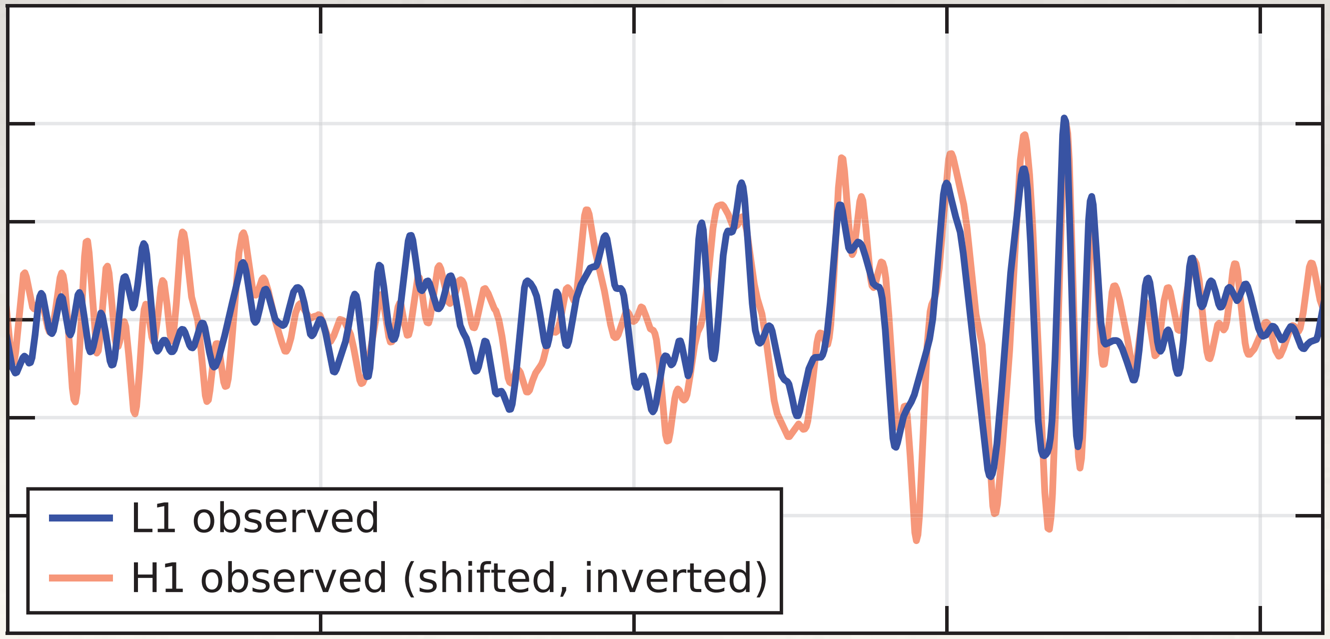
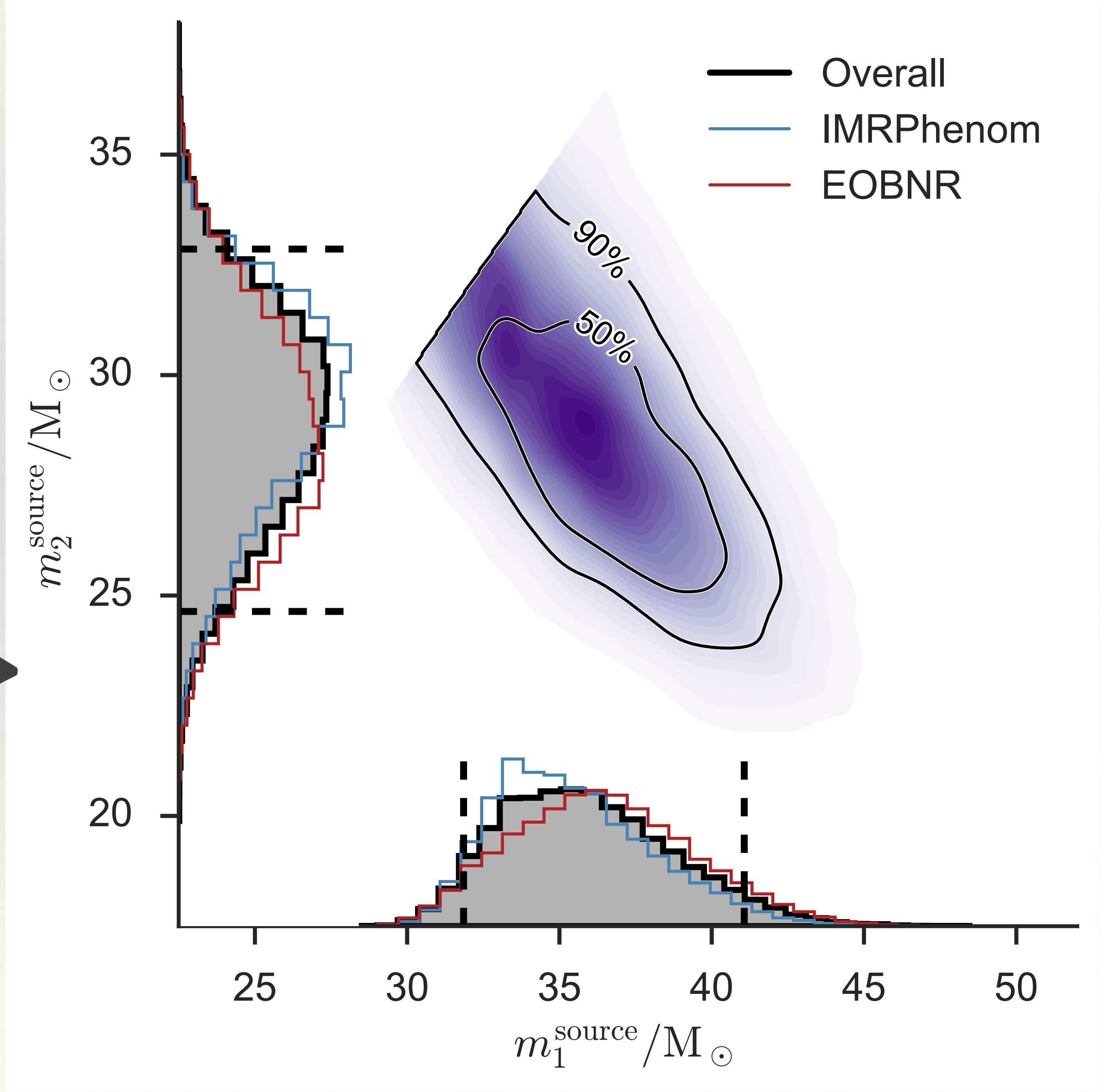
AI for GW emulation and parameter estimation

Geneva *AI/ML for Gravitational Waves* event

Jonathan Gair, Albert Einstein Institute (Potsdam)

in collaboration with: **Stephen Green**, **Max Dax**, Jakob Macke, Michael Pürrer, Bernhard Schölkopf, Alessandra Buonanno



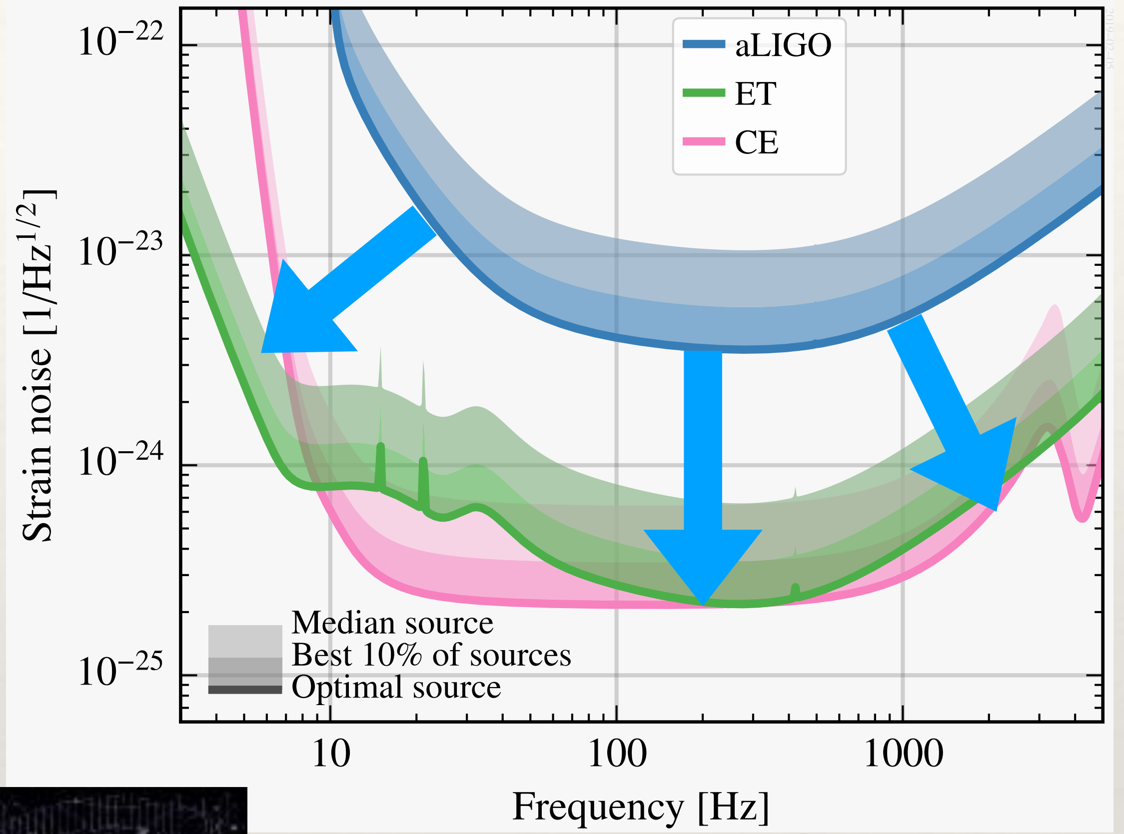


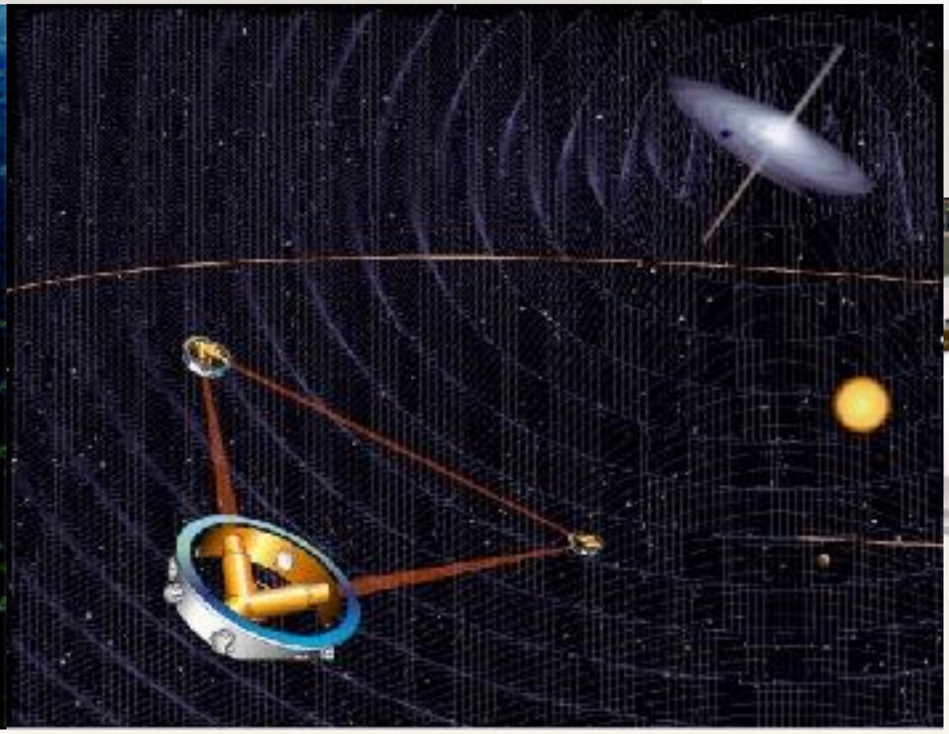
Talk outline

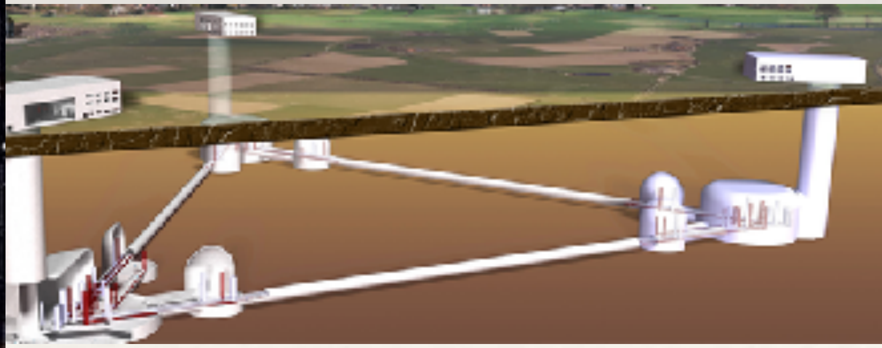
- Current and future gravitational wave detectors
- Gravitational wave parameter estimation
- Emulation of gravitational wave models using Gaussian processes (**briefly**)
- Fast and accurate parameter estimation using neural posterior estimation

Current and future detectors

- **LIGO/Virgo/KAGRA**: Ground-based interferometers currently operating. 100 (likely) astrophysical sources observed to date, over three observing runs.
- **LISA**: space-based interferometer to launch in ~2035, operating in mHz band. ESA-led; NASA contributions,
- **3G**: next generation ground-based detector concepts under development. **Einstein Telescope** (Europe) and **Cosmic Explorer** (US). To start operation in ~2030s.







Overview of GW parameter estimation

- GW parameter estimation typically uses Bayesian inference, in which we obtain samples from the *posterior distribution* after specifying a *prior distribution* and the *likelihood*

$$p(\vec{\theta}|d) = \frac{p(d|\vec{\theta})p(\vec{\theta})}{p(d)}$$

- Typically we assume the detector output is a linear combination

$$s(t) = n(t) + h(t;\vec{\theta})$$

- and that the noise is Gaussian, giving the likelihood

Waveform models are expensive to compute accurately!

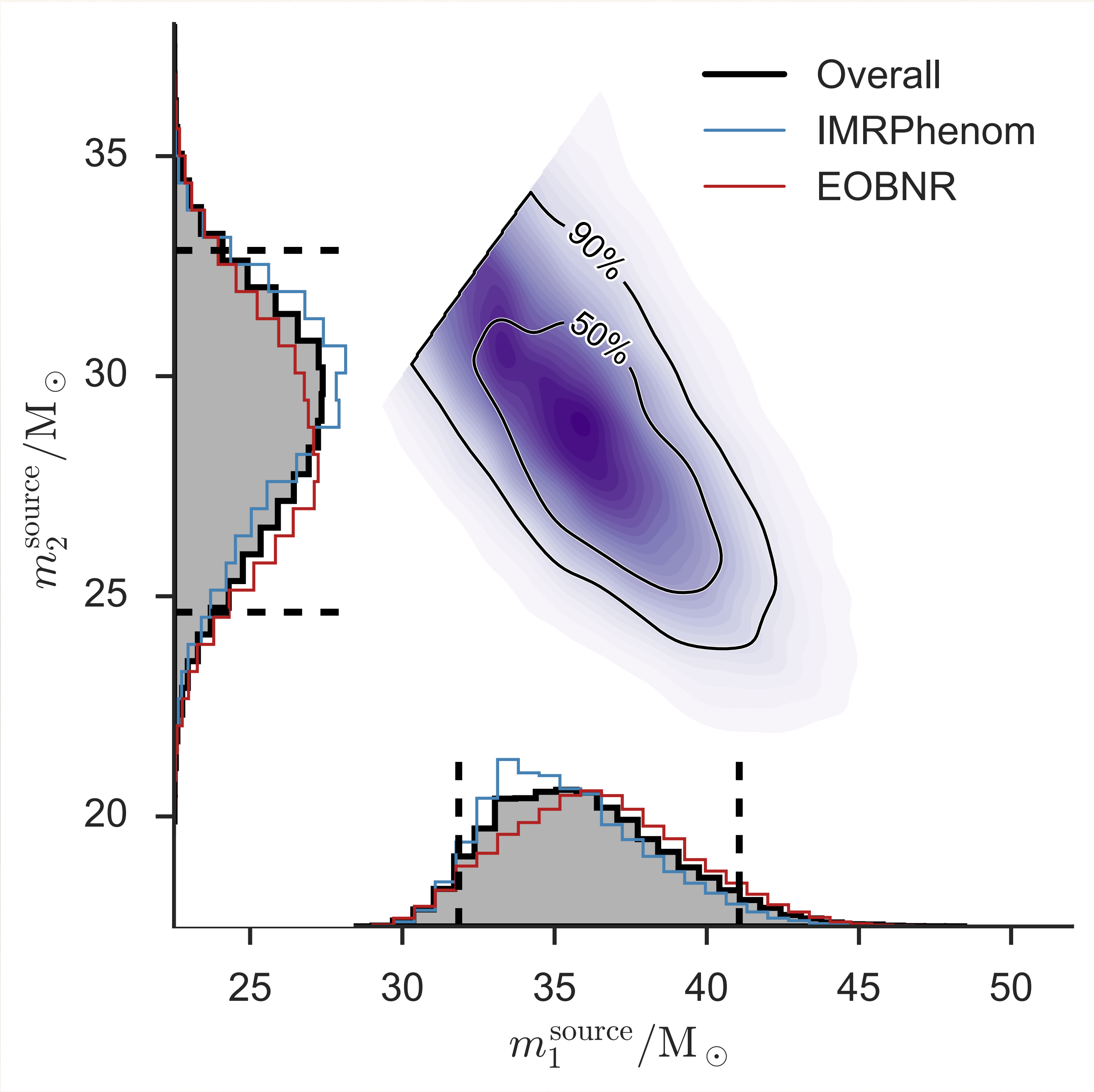
$$p(d|\vec{\theta}) \propto \exp\left[-\frac{1}{2}(d - h(\vec{\theta})|d - h(\vec{\theta}))\right] \qquad (a|b) = \int_{-\infty}^{\infty} \frac{\tilde{a}^*(f)\tilde{b}(f) + \tilde{a}(f)\tilde{b}^*(f)}{S_h(f)}\mathrm{d}f$$

- Inference typically uses *Markov Chain Monte Carlo* or other stochastic sampling methods to draw samples from the posterior distribution - needs **millions** of likelihood evaluations, which rely on constructing **expensive** waveform models.

Computational cost: GW150914

- The analysis of GW150914 used 50 million CPU hours (20,000 PCs running for 100 days). A significant fraction of that was PE.
- Lag between observation and publication of exceptional events mostly dominated by PE (re-)runs.

Primary black hole mass	$36^{+5}_{-4} M_\odot$
Secondary black hole mass	$29^{+4}_{-4} M_\odot$
Final black hole mass	$62^{+4}_{-4} M_\odot$
Final black hole spin	$0.67^{+0.05}_{-0.07}$
Luminosity distance	410^{+160}_{-180} Mpc
Source redshift z	$0.09^{+0.03}_{-0.04}$



Emulation of waveform models

- Faster waveform model evaluation facilitates faster PE with standard methods. Compensate for approximations by modelling errors with Gaussian process regression, representing the model error $\delta h(\vec{\theta})$ by a Gaussian distribution, trained with simulations

$$p(\delta h(\vec{\theta})) \propto \exp\left[\frac{(\delta h(t;\vec{\theta}) - \mu(t;\vec{\theta})|\delta h(t;\vec{\theta}) - \mu(t;\vec{\theta}))}{2\sigma^2(\vec{\theta})}\right]$$

$$\mu(\vec{\theta}) = [\mathbf{K}_*]_i \left[\mathbf{K}^{-1}\right]_{ij} \delta h(\vec{\theta}_j) \qquad \sigma^2(\vec{\theta}) = \mathbf{K}_{**} - [\mathbf{K}_*]_i \left[\mathbf{K}^{-1}\right]_{ij} [\mathbf{K}_*]_j$$

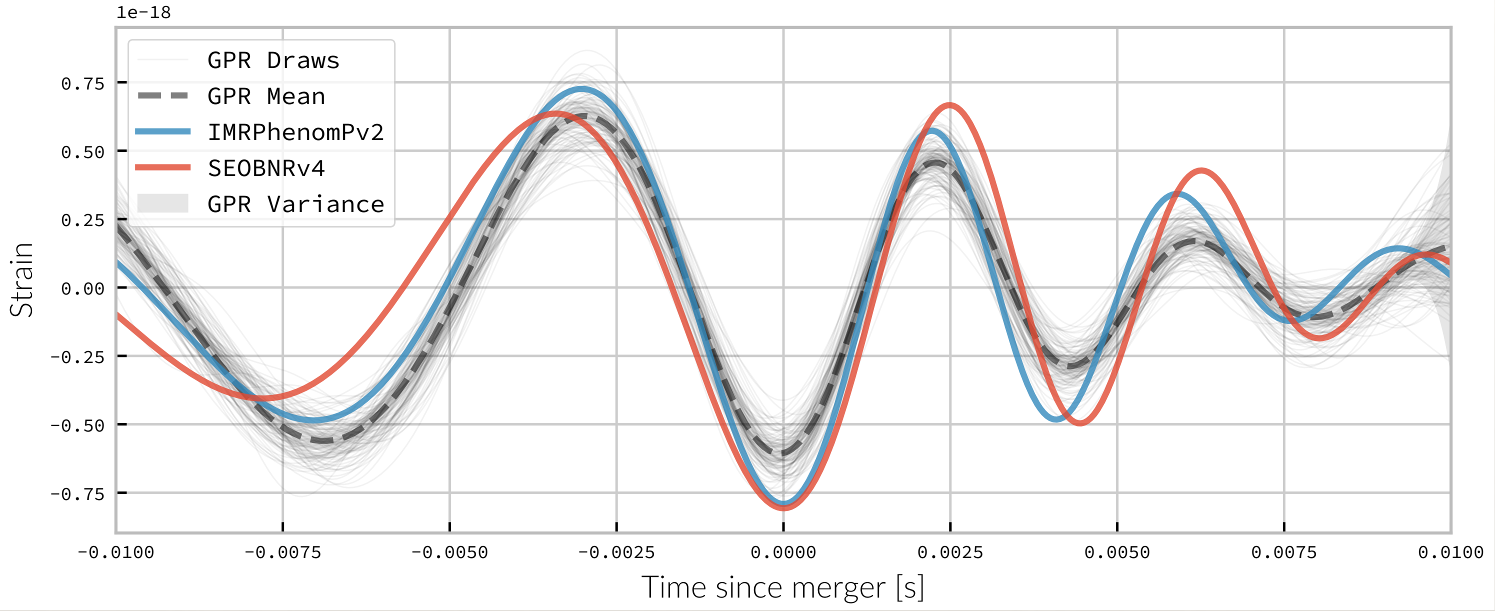
$$[\mathbf{K}]_{ij} = k(\vec{\theta}_i, \vec{\theta}_j) \qquad [\mathbf{K}_*]_i = k(\vec{\theta}, \vec{\theta}_i) \qquad \mathbf{K}_{**} = k(\vec{\theta}, \vec{\theta})$$

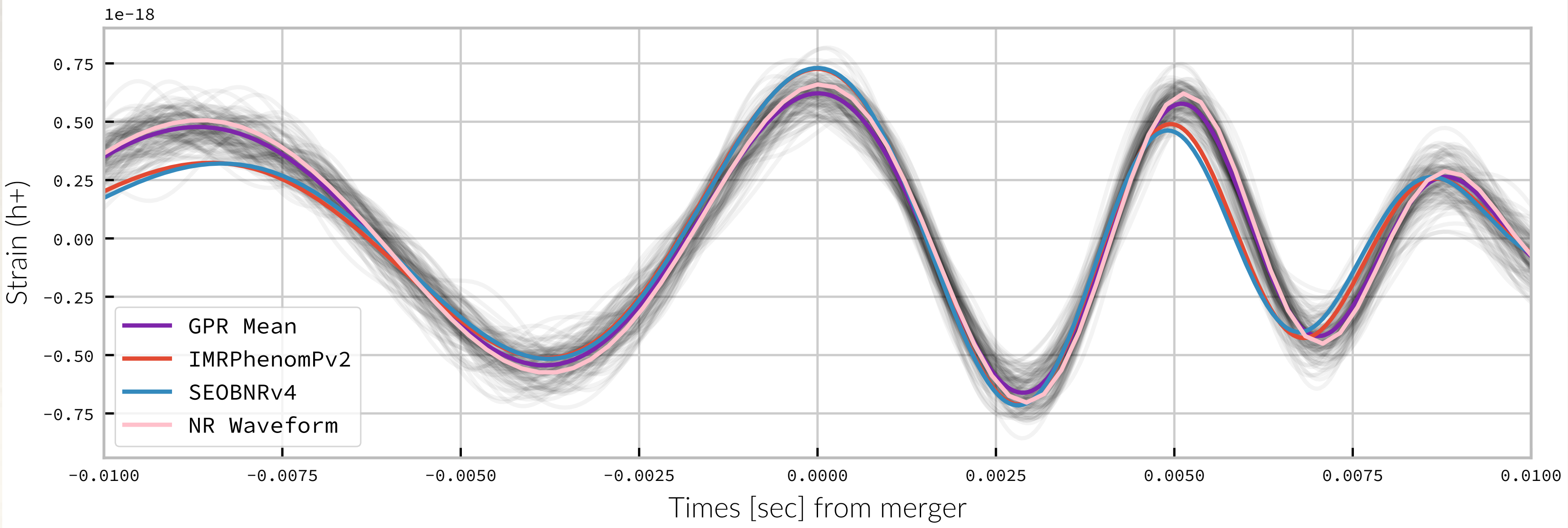
- The corresponding marginalised likelihood for the data is

$$\mathcal{L}_{\mathrm{GP}}(\vec{\theta}) \propto \frac{1}{1+\sigma^2(\vec{\theta})} \exp\left[-\frac{1}{2}\frac{\left(d - h_{\mathrm{AP}}(t;\vec{\theta}) - \mu(t;\vec{\theta})|d - h_{\mathrm{AP}}(t;\vec{\theta}) - \mu(t;\vec{\theta})\right)}{1+\sigma^2(\vec{\theta})}\right]$$

Emulation of waveform models

- GPR is useful for all types of emulation, including population inference. While it mitigates some waveform evaluation costs, its primary appeal is that it provides an error estimate that can be marginalised over (e.g., Williams et al. 2020).





Neural posterior estimation

- Stochastic sampling relies on being able to evaluate the likelihood, $p(d|\theta)$, which is done during sampling and requires a new waveform evaluation at each step.
- Alternative: *simulation based inference*. Construct a neural network that generates samples from a distribution, $q(\theta|d)$, that approximates the target distribution, in this case the parameter posterior distribution, $p(\theta|d)$. Train the neural network by minimising the average *cross-entropy* with the true distribution

$$L = \mathbb{E}_{p(d)}\mathbb{E}_{p(\theta|d)}\left[-\log q(\theta|d)\right] = \mathbb{E}_{p(\theta)}\mathbb{E}_{p(d|\theta)}\left[-\log q(\theta|d)\right]$$

- Compute loss by simulation

$$\textbf{Sample } \theta^{(i)} \sim p(\theta), \quad i = 1, \ldots, N$$

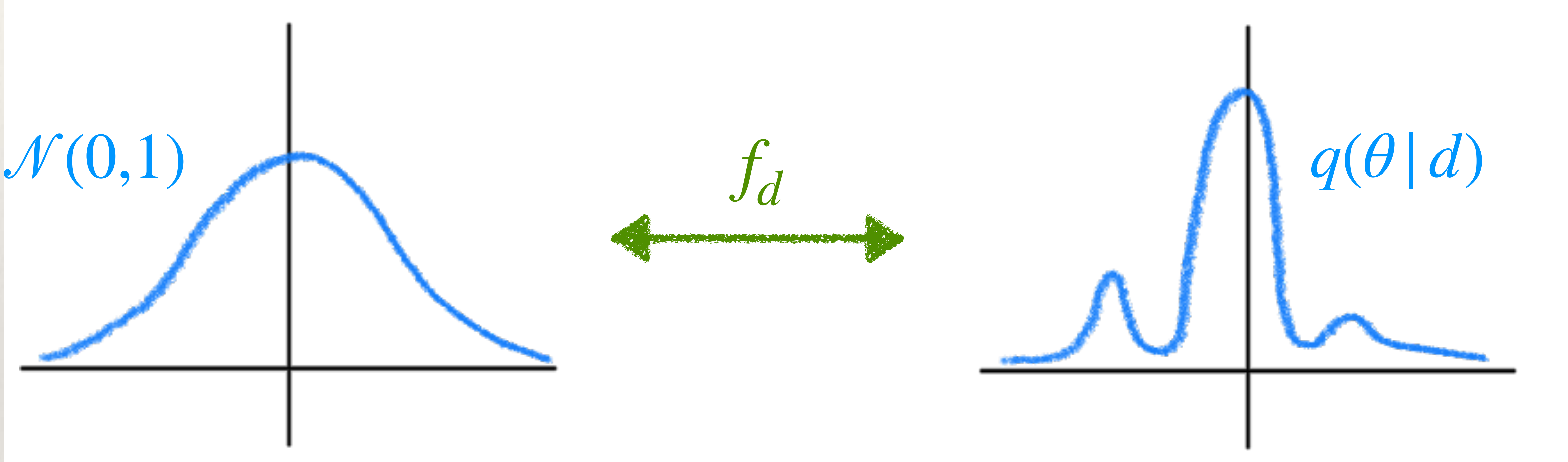
$$\textbf{Simulate } d^{(i)} \sim p(d|\theta^{(i)}); \quad d^{(i)} = h(\theta^{(i)}) + n^{(i)} \text{ with } n^{(i)} \sim p_{S_n}(n)$$

$$\textbf{Compute } L \approx \frac{1}{N}\sum_{i=1}^{N}\left[-\log q(\theta^{(i)}|d^{(i)})\right]$$

- **Advantages**: *likelihood-free*, *amortised* cost of waveform generation, *flexible*.

Normalizing flows

- A normalising flow maps represents a complex distribution as a mapping of a simple one.



- Construct target distribution using

$$q(\theta|d) = \mathcal{N}(0,1)^D \left(f_d^{-1}(\theta)\right) \left|\det J_{f_d}^{-1}\right|$$

- Want mapping to be invertible and have a simple Jacobian determinant. Can represent a normalising flow with these properties using a neural network.

Normalizing flows

- Build normalising flow from a sequence of *coupling transforms*

$$c_{d,i}(u) = \begin{cases} u_i & \text{if } i \leq D/2 \\ c(u_i; u_{i:\frac{D}{2}}, d) & \text{if } i > D/2 \end{cases}$$

- The coupling transforms must be differentiable and invertible.
- We use *spline flows* (Durkan et al. 2019), based on rational quadratic spline interpolation between a set of knots.
- A sequence of transforms can represent very complicated functions.

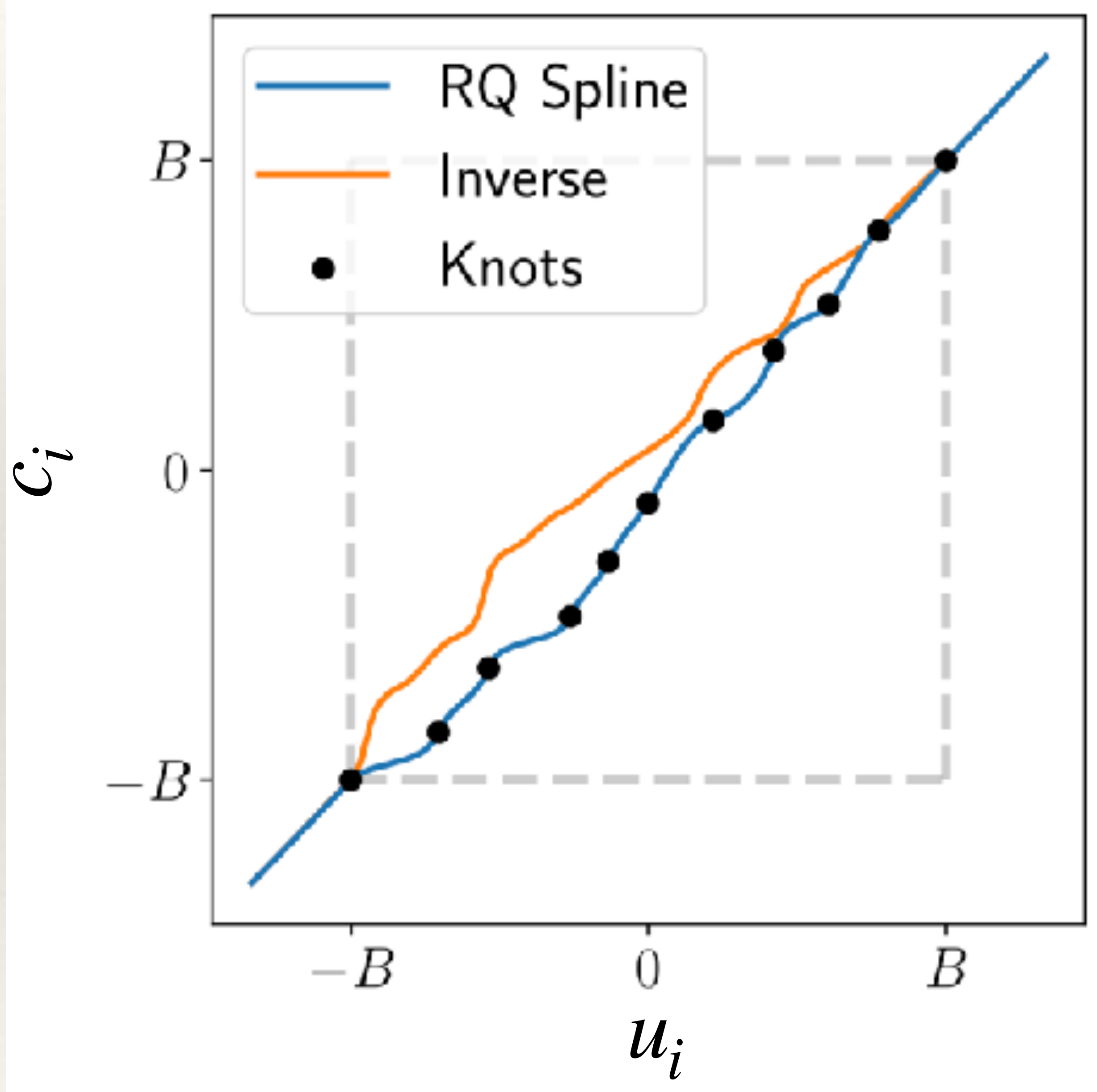


Figure: Durkan *et al* (2019)

NPE refinements: embedding network

- The existence of reduced bases shows that waveform bases can be compressed. Could impose this by hand, but more robust to learn this using an *embedding network*.

$N_{\mathrm{in}} = n_{\mathrm{det}}$ x 24099

(8033 frequency bins) x (Re + Im + PSD)
8s segment, sampled at 1024Hz,
f_{low} = 20Hz

Linear projection

Seed with principal components of *noise-free* waveforms

inductive bias to recognise waveforms

n_{det} x 400

Fully connected residual network

48 hidden layers

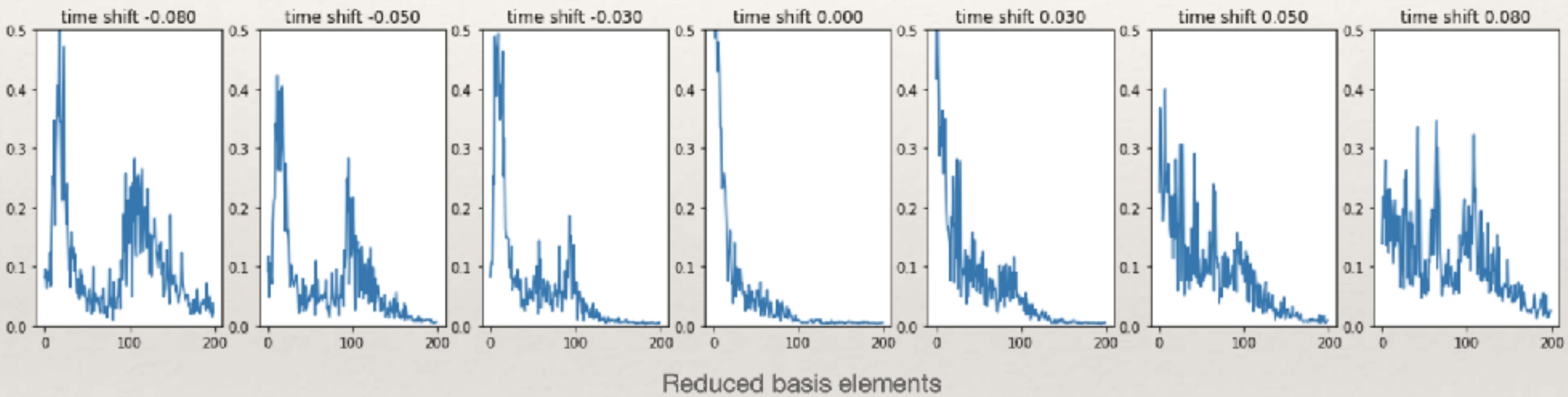
non-linear compression

$N_{\mathrm{out}} = 128$

to flow

NPE refinements: group equivariant NPE

- Representing the time of coalescence, t_I, requires many reduced basis elements. Uses up a lot of training resources and freedom within the network.



- A change in time of coalescence in a single detector corresponds to a (trivial) transformation of the data and template. If the time shift is known, the waveform can be aligned and the learning process significantly simplified.

- Don't know this *a priori* and not an exact symmetry for a detector network.

NPE refinements: group equivariant NPE

- Introduce a blurred estimate of t_I, $\hat{t}_I$, into the parameter space.
- In training and inference, follow a Gibbs sampling procedure

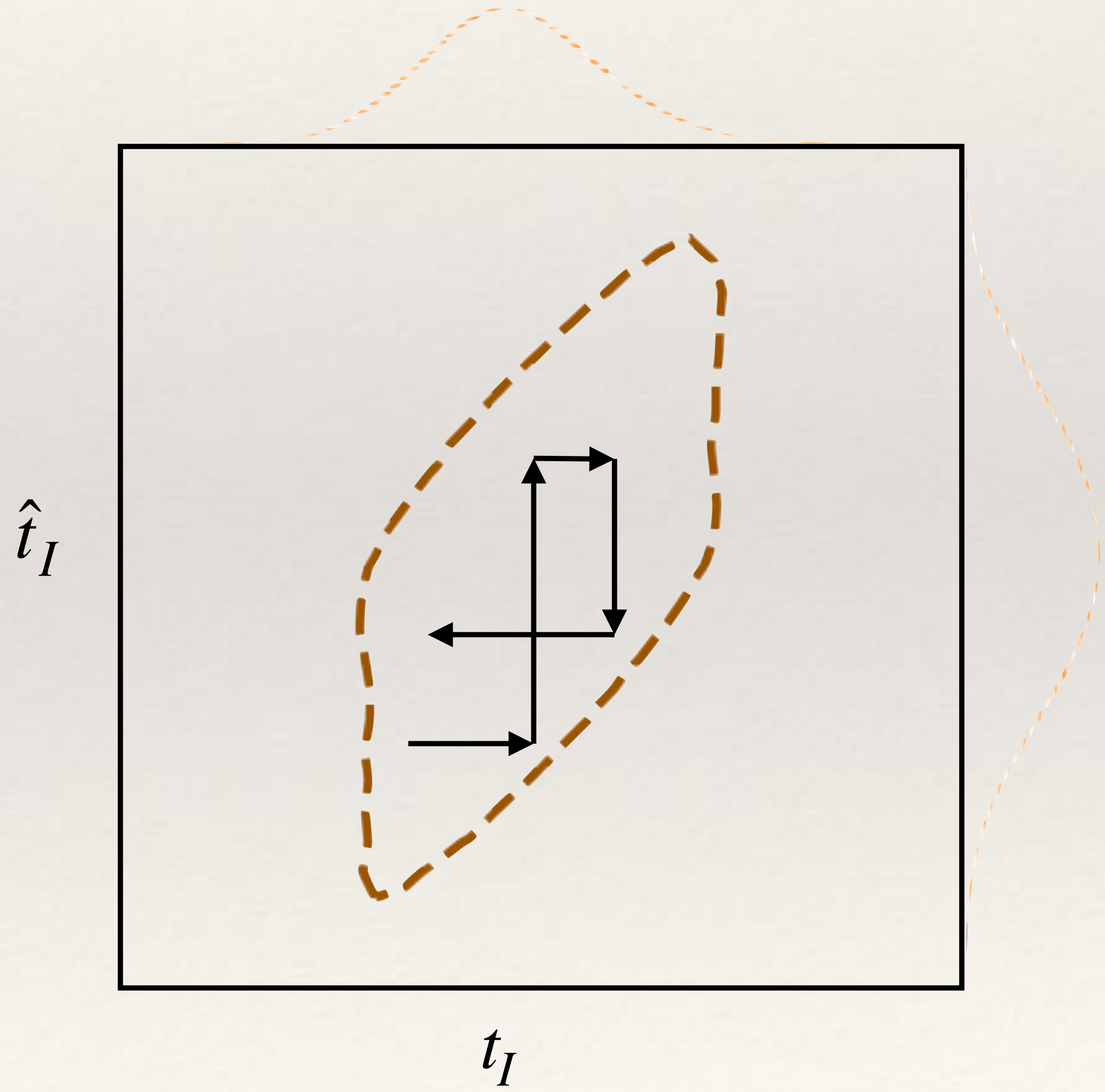
  1. Align data based on $\hat{t}_I$

  $$\theta \sim q(\theta | T_{-\hat{t}_I}(d), \hat{t}_I)$$

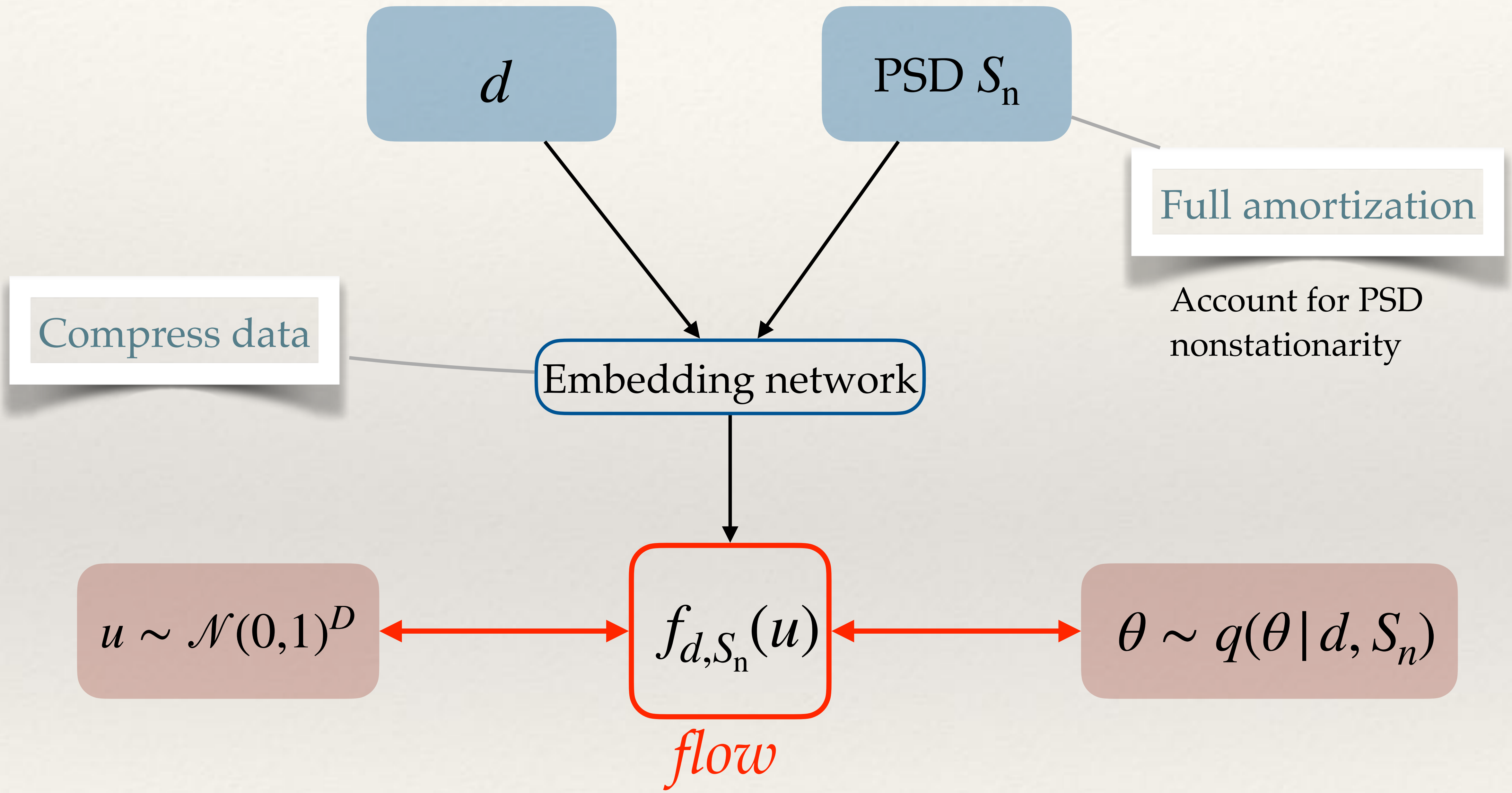
  2. Sample $\hat{t}_I$ from a fixed kernel

  $$\hat{t}_I \sim p(\hat{t}_I | t_I)$$

- Converges in O(10) iterations.
- GNPE exploits (near-) symmetries to simplify the learning task.



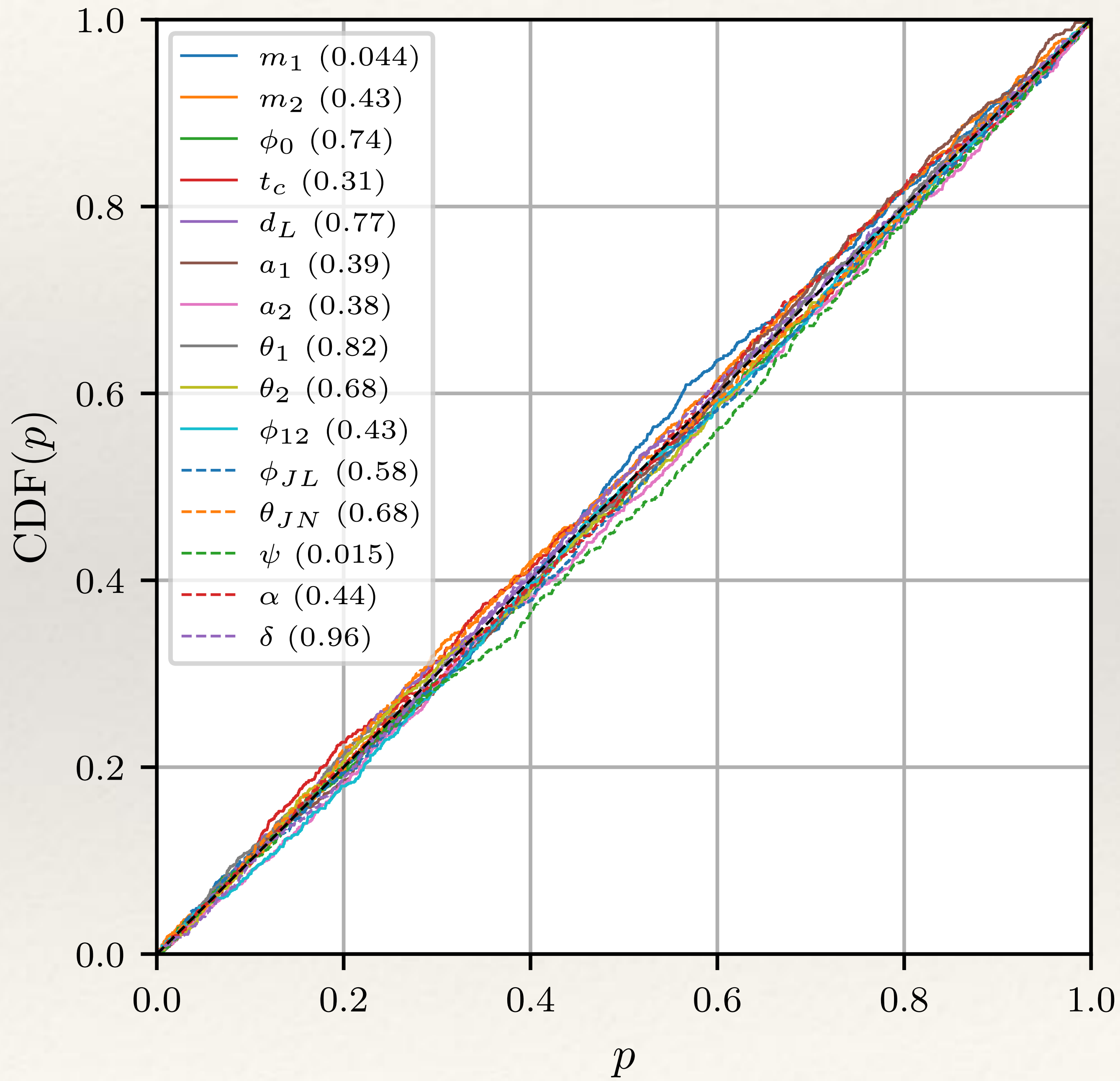
NPE network



Big neural networks: $\approx$ 350 layers and 150 million parameters

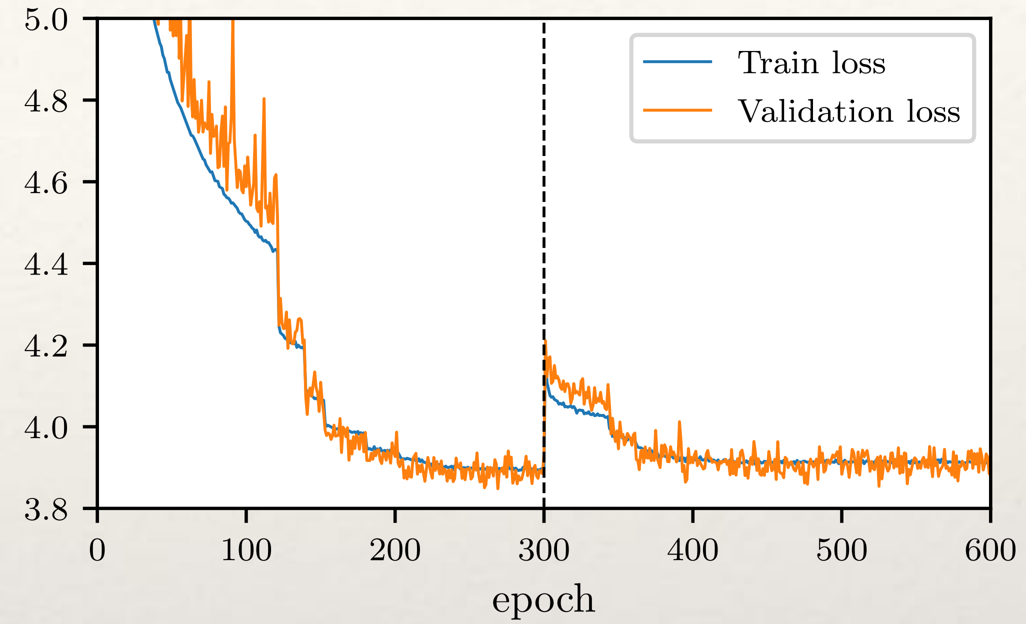
NPE validation

- Check **internal** or **within-distribution** consistency of network by generating a p-p plot.
- Check **external** or **out-of-distribution** consistency by comparing to posterior distributions computed for real observations using standard stochastic samplers.



NPE validation: GWTC-1 BBHs

- Used 5×10^6 waveforms for training
    - IMRPhenomPv2
    - $T = 8$ s, $f_{\min} = 20$ Hz, $f_{\max} = 1024$ Hz
    - 15D parameter space
    - $m_1, m_2 \in [10{,}80]\ M_\odot$

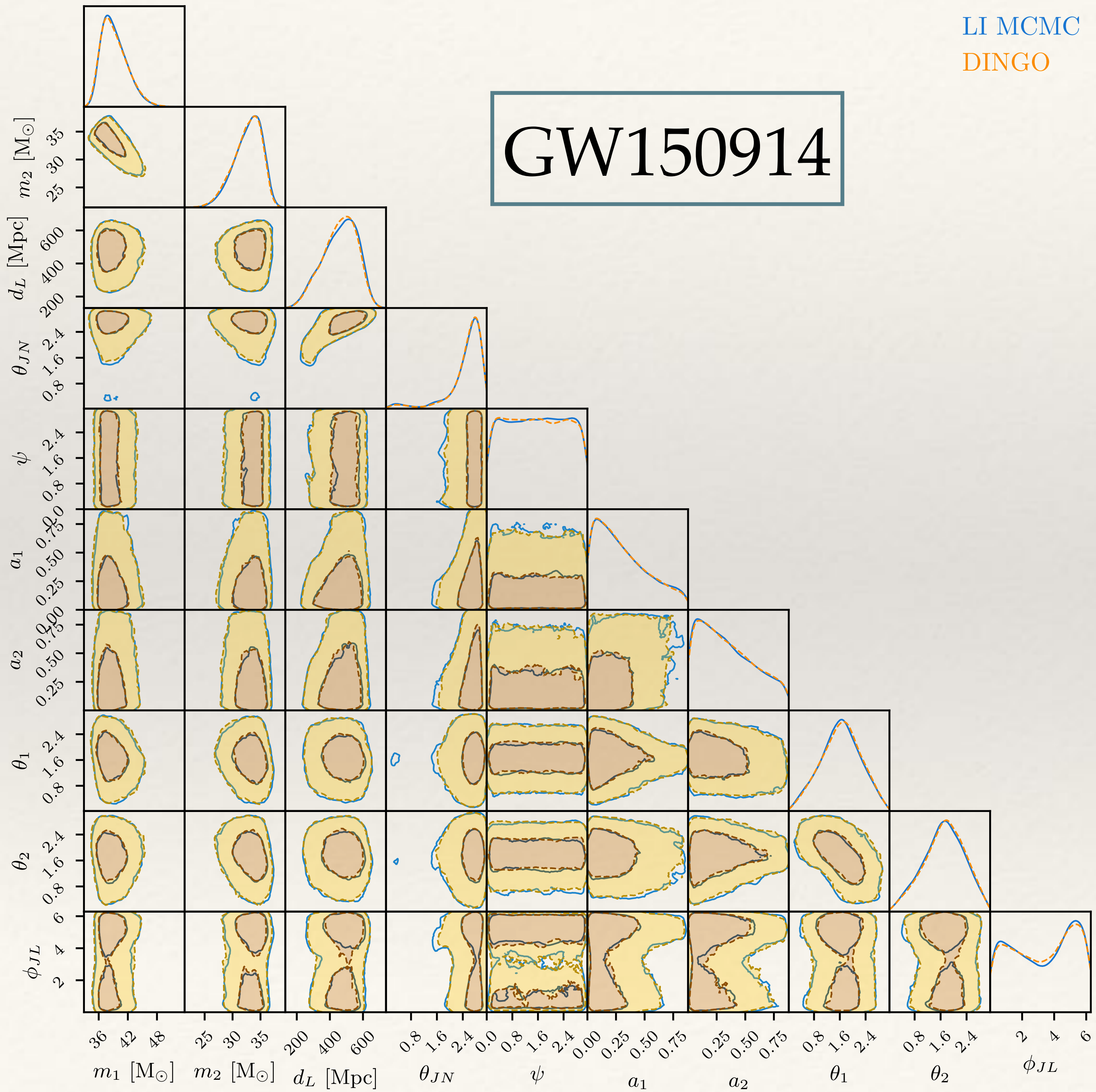


- + stationary Gaussian noise realisations consistent with given PSD

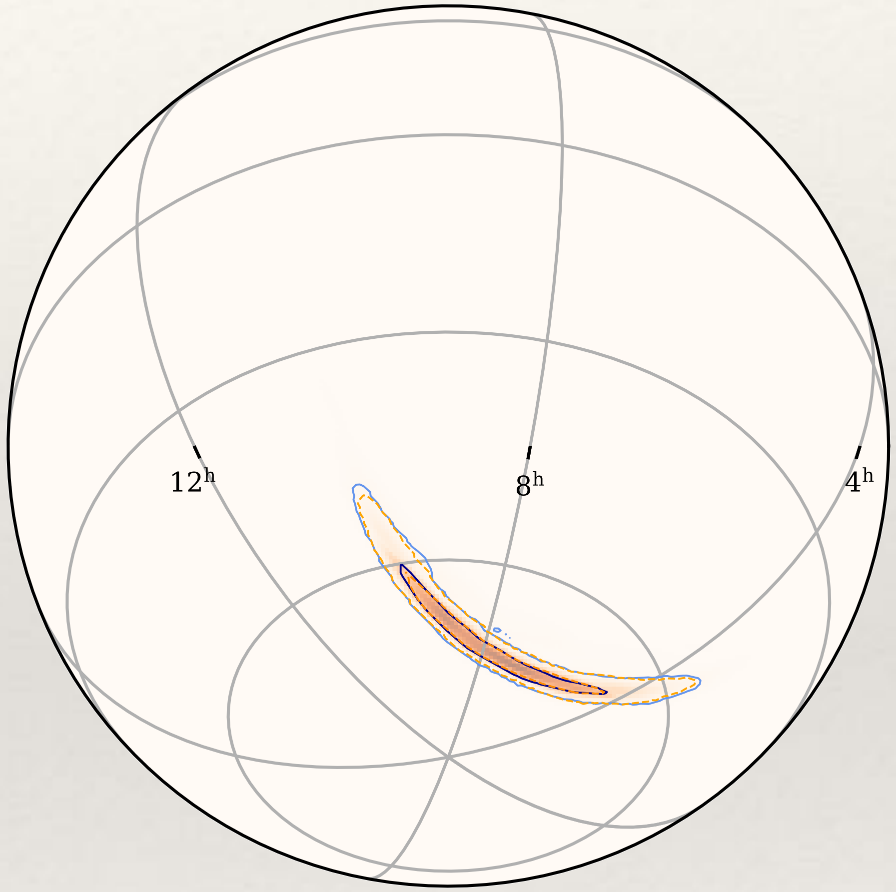
- Train several neural networks based on different noise level / number of detectors / distance range:

Observing run	Detectors	Distance range [Mpc]
O1	HL	[100, 2000]
O2	HL	[100, 2000]
		[100, 6000]
	HLV	[100, 1000]

NPE validation: GWTC-1 BBHs



GW150914



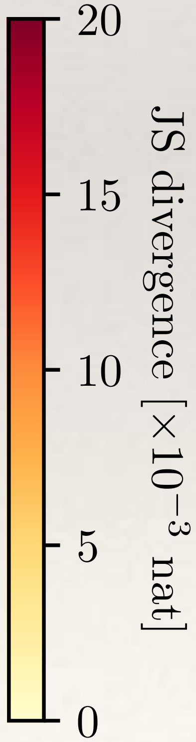
50,000 samples in ~ 20 s

NPE validation: GWTC-1 BBHs

- Compare NPE posteriors to “standard” posteriors generated by *LALInference* and *Bilby*. Use JS divergence as a metric for comparison.

- JS divergence less than 2 nats considered *indistinguishable*.

	m_1	m_2	ϕ_0	d_L	a_1	a_2	θ_1	θ_2	ϕ_{12}	ϕ_{JL}	θ_{JN}	ψ	α	δ
GW150914	0.8	1.1	0.2	0.8	0.2	0.3	0.5	0.5	0.1	0.3	0.8	0.2	0.7	1.4
GW151012	2.7	1.6	0.1	0.9	0.4	0.2	0.5	0.5	0.1	0.1	0.6	0.1	1.4	0.5
GW170104	6.4	2.6	0.2	0.4	0.7	0.1	0.7	0.4	0.1	0.1	0.3	0.3	0.8	0.6
GW170729	0.9	1.5	0.4	6.3	0.2	0.2	1.0	0.8	0.2	0.3	3.4	0.3	1.2	1.2
GW170809	0.5	0.8	0.1	0.5	0.2	0.1	0.4	0.4	0.1	0.5	1.4	0.2	2.2	5.5
GW170814	1.2	1.3	0.2	1.5	0.2	0.2	0.4	0.3	0.2	1.4	1.4	1.2	2.5	2.0
GW170818	1.6	1.3	0.2	1.1	1.0	0.2	1.9	0.5	0.1	2.4	1.8	0.4	3.8	2.4
GW170823	0.5	0.6	0.1	0.9	0.2	0.2	0.4	0.2	0.2	0.2	0.5	0.2	0.4	0.4



Summary

- Gravitational wave science relies on obtaining **parameter posterior distributions** for all observed sources. Multi-messenger applications require **rapid estimation** of sky position, and perhaps other parameters.

- Current PE codes are computationally intensive—need new methods that are **fast**, **robust** and **accurate**.

- **Neural posterior estimation** is a new, machine learning approach that now has comparable performance to standard methods in a fraction of the time. Training cost is amortised, allowing **near real-time analysis** of new observations.

- **Group equivariant NPE** can be used to simplify training of a neural network by exploiting near symmetries.

- **Future detectors pose many developmental challenges**: long waveforms, non-stationary noise, new sources, overlapping sources, population inference…. NPE could provide a viable route to solving (some of) these problems, but further developments are required.