Gaussian process regression for gravitational astrophysics

Daniel Williams
Institute for Gravitational Research
University of Glasgow

INSPIRING PEOPLE

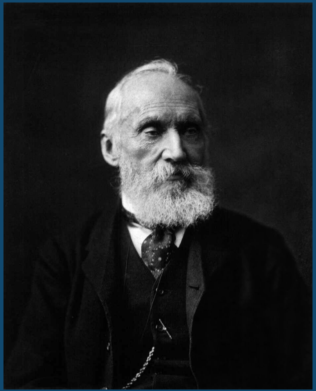
A little about me

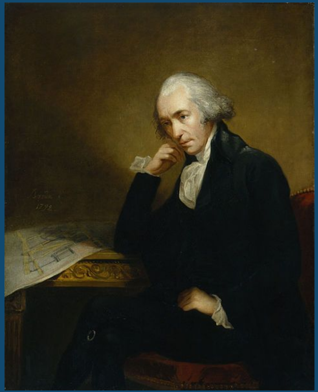
- Postdoc at University of Glasgow
    - Previously: PhD University of Glasgow
    - Member of GEO / LIGO

- Research foci:
    - statistical techniques for approximating GW waveforms
    - multimessenger astrophysics with gravitational-waves
    - gravitational-wave burst search sensitivity measurement

A little about the University of Glasgow

- 4th-oldest university in English-speaking world
    - Founded 1451
    - 2nd-oldest in Scotland
    - Centre of the Scottish Enlightenment





VIA VERITAS VITA

Institute for Gravitational Research

- Founded in 2000 by Prof Sir James Hough
    - ~70 members
- Research foci
    - Gravitational-wave detectors
        - Mirror suspensions
        - Mirror coatings
        - Interferometer R&D
    - Gravitational-wave data analysis
        - Bayesian & machine learning techniques
        - Continuous wave [GW pulsar] sources (Woan, Bayley, [Pitkin])
        - Compact binaries (Veitch, Williams, Messenger, *Gabbard*, *Williams*)
        - Bursts (Heng, Williams, McGinn)
        - GW cosmology (Hendry, Messenger, *Gray*)
        - Multimessenger (Heng, Williams, *Hayes*, *Datrier*)
    - Miniaturisation of gravimeters

Outline

1. Black hole waveforms from Numerical Relativity
2. An introduction to Gaussian Process Regression
3. The *Heron* waveform model
4. Using Conditional Variational Autoencoders for Bayesian Inference

Binary black hole waveforms

Adventures in General Relativity

A need for waveforms

- Gravitational wave searches rely on **matched filtering** where template waveforms generated from theoretical models are compared to noisy detector data
- The accuracy of the waveform directly affects the performance of the search

Important for

- Signal detection in noisy data-streams
- GW parameter estimation
- Testing general relativity in strong-field scenarios

Solving the Einstein Field Equations

- Studying compact binaries using GWs relies on solving the relativistic 2-body problem

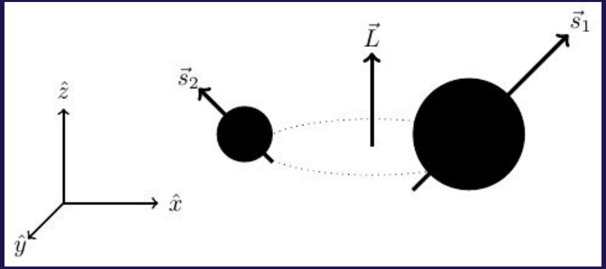
- In *Newtonian gravity* the 2-body has solutions in the form of *Keplerian orbits*

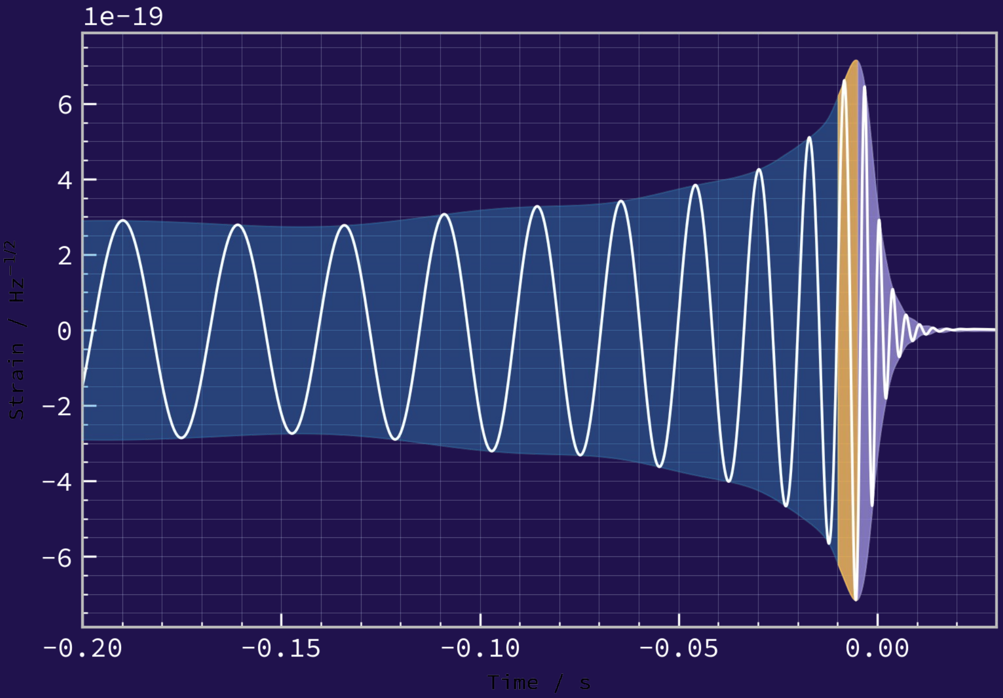
- Weak-field scenarios in GR can be approximated through *post-Newtonian expansion*
    - E.g. analysis of the precession of the orbit of Mercury

- **Strong**-field is analytically intractable.

Binary black hole systems

- Compact system of two black-holes
- Orbit inspirals due to energy lost as GWs
- Waveform affected by spin-spin and spin-orbit effects





Numerical relativity

- Numerical solution of the EFEs became possible in mid-2000s.
- Numerical relativity simulations produce most precise waveforms available.
- NR waveforms are *very* slow to produce: can take weeks to run, and cost many thousands of dollars.

Numerical relativity: Catalogues

- Largest source of NR waveforms is the **SXS Catalogue**
    - https://data.black-holes.org/waveforms/
    - arXiv:1904.04831

- Catalogues also available from
    - Georgia Tech: arXiv: 1605.03204
    - RIT: 1703.03423

Approximant models

Two major approaches to approximating the waveform:

Effective one-body approximation
(S)EOBNR-family waveforms

- Use simplified physics
- Quite accurate
- (Very) Slow

Phenomenological fitting
IMRPhenom-family waveforms

- Use analytical fits to the three stages of the waveform
  - Inspiral (post-Newtonian)
  - Merger
  - Ringdown (Lorentzian decay)
- Fast
- Less accurate

Both calibrated against **NR**-derived waveforms.

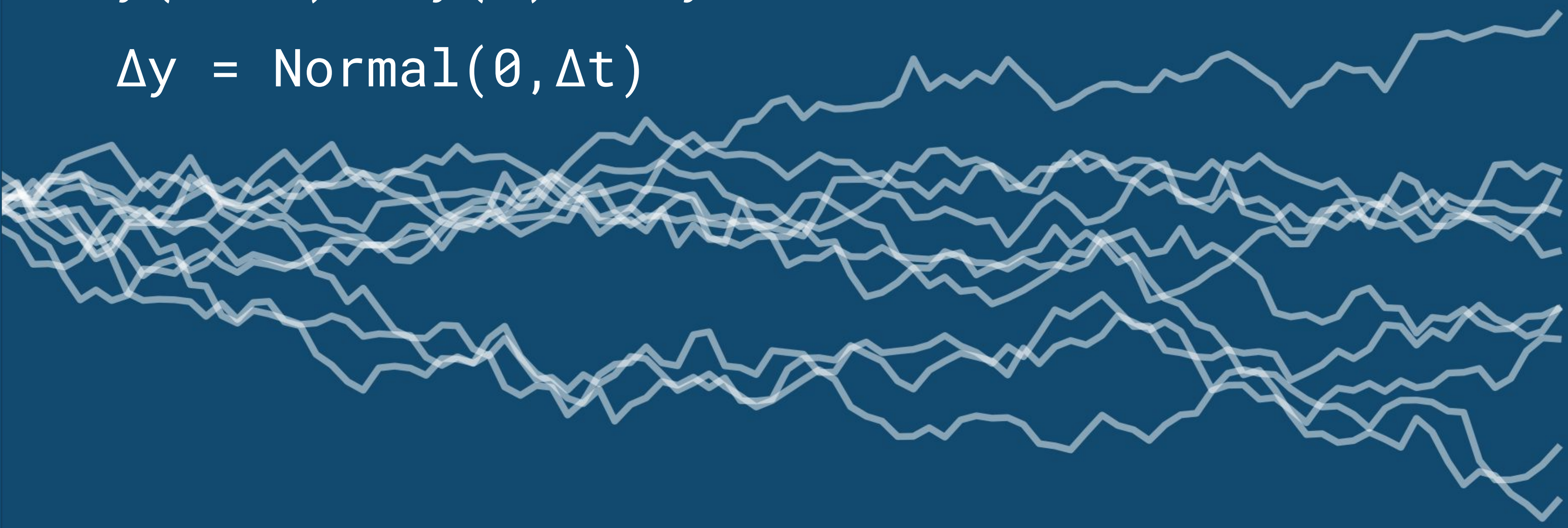
Statistical function modelling

Random walks

Stochastic process

$$y(t+\Delta t) = y(t) + \Delta y$$

$$\Delta y = \mathrm{Normal}(0, \Delta t)$$



Gaussian process

$$\text{data} = \{(\mathbf{x},\ y)_i,\ i \in 1, \ldots, n\}$$

$$y_i(\mathbf{x}_i) = f(\mathbf{x}_i) + \varepsilon_i$$

$$\varepsilon_i \sim \text{Normal}(0, \sigma^2)$$

Gaussian process

$$p(f \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \mathrm{Normal}_N(0, \underline{\mathbf{K}})$$

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \lambda)$$

Gaussian process

$$f = f(\mathbf{x}),$$
$$f_* = f(\mathbf{x}_*)$$

Posterior

$$p(f, f_* | y) \propto p(f, f_*)\, p(y | f)$$

Prior

likelihood

Gaussian process

Stochastic process

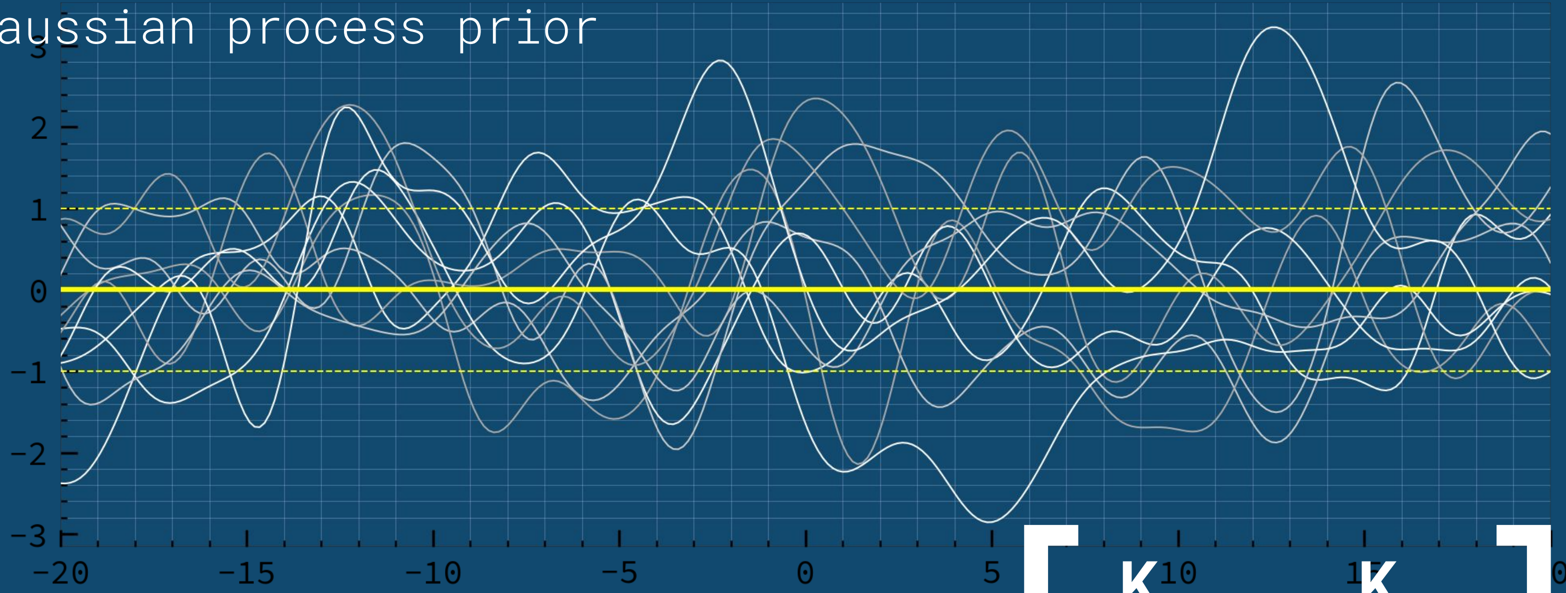
$$p(f_*|y) = \int p(f, f_*|y)\, df$$

$$\propto \int p(f, f_*)\, p(y|f)\, df$$

$$\propto \int \mathrm{Normal}_N(0, \underline{\mathbf{K}}^{+})\, \mathrm{Normal}(f,\ \sigma^2\mathbf{I})\, df$$

Gaussian process

The Gaussian process prior



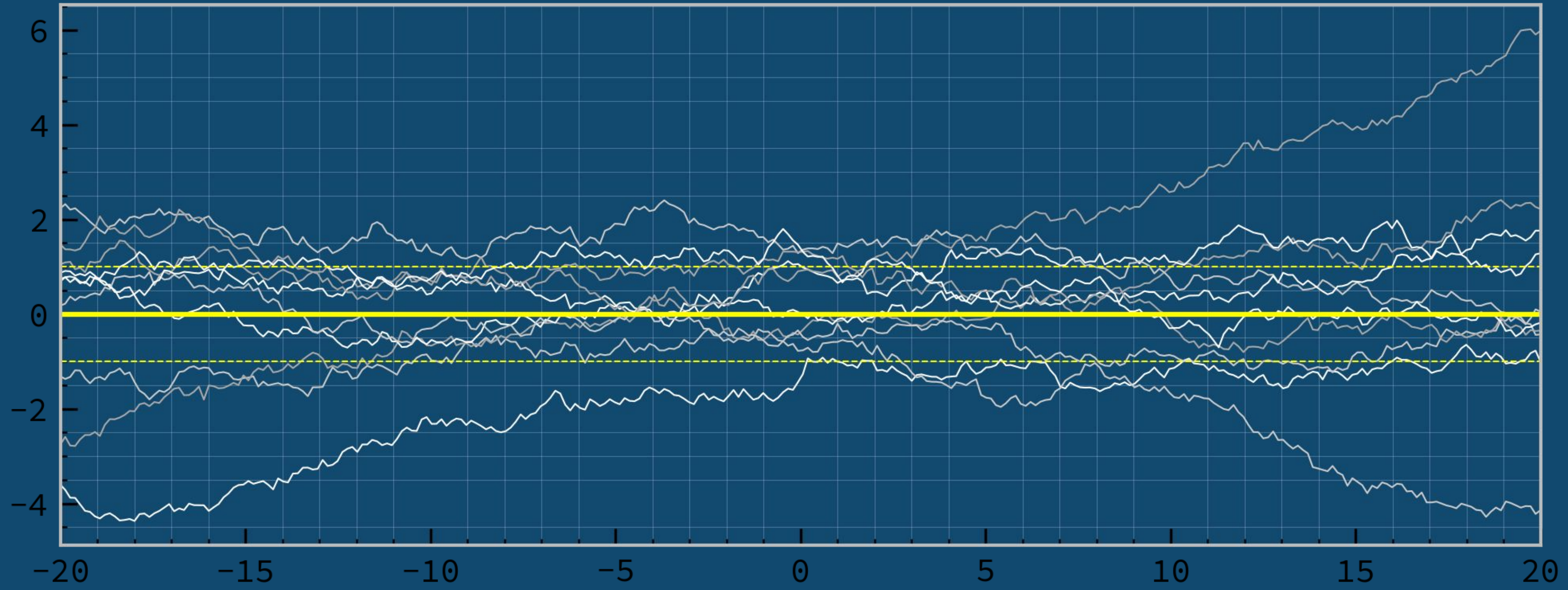
$$p(f, f_*) = \mathrm{Normal}(\mathbf{0}, \underline{\mathbf{K}}^{+}) \qquad \underline{\mathbf{K}}^{+} = \begin{bmatrix} \underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}} & \underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}'} \\ \underline{\mathbf{K}}_{\mathbf{x}',\mathbf{x}} & \underline{\mathbf{K}}_{\mathbf{x}',\mathbf{x}'} \end{bmatrix}$$

Gaussian process

Covariance functions

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \lambda)$$

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j; \lambda) = \exp(½|\mathbf{x}_i, \mathbf{x}_j| \cdot \lambda^{-2})$$


6
4
2
0
−2
−4
−20
−15
−10
−5
0
5
10
15
20

Gaussian process

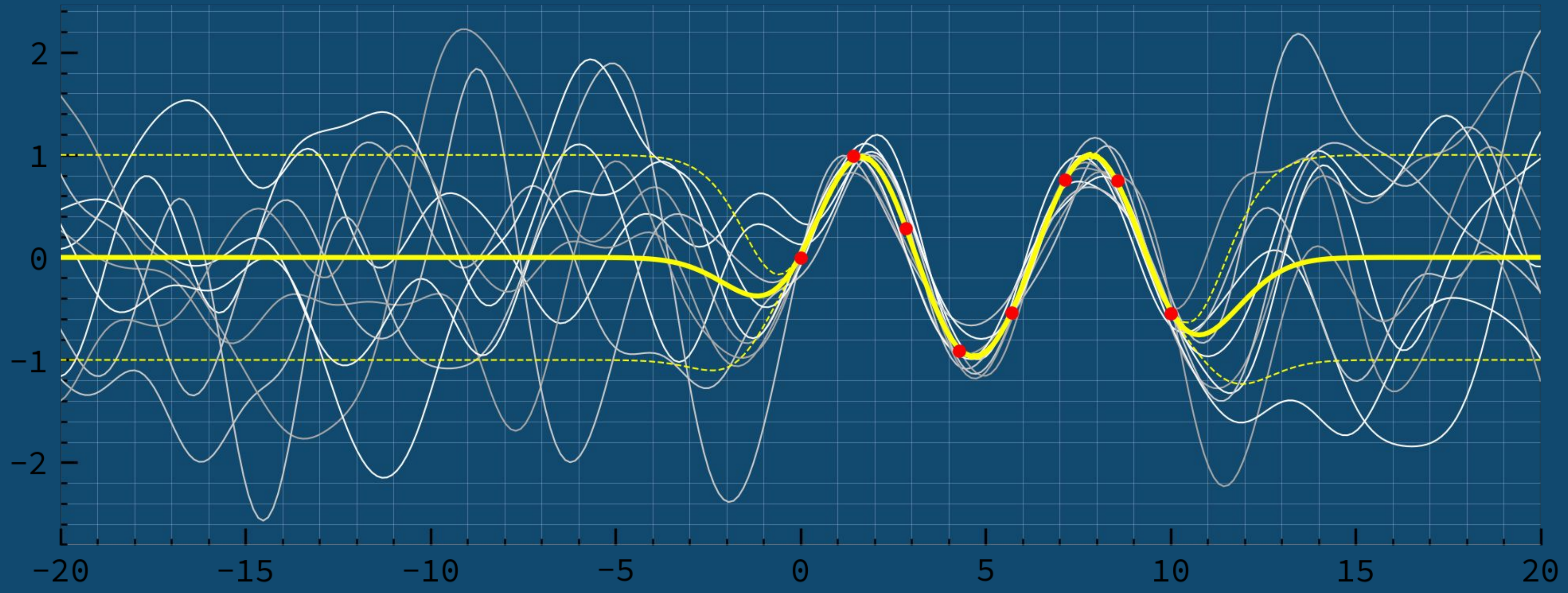
Marginalised posterior

$$\underline{\mathbf{K}}^{+} = \begin{bmatrix} \underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}} & \underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}'} \\ \underline{\mathbf{K}}_{\mathbf{x}',\mathbf{x}} & \underline{\mathbf{K}}_{\mathbf{x}',\mathbf{x}'} \end{bmatrix}$$

$$p(f_*|y) = \mathrm{Normal}(\boldsymbol{\mu},\boldsymbol{\Sigma})$$

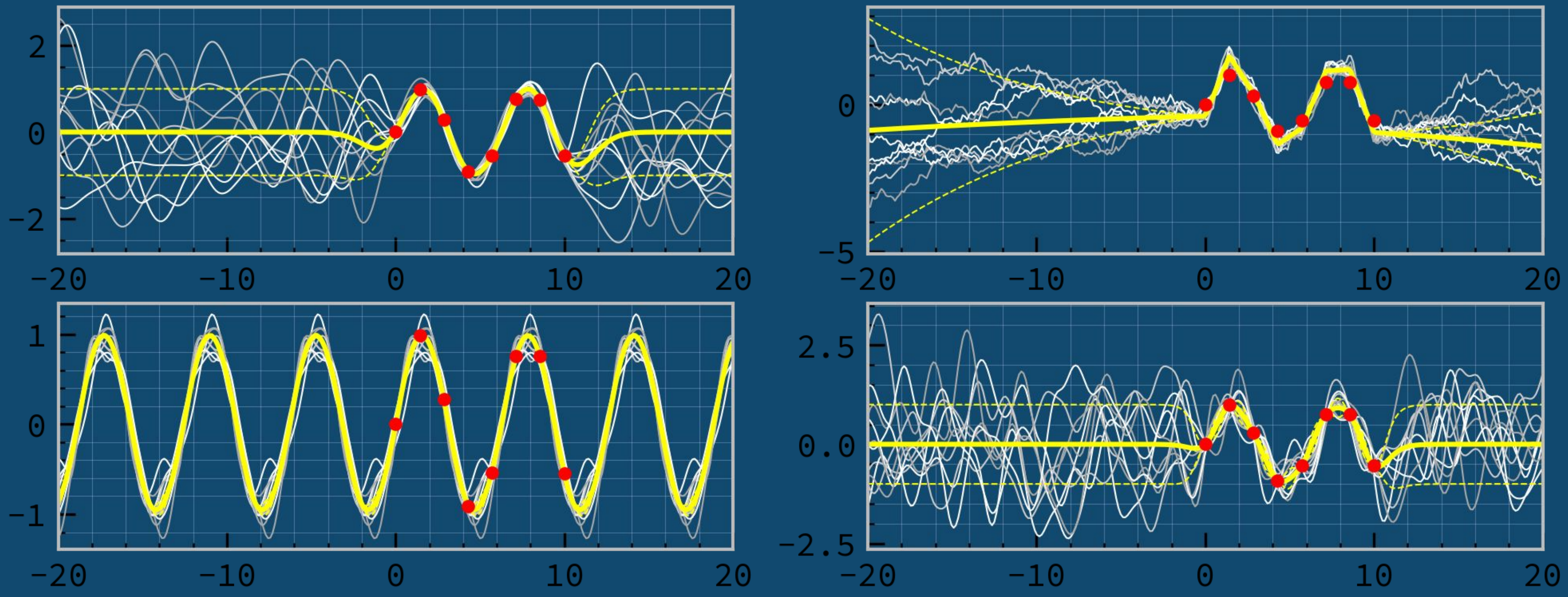
$$\boldsymbol{\mu} = \underline{\mathbf{K}}_{\mathbf{x}',\mathbf{x}}\,(\underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I})^{-1}\,\mathbf{y}$$

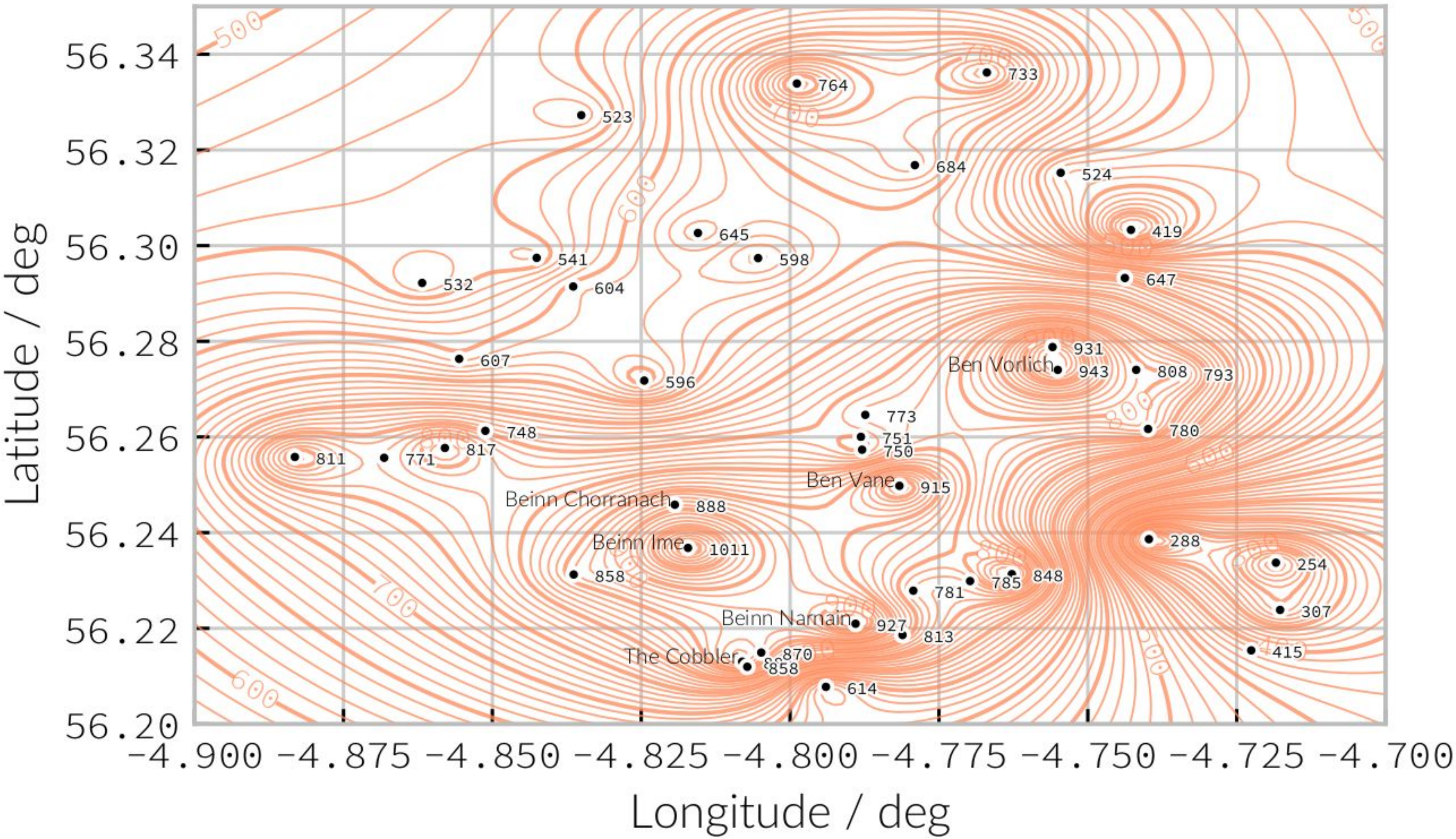
$$\boldsymbol{\Sigma} = \underline{\mathbf{K}}_{\mathbf{x}',\mathbf{x}'} - \underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}'}\,(\underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I})^{-1}\,\underline{\mathbf{K}}_{\mathbf{x},\mathbf{x}'}$$


2
1
0
−1
−2
−20
−15
−10
−5
0
5
10
15
20

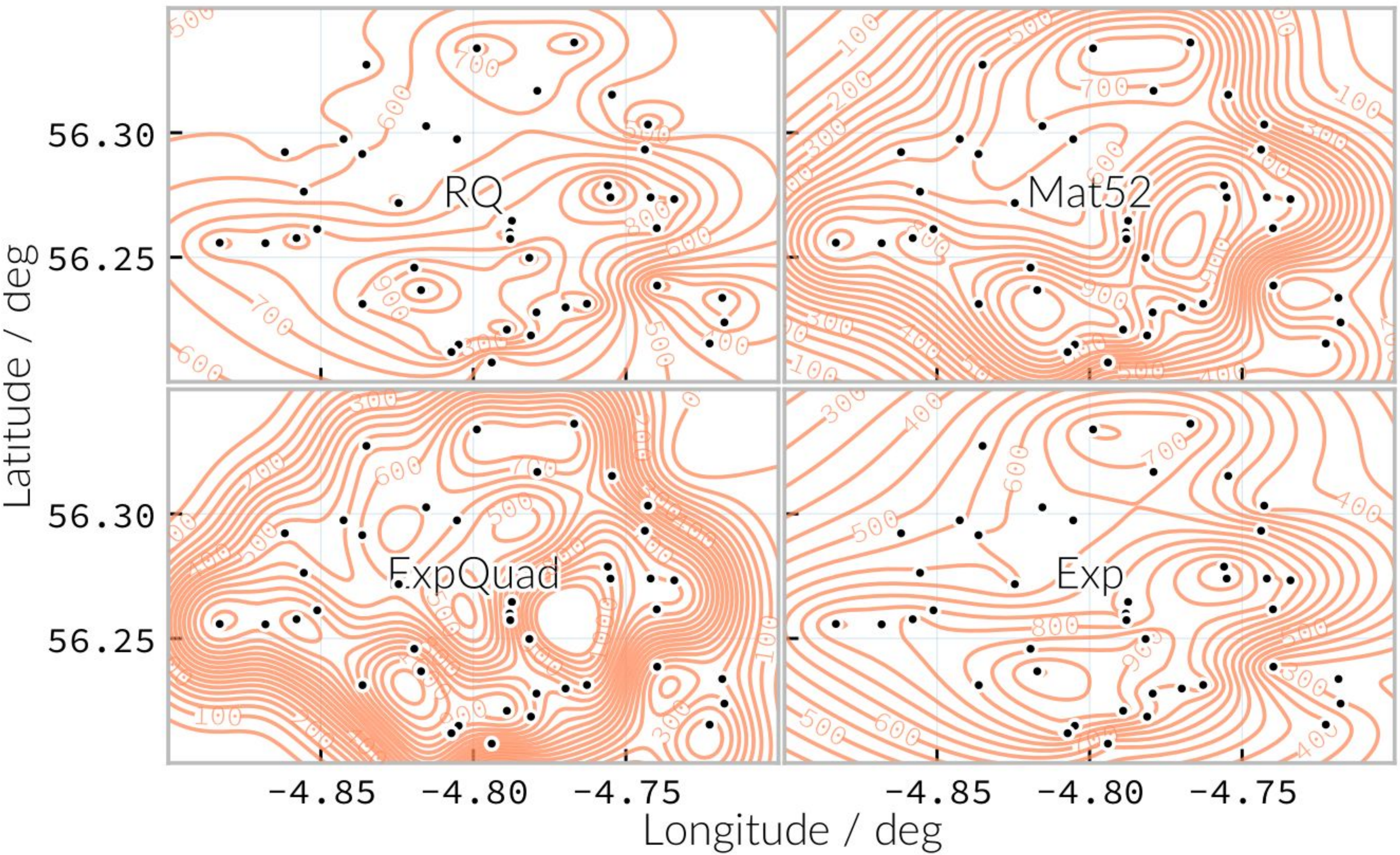
Gaussian process

Covariance functions revisited





Latitude / deg
Longitude / deg
56.34
56.32
56.30
56.28
56.26
56.24
56.22
56.20
-4.900
-4.875
-4.850
-4.825
-4.800
-4.775
-4.750
-4.725
-4.700
764
733
523
684
524
419
645
598
541
532
604
647
931
Ben Vorlich
943
808
793
607
596
773
751
750
748
811
771
817
780
Ben Vane
915
Beinn Chorranach
888
Beinn Ime
1011
858
288
254
848
785
781
307
Beinn Narnain
927
813
The Cobbler
870
858
415
614
500
600
700
800


RQ
Mat52
ExpQuad
Exp
Latitude / deg
Longitude / deg
56.30
56.25
-4.85
-4.80
-4.75

Gaussian process

Covariance functions - learning hyperparameters

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j;\ \lambda) = \exp(½|\mathbf{x}_i, \mathbf{x}_j|\cdot\lambda^{-2})$$

- Hyperparameters (e.g. λ) give us flexibility in specifying the form of the covariance function

- We can determine optimal values for these by maximising the evidence of the data-conditioned Gaussian process e.g. using Gradient descent (empirical Bayesian approach) or MCMC.

왜가리과

Heron

A GPR and NR-driven BBH model

Design goals

- Create a waveform model which maintains a measure of output uncertainty
  - **Gaussian process regression** an ideal approach
  - Important for use in Bayesian inference e.g. for LIGO parameter estimation
- Create a waveform model using NR waveforms directly
  - No physical approximations
  - Minimal assumptions about functional form of the waveform

Model design

- **Two** Gaussian processes
  - One to model $\mathbf{h}_+$ polarisation
  - One to model $\mathbf{h}_\times$ polarisation

- Assumption that the waveforms will be *smooth*
  - Important for choice of covariance function

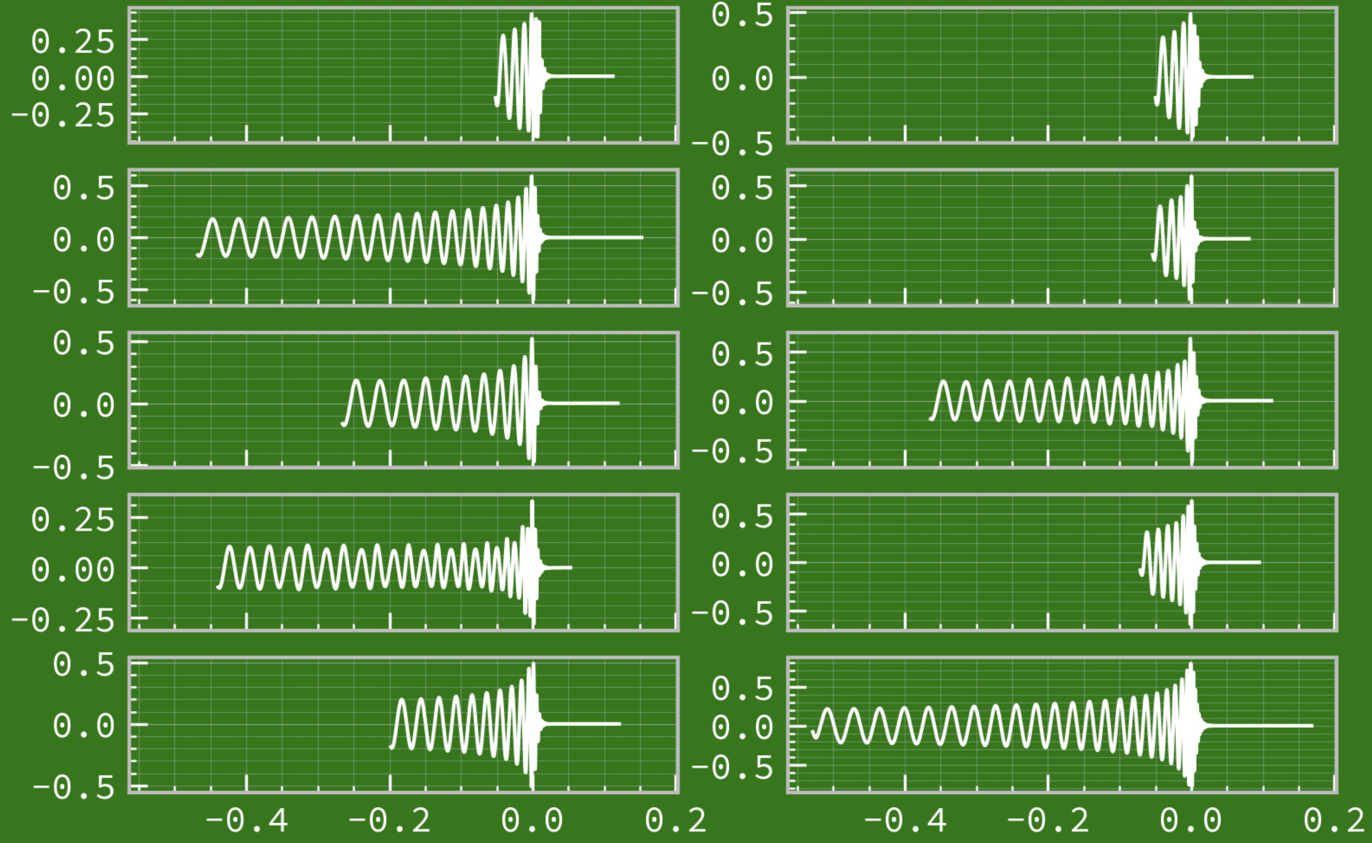
- Covariance function:

  8-dimensional Squared-exponential kernel

- Trained using ADAM and mini-batches

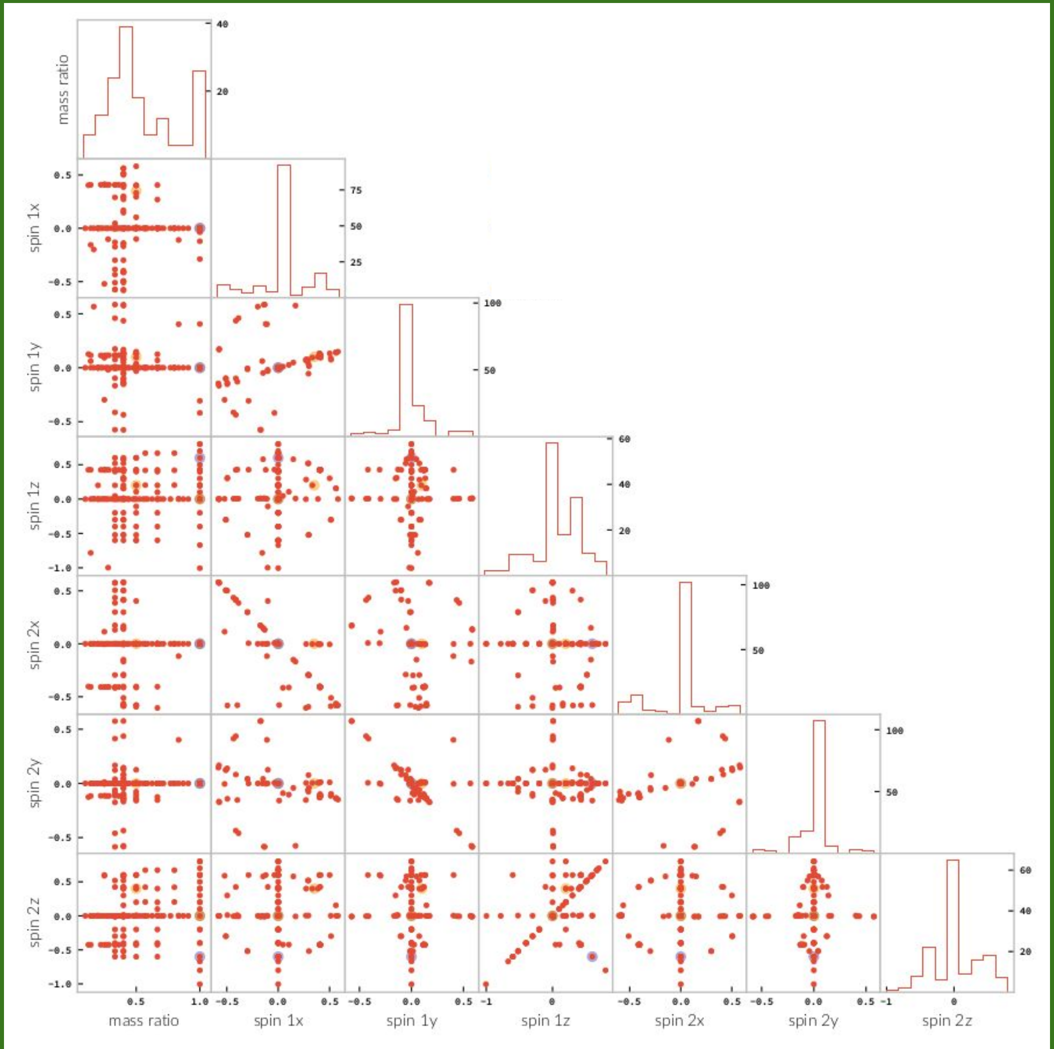
- Employs a matrix inversion approximation to cope with memory requirements

Some waveforms from the GT catalogue

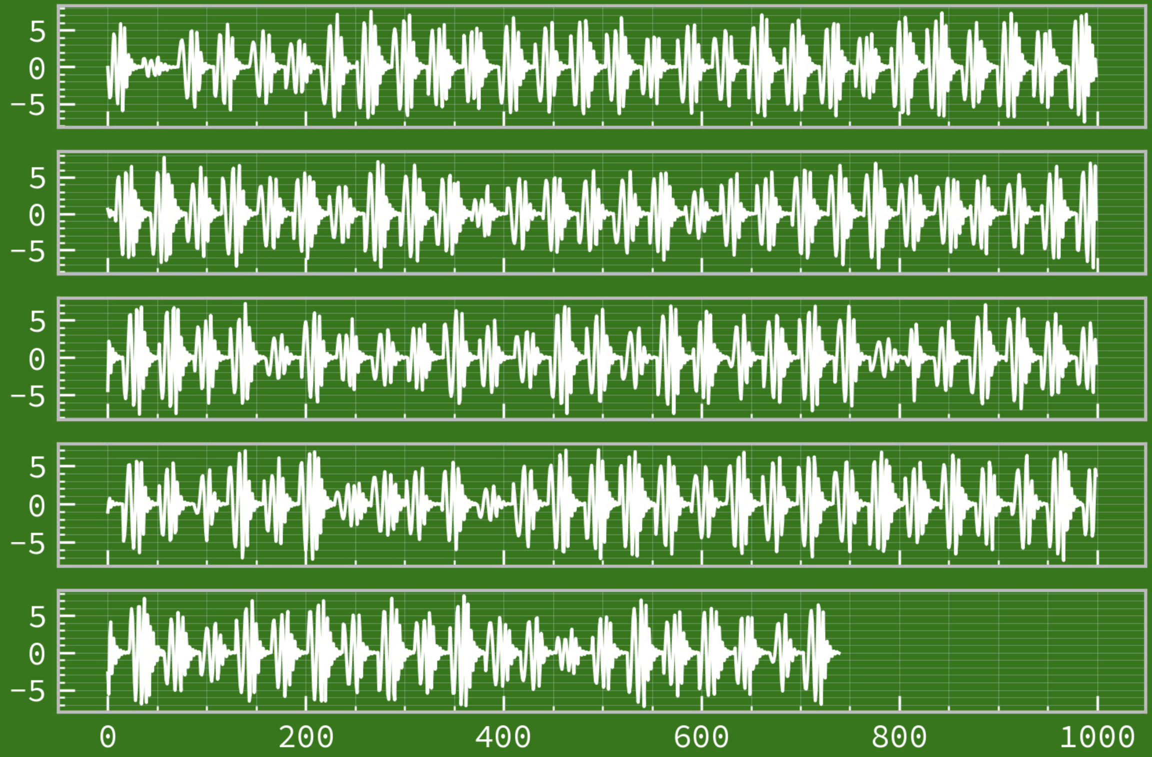


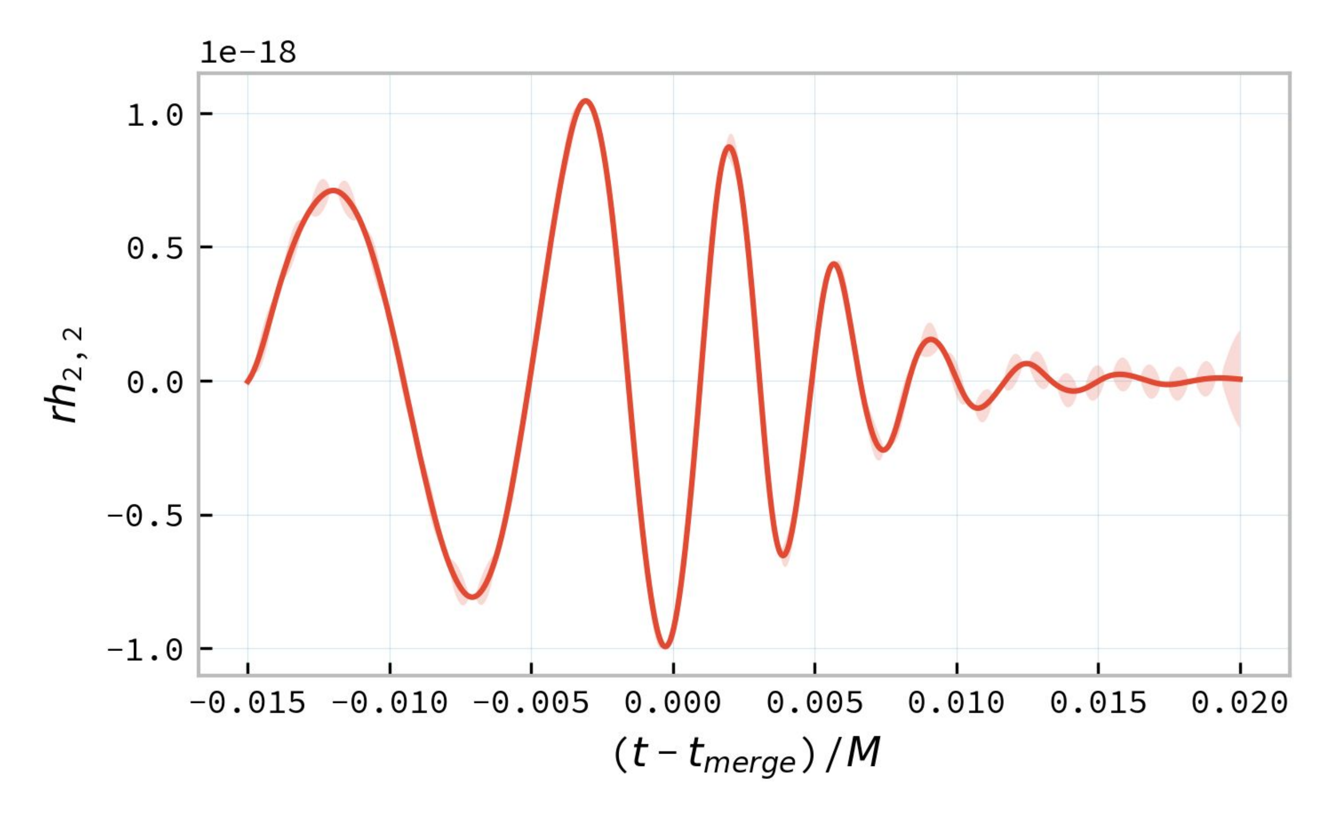
Training data

Waveforms span the 7-dimensional parameter space, but regions of the parameter space often *very* sparsely sampled



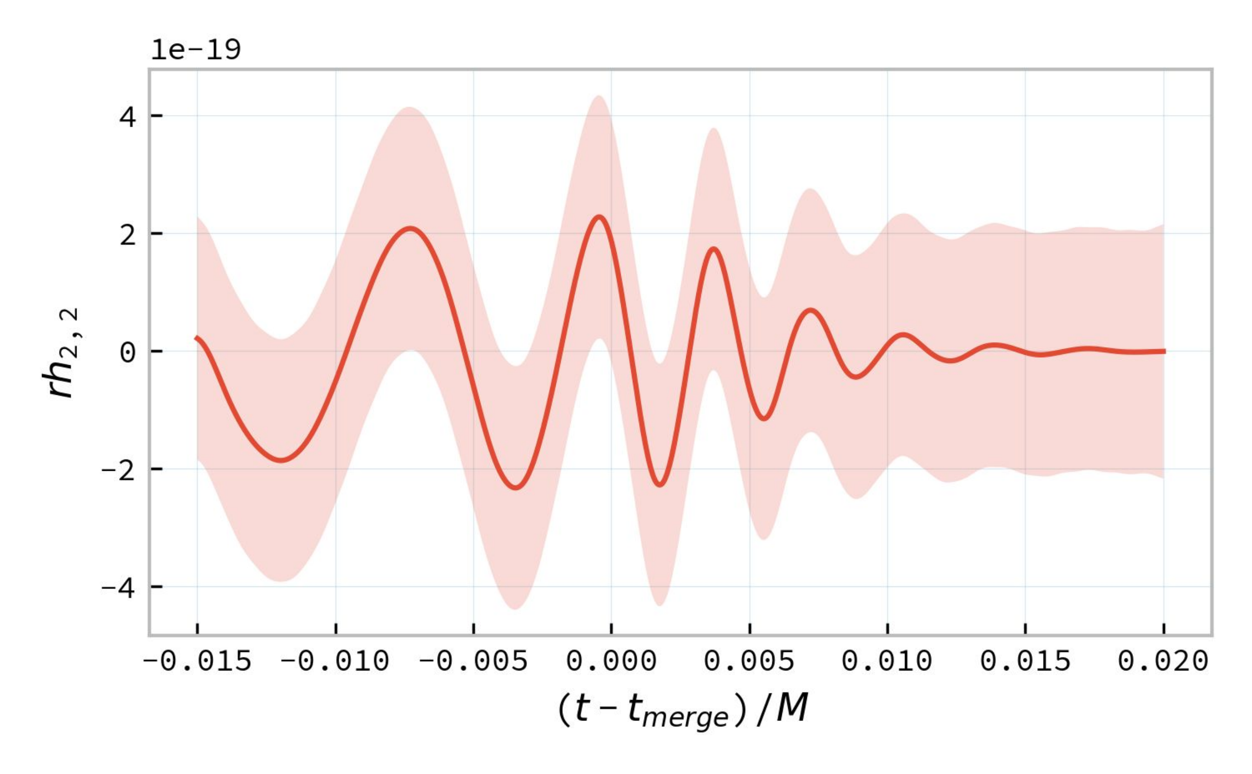
Training data





mass-ratio	1.0
Spin 1x	0
Spin 1y	0
Spin 1z	0
Spin 2x	0
Spin 2y	0
Spin 2z	0

Non-spinning waveform (h_+)



mass-ratio	0.5
Spin 1x	0
Spin 1y	0
Spin 1z	0.1
Spin 2x	0
Spin 2y	0
Spin 2z	0.25

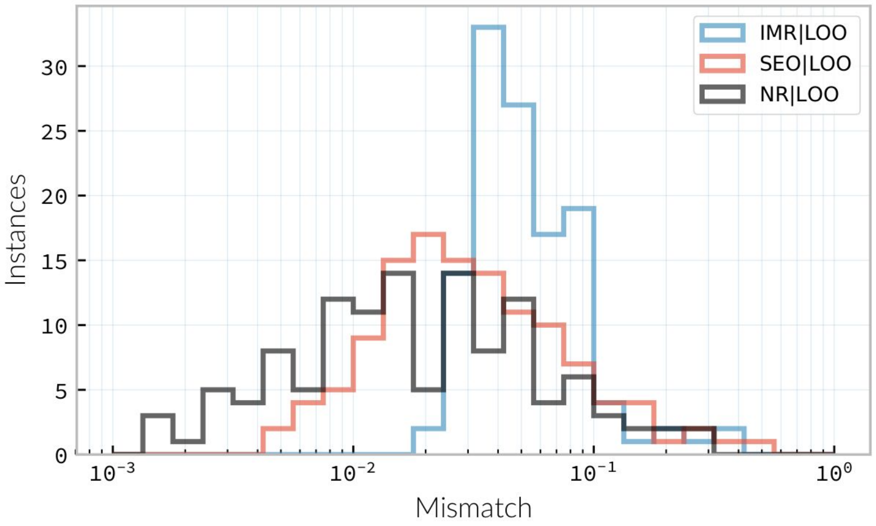
Aligned-spin waveform (h_+)

Testing

“Leave-one-out” tests

1. Choose waveform from GT catalogue
2. Remove this waveform from the Heron training set
3. Retrain Heron model with one waveform removed
4. Calculate the mismatch between the Heron waveform and the GT waveform

Repeat for all GT waveforms

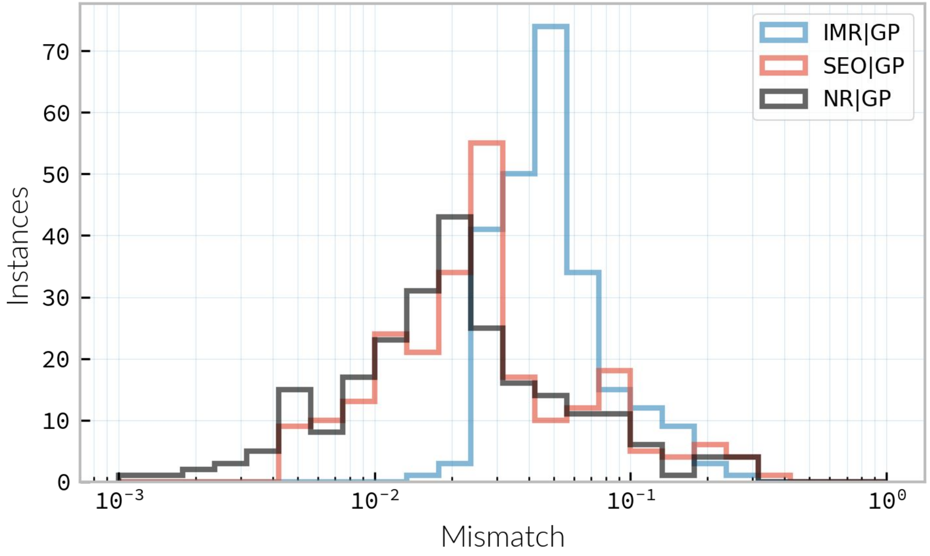


Testing

Comparison with SXS

1. Choose waveform from SXS catalogue
2. Generate corresponding waveform from *Heron* GPR model
3. Calculate the mismatch between the two waveforms

Repeat for all SXS waveforms



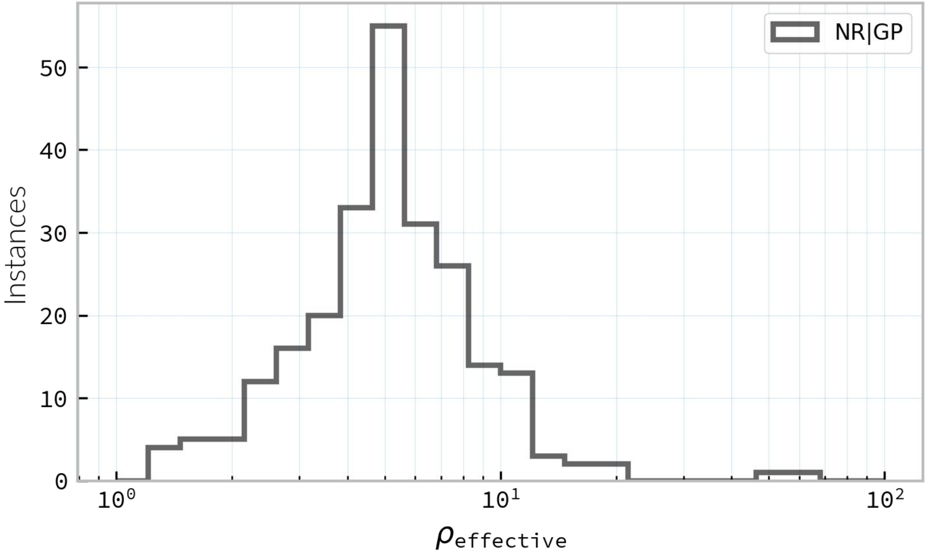
Testing

Are *Heron* waveforms good enough?

- Waveform accuracy becomes more important as SNR increases
    - At low SNR noise exceeds uncertainty introduced by waveform inaccuracies

$$\rho_{\mathrm{eff}} = (2[1-\mathcal{M}])^{-0.5}$$

- Defines the SNR at which waveform systematics exceed uncertainty from



Roadmap

- Much work still to be done
  - Speed improvements
  - Larger training set i.e. more NR waveforms
  - Improved handling of precession effects
  - Produce longer waveforms - more inspiral cycles
  - Interface with parameter estimation code e.g. *bilby*

Current **proof-of-concept** version described in

arXiv:1903.09204

Code available at

https://github.com/transientlunatic/heron

Please forgive the currently sparse documentation: I'm working on it!

Conclusions

The good, the bad, and the ugly

The good

- Provides data-driven model with no assumptions about underlying physics
- Provides waveform uncertainties throughout parameter space

The bad

- Quite slow, gets slower as more data added

The ugly

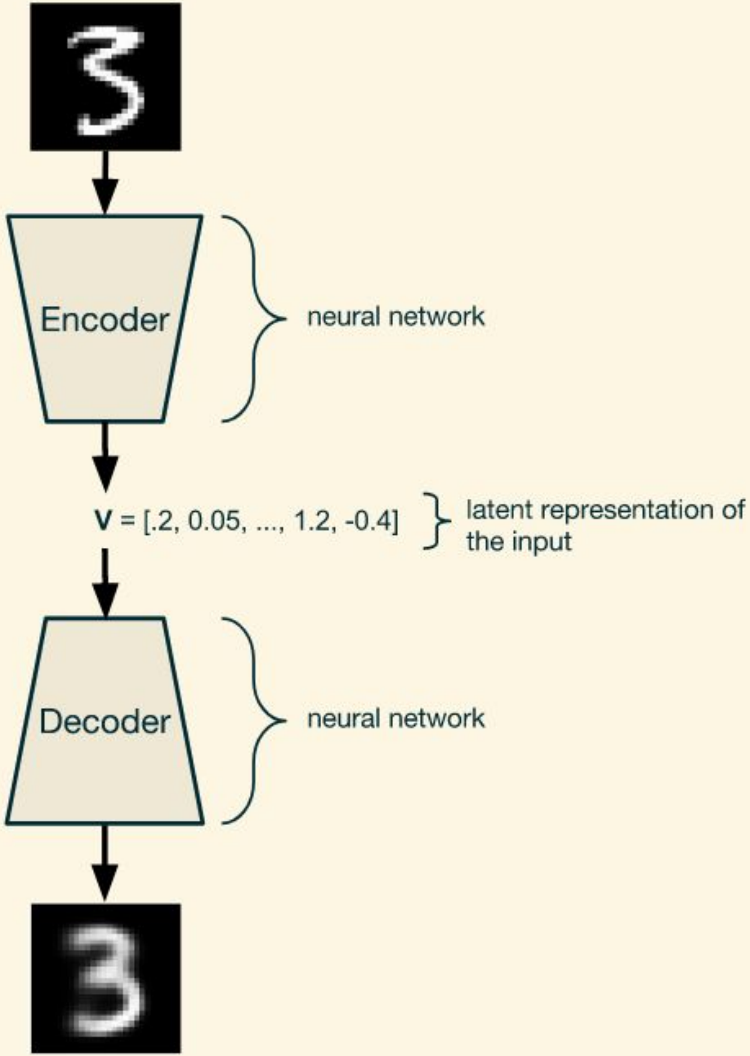
- Sparse training data available (though this is getting better)
- Much work required to integrate in existing Bayesian inference pipelines

Variational autoencoders

Accelerating Bayesian inference with VItamin

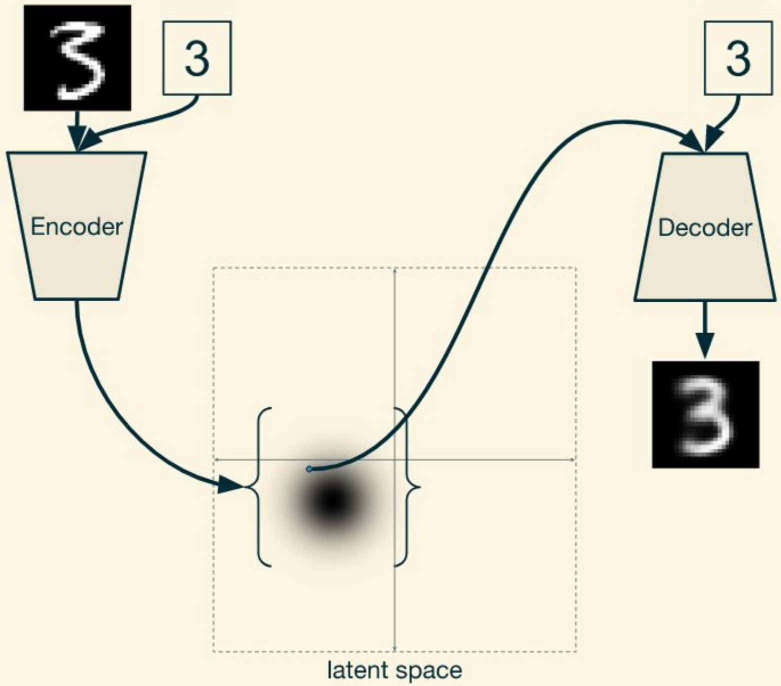
Autoencoders

- Two networks
  - One learns mapping from data-space to (latent) feature space
  - One learns mapping from feature space to data space
- Useful for compression & dimensionality reduction



Conditional variational autoencoders

- Two networks
  - One learns mapping from data-space to a distribution in feature space
  - One learns mapping from points feature space to data space



VItamin

- Project to develop a network which can map **noisy data** vector to a distribution in feature space
- Goal to produce a fast way to approximate Bayesian posteriors
- “Estimating Bayesian parameter estimation”

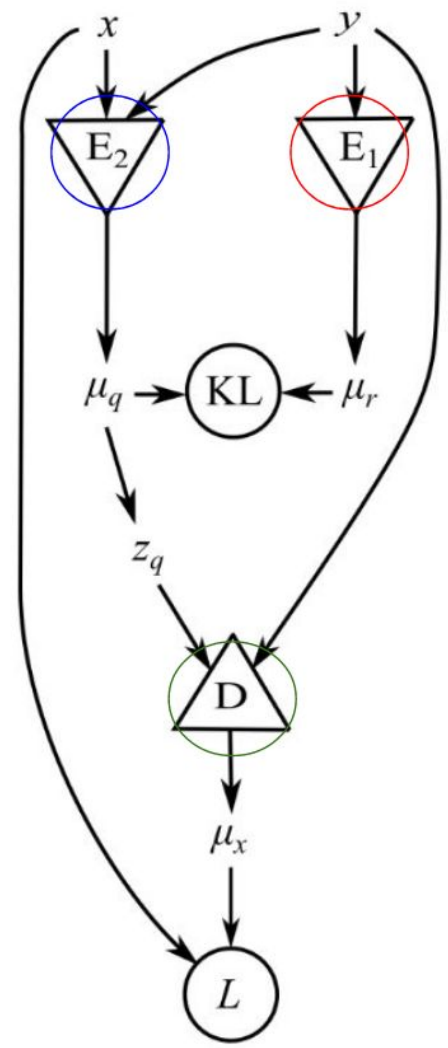
See **Gabbard+2019**

“Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy”

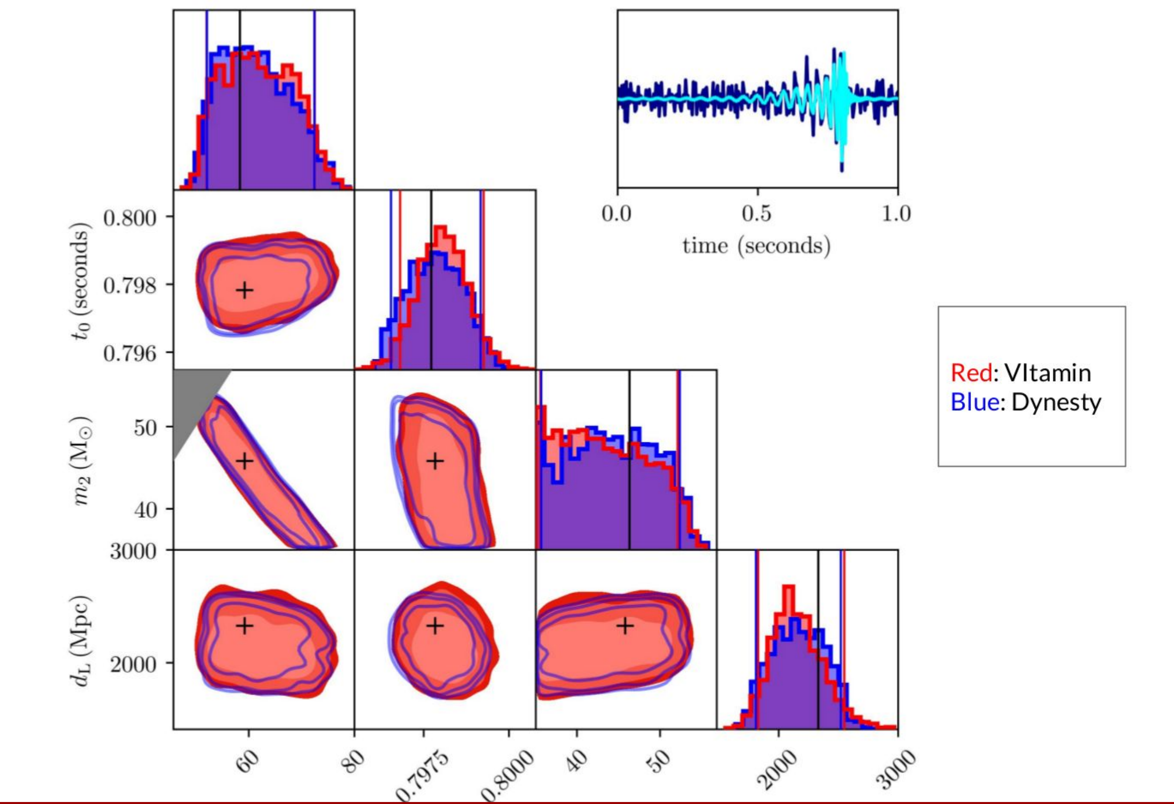
arxiv:1909.06296

How VItamin works

- Two encoder networks are trained
  - One sees the noisy waveforms and also the true parameters
  - One sees only the noisy waveform
- The encoders are trained to minimise the KL divergence of the distributions they produce
- Samples are drawn from the distribution from the all-seeing network
- Discriminator is trained by minimising the difference between discriminator output and the true parameters



Comparing nested sampling to VItamin



VItamin : speed improvements

	run time (seconds)			
	min	max	median	ratio
Dynesty[a]	602	1538	774	2.6×10^{-6}
Emcee	2005	11927	4351	4.6×10^{-7}
Ptemcee	3354	12771	4982	4.0×10^{-7}
Cpnest	1431	5405	2287	8.8×10^{-7}
`VItamin`[b]		$\mathbf{2 \times 10^{-3}}$		1

VItamin Conclusions

Opportunities

- Potential for 7 orders of magnitude speed-up compared to conventional Bayesian methods

Challenges

- Need to scale-up to all intrinsic and extrinsic parameters
- Still problems emulating multimodal posteriors
- Working with non-Gaussian noise realisations

Any questions?

📧 daniel.williams@glasgow.ac.uk
🐦 @daniel_williams
🐦 @UofGravity

Science and Technology Facilities Council