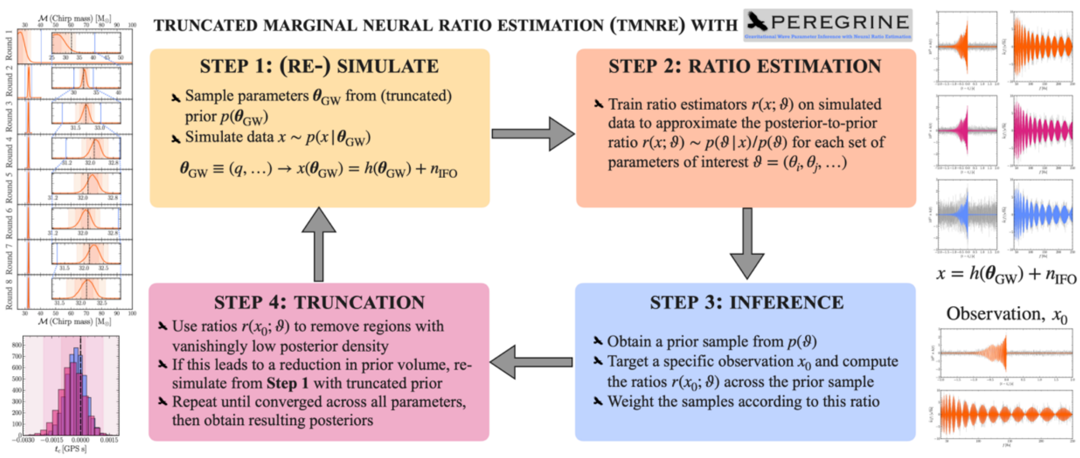
Peregrine Paper

- Sequential simulation-based inference approach to study broad classes of GW signals
- Able to fully reconstruct the posterior of a spinning, precessing compact binary coalescence using only 2% of the waveform evaluations that are used in nested sampling
- Key goal in inference is to reconstruct the posterior – for these systems, typically 12-17 parameters, so stochastic sampling methods usually used – traditionally use MCMC or nested sampling, but very computationally expensive
- Simulation based inference – highly simulation-efficient, do not require an explicit likelihood function to be written down, just a realistic forward simulator
- I.e. given a forward model *p(x, θ)* can we construct meaningful posteriors *p(θ|x)*
- NPE has been applied to perform fully amortised inference on compact binaries
- Peregrine uses TMNRE – truncated marginal neural ratio estimation based on **swyft** software
- TMNRE is truncated in the sense that estimates are done for one specific observation, and all the priors are then truncated before generating training data for the next round
- Marginal posteriors are generated – posterior for a subset of the parameters – both significantly reduce the computational budget
- The ratio we are estimating is $r(x;\vartheta) = \frac{p(x|\vartheta)}{p(x)} = \frac{p(\vartheta|x)}{p(\vartheta)} = \frac{p(x,\vartheta)}{p(x)p(\vartheta)}$ from Bayes' theorem - ϑ is a subset of the parameters – this becomes a binary classification problem – we set up a classifier with trainable parameters that outputs 0 when the data is drawn marginally and 1 when jointly – we minimise the binary loss cross-entropy loss function
- There is an analytic expression for this ratio in terms of the optimal classifier function, so we can recover the posterior to prior ratio
- A lot of this data is very high dimensional, so we use neural network architectures to do these calculations
- The TMNRE algorithm in steps is:
    - 1. Generate a batch of simulations from proposal prior and forward simulator
    - 2. Train classifiers to tell the difference between joint and marginal samples and get the ratio as above
    - 3. Get ratio estimates for one specific target observation
    - 4. Truncate the initial proposal distribution to exclude regions of very low posterior density so variance in training data is very low
    - 5. Repeat until the posterior estimate stabilises
- UNet architectures are implemented – it uses two separate, parallelly running UNets on the time domain data and the frequency domain data
- The comparison between peregrine and other methods was done using the Jensen-Shannon divergence



Peregrine Paper