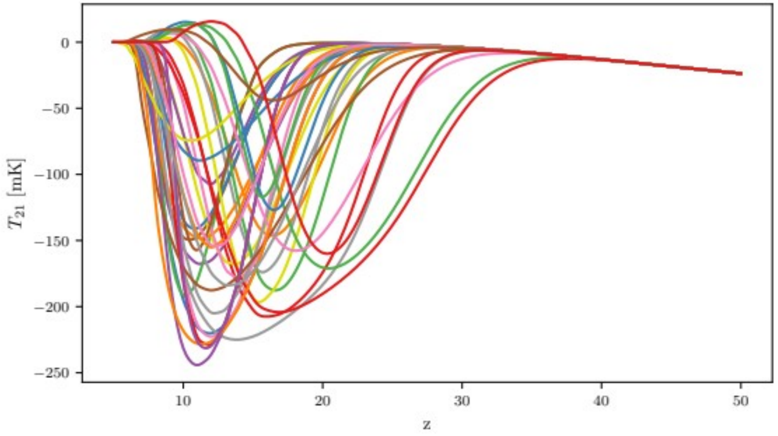
REACH, EDGES, SARAS, LEDA etc.

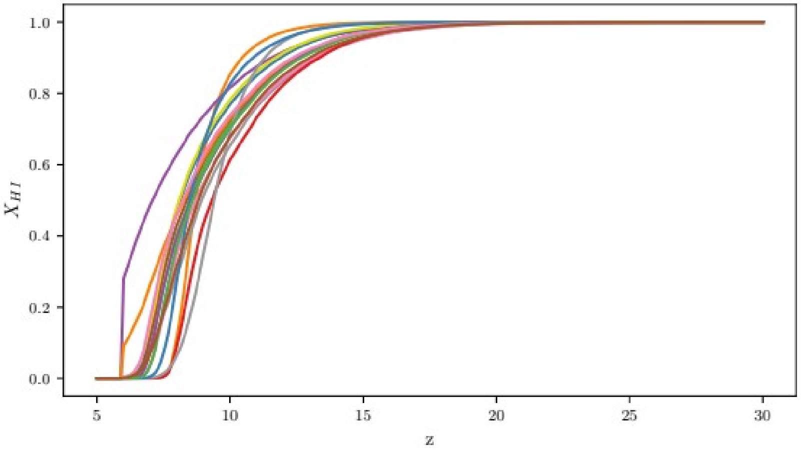
SKA, HERA, MWA, LOFAR etc.

Koopmans, L.V.E., Barkana, R., Bentum, M. et al. Exp Astron 51, 1641–1676 (2021).

Why are we interested in signal emulators?

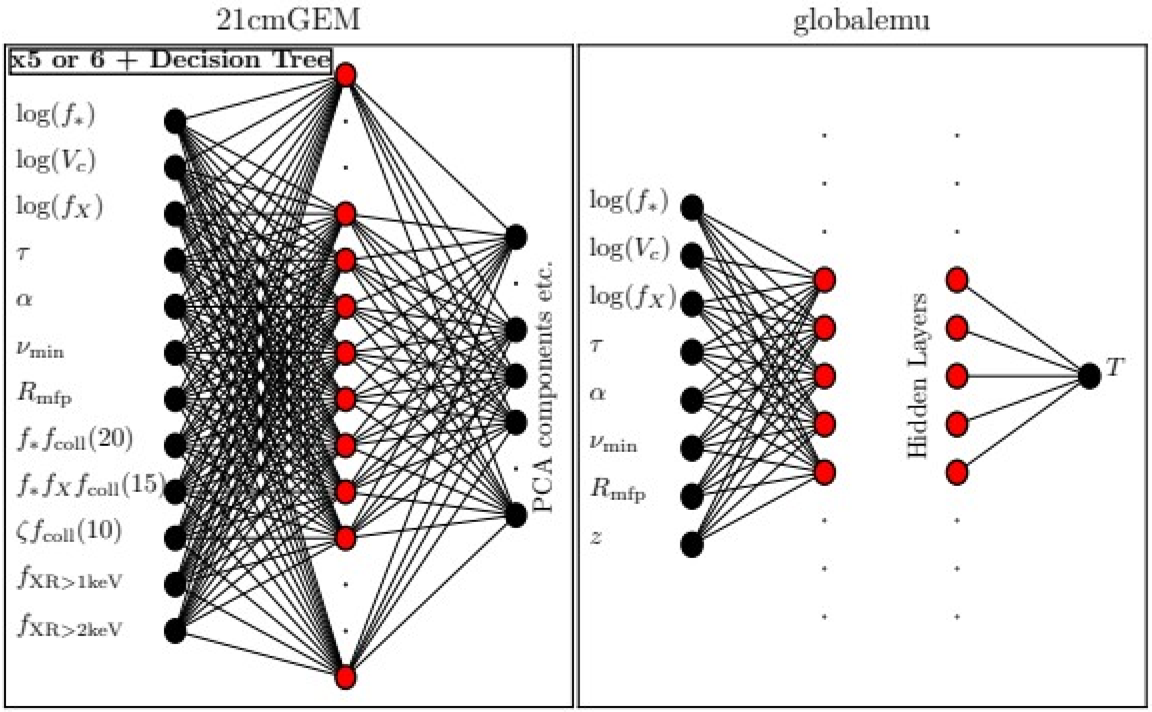
- Need a quick method to generate physical signal models to fit to our data
- Full semi numerical simulations take of order ~hrs
- Neural network emulators produce signals ~ms





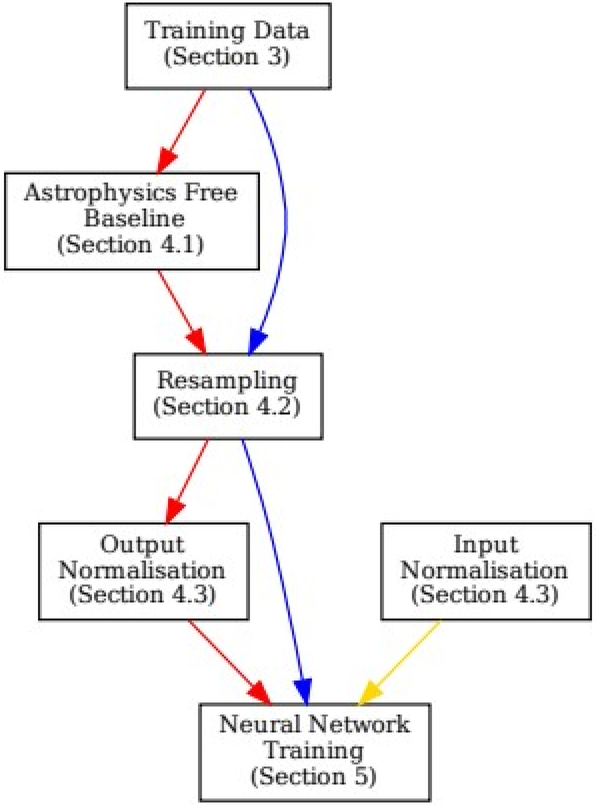
What is different about *globalemu*?

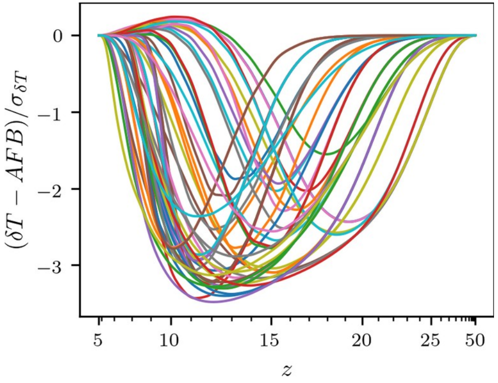
- Redshift as an input
- One simple and small neural network that we can make vectorised calls to
- Train and test on the same dataset for comparison

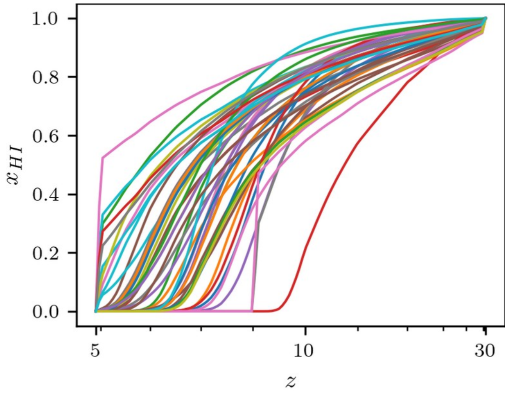


Physically Motivated Preprocessing

- Lots of details in the paper
- Subtract astrophysics free structure at high redshift and add it back in later
- Resample signals to capture regions of high variation



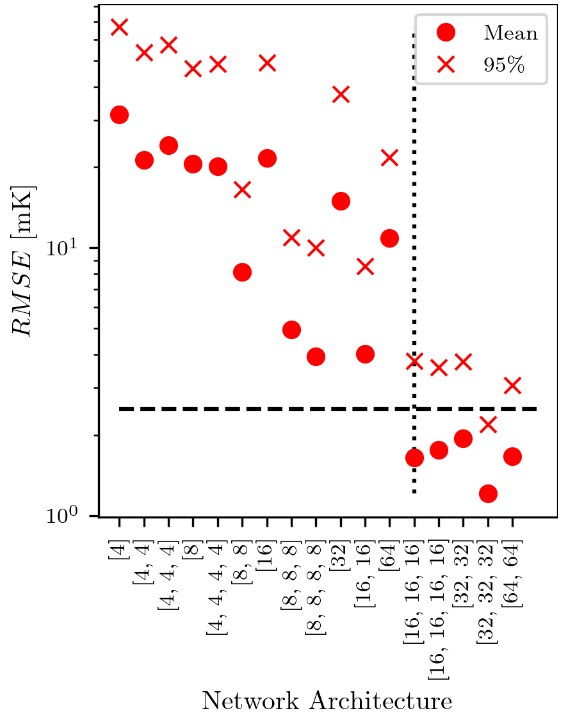




Neural Network Specifics

- Practical decisions about our required accuracy based on the expected noise in the REACH experiment

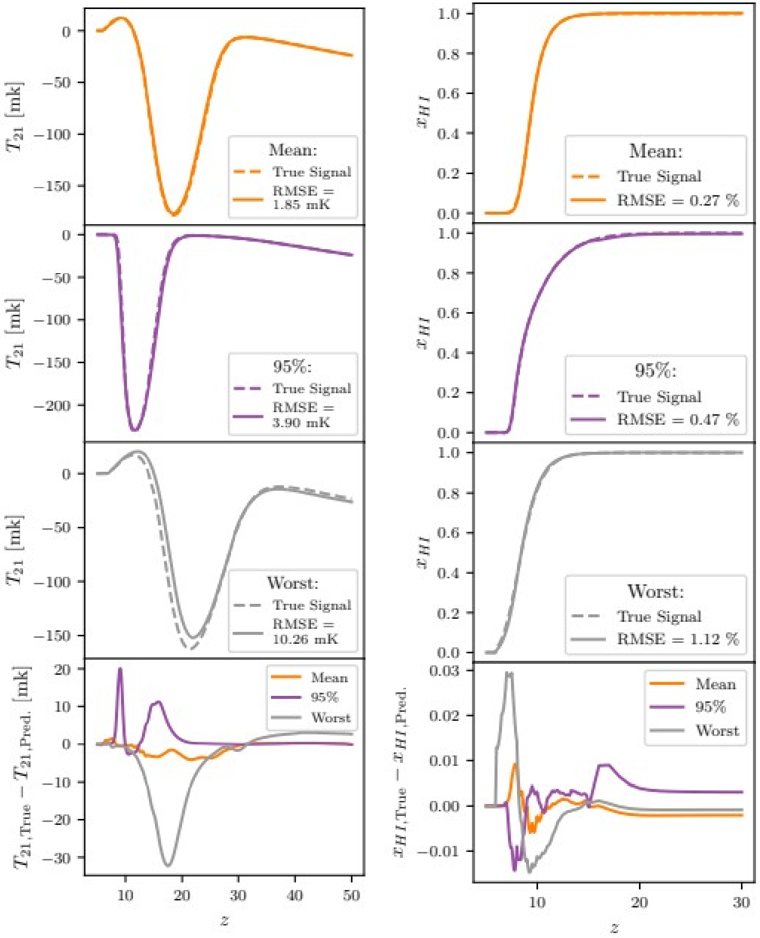
- Loss function and optimizer



Results

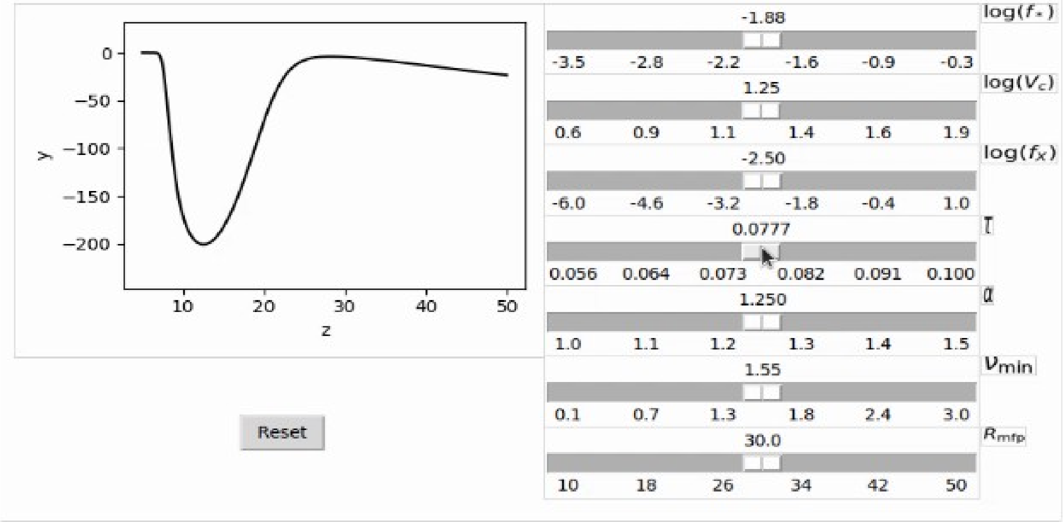
For the global signal:

- ***globalemu:***
  - Max Normalised RMSE = 6.32 %
  - Emulation time: 1.3 ms
- ***21cmGEM:***
  - Max Normalised RMSE = 10.55 %
  - Emulation time: 133 ms



Accessible and Adaptable

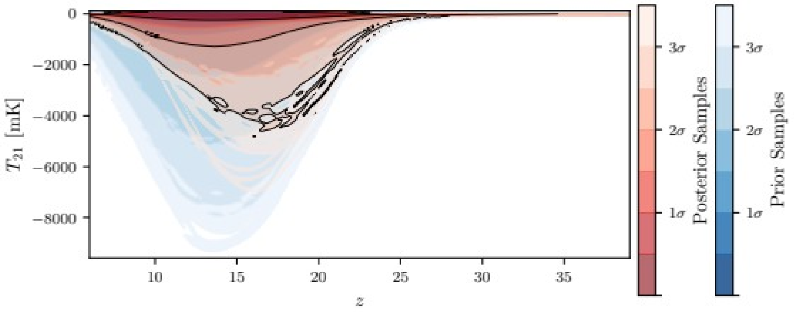
- Pip installable and fully documented
- Available on github with some trained models
- Emulator GUI



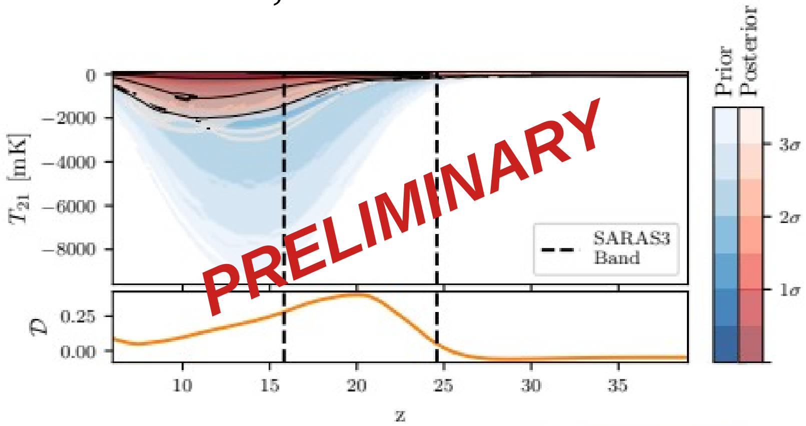

```python
import numpy as np
from globalemu.preprocess import process
from globalemu.network import nn
import tensorflow as tf
import matplotlib.pyplot as plt

tf.random.set_seed(1420.4)

data_dir = 'data/'
base_dir = 'saved_model/'
z = np.arange(5, 50.1, 0.1)

process('full', z, base_dir=base_dir, data_location=data_dir)

nn(batch_size=len(z), epochs=500, base_dir=base_dir, layer_sizes=[16]*4)

from globalemu.eval import evaluate

predictor = evaluate(base_dir=base_dir)

parameters = np.loadtxt(data_dir + 'test_data.txt')
labels = np.loadtxt(data_dir + 'test_labels.txt')

signals, z = predictor(parameters)

[plt.plot(z, signals[i]) for i in range(len(signals))]
plt.show()
```

globalemu in action

SARAS2: https://doi.org/10.1093/mnras/stac1158



SARAS3: Submitted to Nature, Under Review

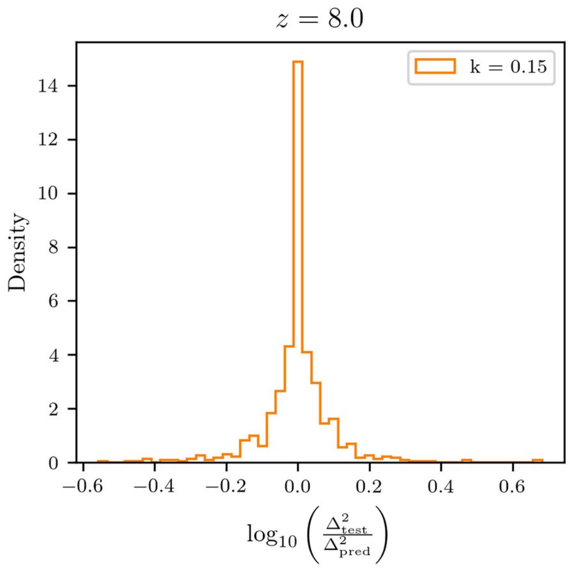


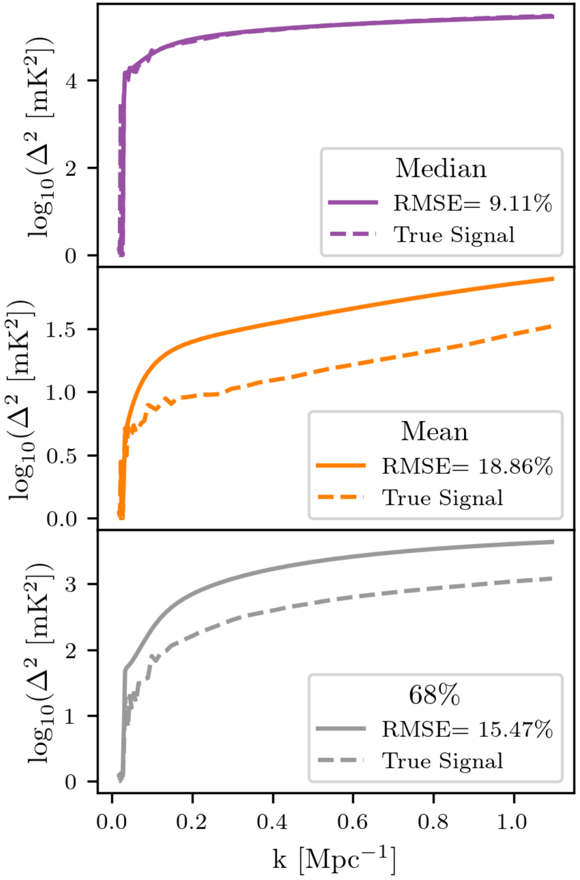
Harry T. J. Bevins, Eloy de Lera Acedo, Anastasia Fialkov, Will J. Handley, Saurabh Singh, Ravi Subrahmanyan and Rennan Barkana

Not just a global signal emulator...

- Training on some excess radio background power spectrum models and assessing at z=8 and k=0.15

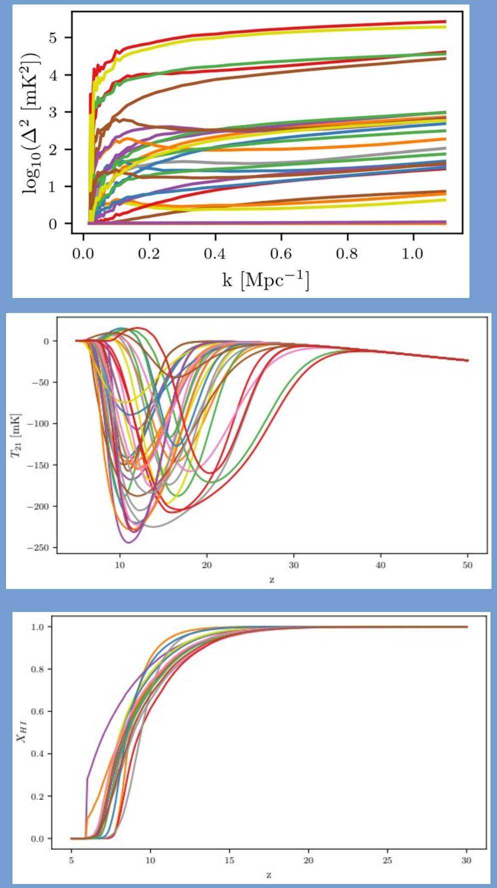
- Comparable accuracy to emulator used for recent HERA analysis (z=7.8, k=0.13)





Why is this important for the SKA?

globalemu



Conclusions

- globalemu is your one stop shop for accurate and fast 21-cm signal emulation
- For the global signal we see a factor of approximately two improvement in emulation accuracy and a factor of 102 improvement in emulation time
- For the power spectrum we see a comparable level of accuracy with the emulator used in the recent HERA analysis

globalemu paper: https://doi.org/10.1093/mnras/stab2737

globalemu in action (fitting SARAS2 data):
https://doi.org/10.1093/mnras/stac1158

github: https://github.com/htjb/globalemu