Likelihood-free gravitational-wave parameter estimation with neural networks

Stephen R. Green
Albert Einstein Institute Potsdam

based on arXiv:2002.07656 with C. Simpson and J. Gair

Gravity Seminar
University of Southampton
February 27, 2020

Outline

1. Introduction to Bayesian inference for compact binaries

2. Likelihood-free inference with neural networks

    (a) Basic approach

    (b) Normalizing flows

    (c) Variational autoencoders

3. Results

Introduction to parameter estimation

- Bayesian inference for compact binaries:

  Sample posterior distribution for system parameters θ (masses, spins, sky position, etc.) given detector strain data s.

likelihood prior

$$p(\theta | s) = \frac{p(s | \theta) p(\theta)}{p(s)}$$

evidence (normalizing factor)

- Once likelihood and prior are defined, right hand side can be evaluated (up to normalization).

Introduction to parameter estimation

- *Likelihood based on assumption that if the gravitational-wave signal were subtracted from s, then what remains must be noise.*

- Noise n assumed to follow stationary Gaussian distribution, i.e.,

$$n \sim p(n) \propto \exp\left(-\frac{1}{2}\left(n \,|\, n\right)\right)$$

where the noise-weighted inner product is

$$\left(a \,|\, b\right) = 2\int_0^\infty \mathrm{d}f \, \frac{\hat{a}(f)\hat{b}(f)^* + \hat{a}(f)^*\hat{b}(f)}{S_n(f)}$$

detector noise power spectral density (PSD)

- Summed over detectors, this gives the likelihood,

$$p(s \,|\, \theta) \propto \exp\left(-\frac{1}{2}\sum_I \left(s_I - h_I(\theta) \,|\, s_I - h_I(\theta)\right)\right)$$

Introduction to parameter estimation

- Prior $p(\theta)$ based on beliefs about system before looking at data,

  e.g., uniform in m_1, m_2 over some range,
  uniform in spatial volume,
  etc.

- With prior and likelihood defined, the posterior can be evaluated up to normalization.

- Method such as Markov chain Monte Carlo (MCMC) is used to obtain posterior samples.

  Move around parameter space, and compare strain data s against waveform model $h(\theta)$.

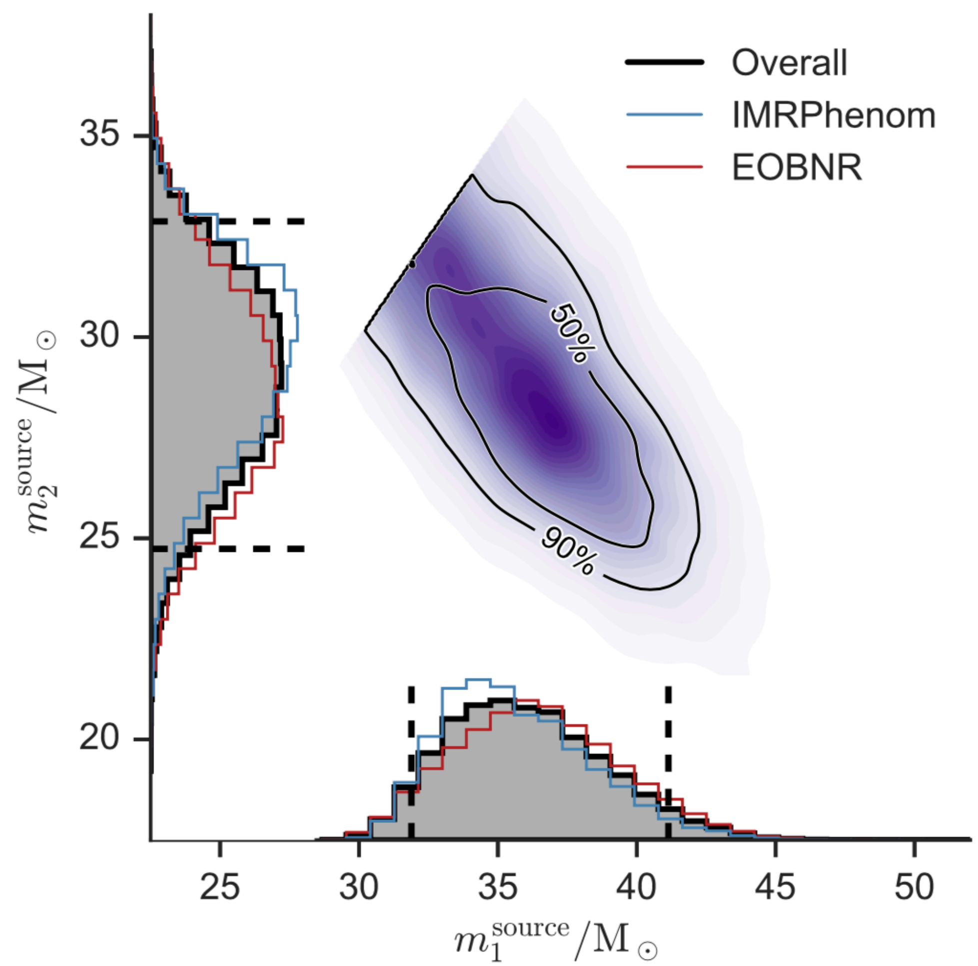


Image: Abbott et al (2016)

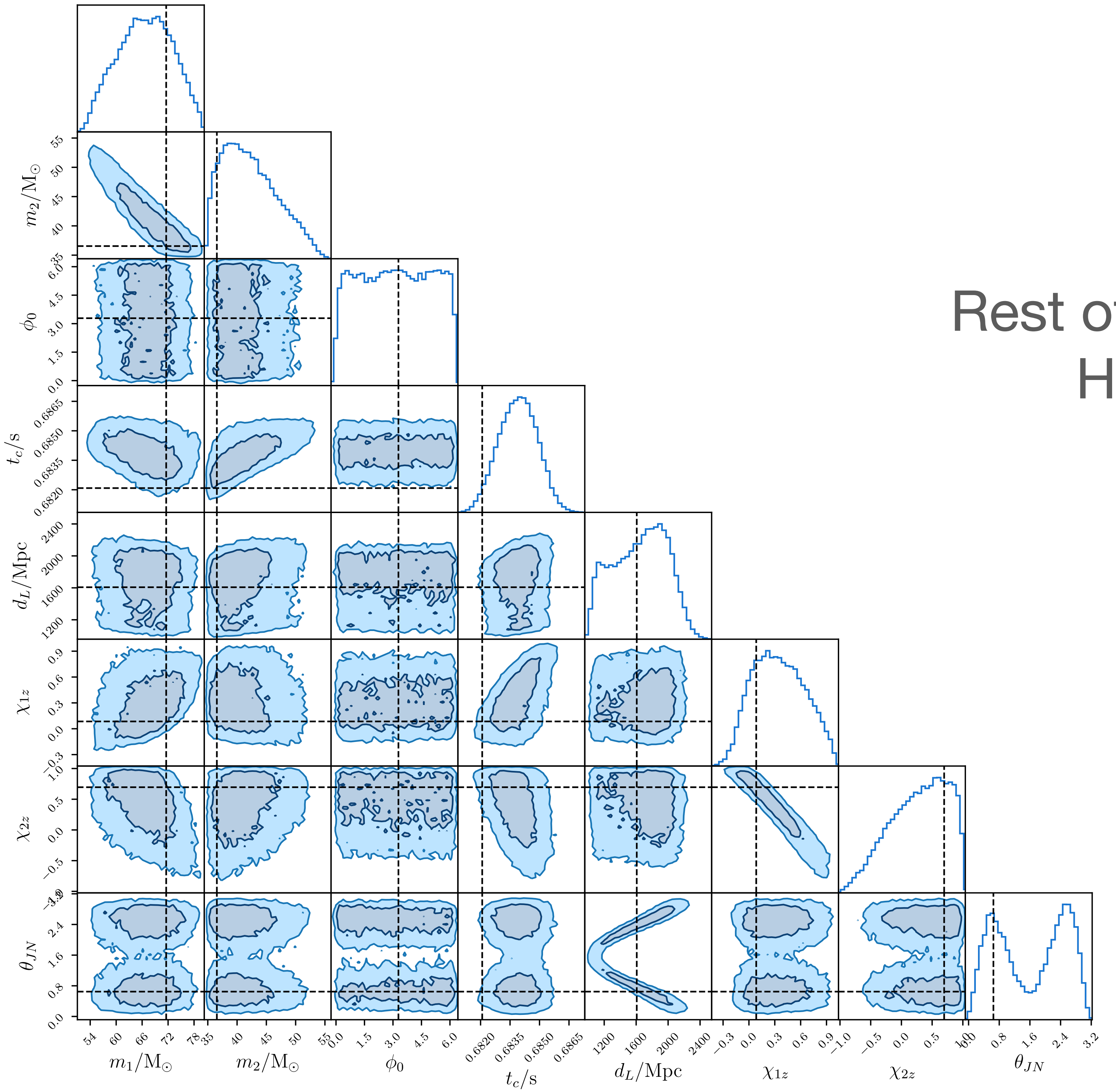
Need for new methods

- **Standard method expensive:**
    - Many likelihood evaluations required for each independent sample
    - Likelihood evaluation slow, requires a waveform to be generated
        - Various waveform models (EOBNR, Phenom, …) created as faster alternatives to numerical relativity; reduced-order surrogate models for even faster evaluation.
    - Days to months for parameter estimation of a single event, depending on type of event and waveform model.

Goal of this work:

Develop deep learning methods to do parameter estimation much faster. Model the posterior distribution $p(\theta \mid s)$ with a neural network.

Main result: very fast posterior sampling



Rest of this talk:

How did we do this?

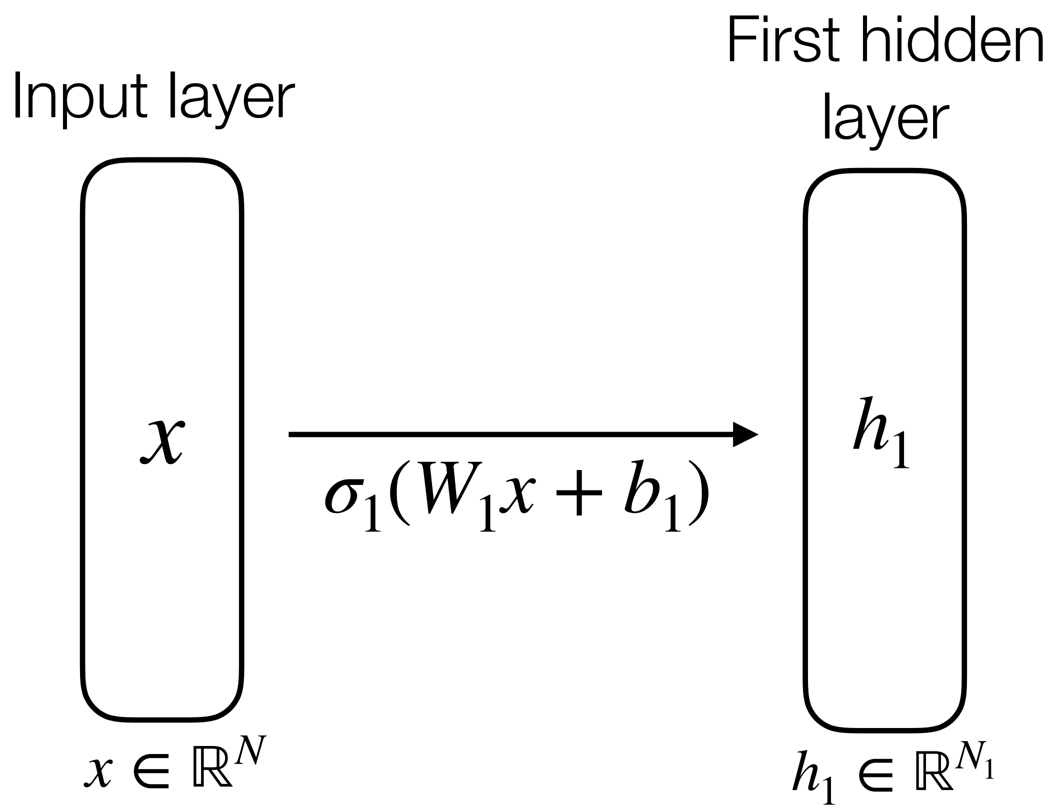
Two key ideas

1. A conditional probability distribution can be described by a neural network.

2. The network can be trained to model a gravitational wave posterior distribution without ever evaluating a likelihood. Instead, it only requires samples (θ, s) from the data generating process.

Introduction to neural networks

- Nonlinear functions constructed as composition of mappings:



Consists of:

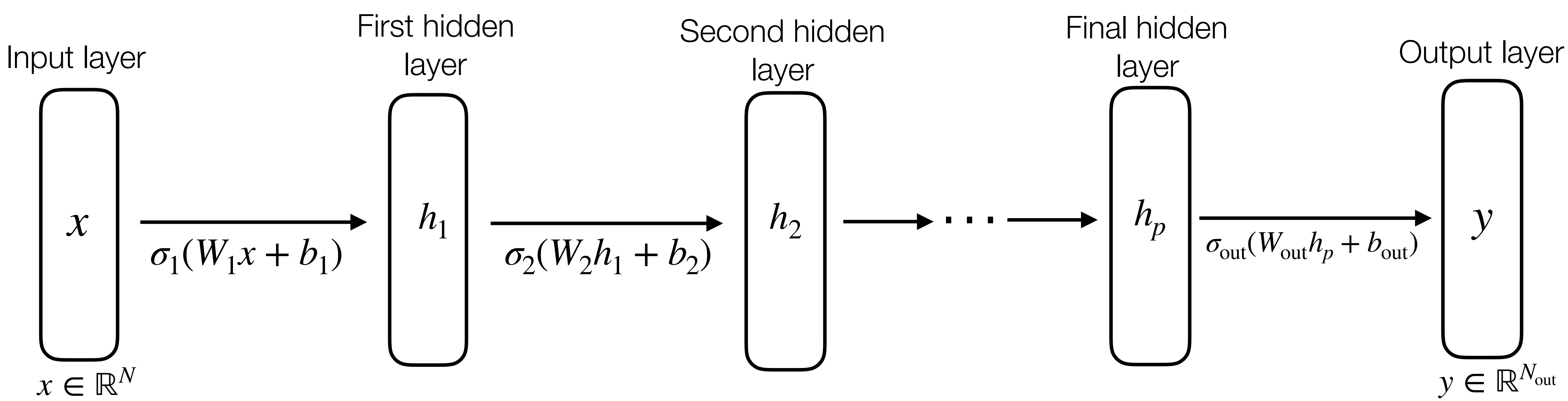
1. Linear transformation

$$W_1 x + b_1$$

2. Simple element-wise nonlinear mapping.

E.g., $\sigma_1(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$

Introduction to neural networks



- Training/test data consist of *(x, y)* pairs.
- Train network by tuning the weights *W* and biases *b* to minimize loss function $L(y, y_{\text{out}})$
- Stochastic gradient descent combined with chain rule (“backpropagation”) to adjust weights and biases.

Neural networks as probability distributions

- *Since conditional probability distributions can be parametrized by functions, and neural networks are functions, conditional probability distributions can be described by neural networks.*

  E.g., multivariate normal distribution

$$
\begin{aligned}
p(x|y) &= \mathcal{N}(\mu(y), \Sigma(y))(x) \\
&= \frac{1}{\sqrt{(2\pi)^n |\det \Sigma(y)|}} \exp\left(-\frac{1}{2}\sum_{ij=1}^{n}(x_i - \mu_i(y))\Sigma_{ij}^{-1}(y)(x_j - \mu_j(y))\right)
\end{aligned}
$$

  where $\mu(y), \Sigma(y) = \mathrm{NN}(y)$.

- For this example, it is trivial to draw samples and evaluate the density.

- More complex distributions may also be described by neural networks (later in talk).

Likelihood-free inference with neural networks

[First applied to GW by Chua and Vallisneri (2020), Gabbard et al (2019)]

- Goal is to train network to model true posterior, as given by prior and likelihood that we specify, i.e.,

$$p(\theta \mid s) \to p_{\text{true}}(\theta \mid s)$$

- Minimize expectation value (over s) of cross-entropy between the distributions

$$L = -\int \mathrm{d}s\, p_{\text{true}}(s) \int \mathrm{d}\theta\, p_{\text{true}}(\theta \mid s)\, \log p(\theta \mid s)$$

Intractable with knowing posterior for each s!

- Bayes' theorem $\implies$ $p_{\text{true}}(s)\, p_{\text{true}}(\theta \mid s) = p_{\text{true}}(\theta)\, p_{\text{true}}(s \mid \theta)$

$$\boxed{\therefore\; L = -\int \mathrm{d}\theta\, p_{\text{true}}(\theta) \int \mathrm{d}s\, p_{\text{true}}(s \mid \theta)\, \log p(\theta \mid s)}$$

Only requires samples from likelihood,
not the posterior!

Likelihood-free inference with neural networks

- Loss function

$$
\begin{aligned}
L &= -\int \mathrm{d}\theta \, p_{\mathrm{true}}(\theta) \int \mathrm{d}s \, p_{\mathrm{true}}(s|\theta) \, \log p(\theta|s) \\
&\approx -\frac{1}{N} \sum_{i=1}^{N} \log p(\theta^{(i)}|s^{(i)}), \qquad \text{where } \theta^{(i)} \sim p_{\mathrm{true}}(\theta), \;\; s^{(i)} \sim p_{\mathrm{true}}(s|\theta^{(i)})
\end{aligned}
$$

Estimate on minibatch of size N

Easy to evaluate from neural network

Sample strain data from generative process (likelihood)

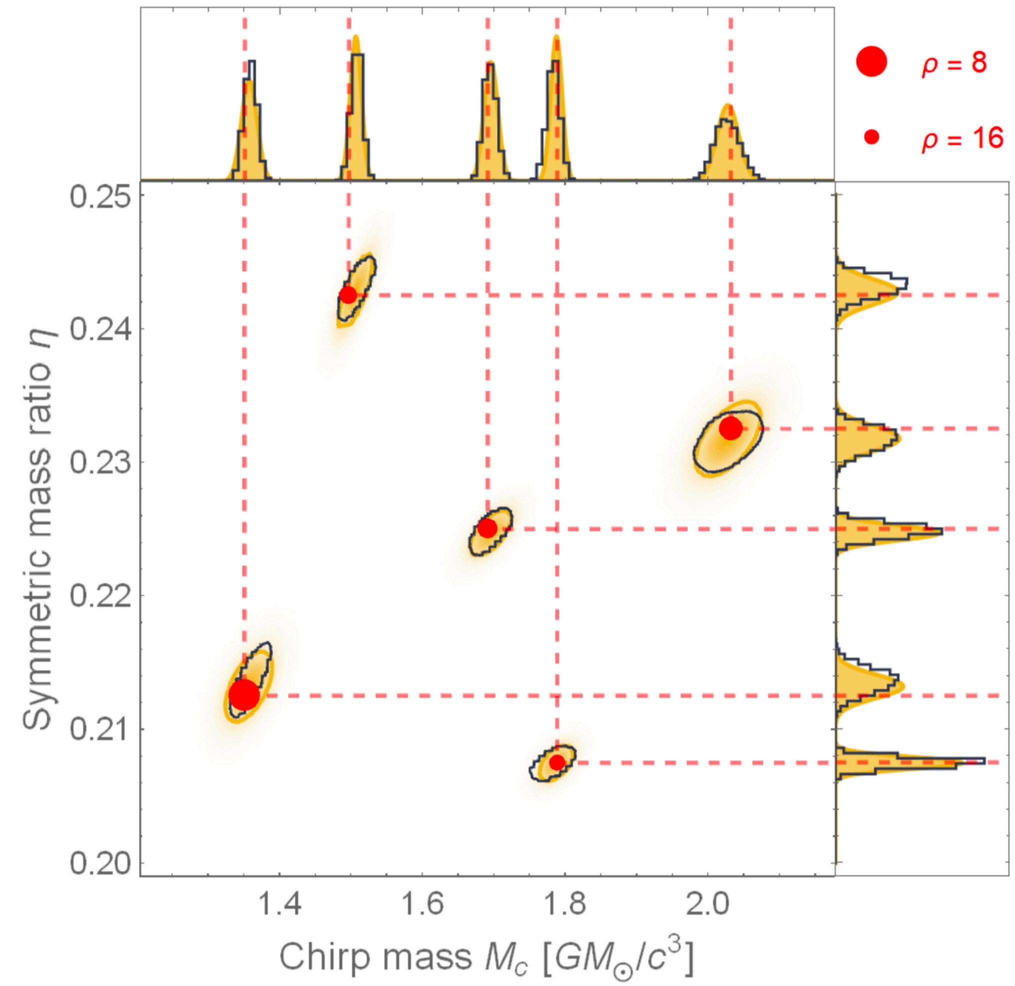
Sample parameters from prior

- Choose network parameters that minimize L: compute gradient of L with respect to network parameters (weights and biases) and use stochastic gradient descent.

- Never evaluate a likelihood and no need for posterior samples!

Gravitational-wave parameter estimation

- Chua and Vallisneri (2019) applied this method (with a Gaussian posterior model) to gravitational waves:



- *A Gaussian may be adequate for very high signal-to-noise, but more generally distributions can have higher moments and multimodality.*

Normalizing flows

Rezende and Mohamed (2015)

- **Our approach** to make gravitational-wave posterior more flexible: use a **normalizing flow**.

- Change of variables rule for probability distributions: if $\pi(u)$ is a probability distribution, and $f : u \mapsto x$ is a mapping on the sample space, then in the new coordinates, the distribution is

$$\boxed{p(x) = \pi(f^{-1}(x)) \left| \det \frac{\partial(f_1^{-1}, \ldots, f_n^{-1})}{\partial(x_1, \ldots, x_n)} \right|}$$

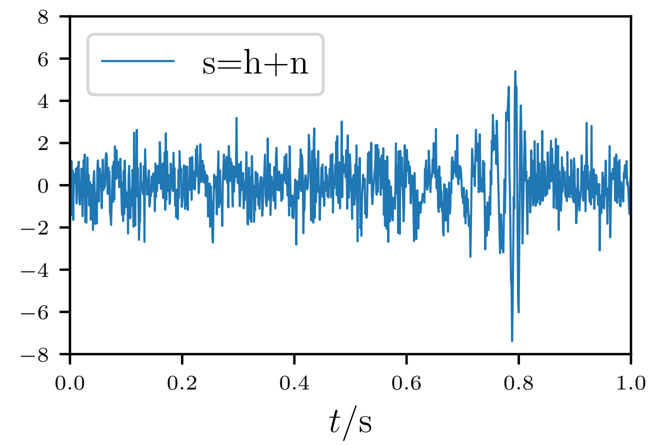
- A normalizing flow is an invertible mapping f with simple Jacobian determinant.

- *If $\pi(u)$ can be easily sampled and its density evaluated, and f is a normalizing flow, then the same holds for $p(x)$.*

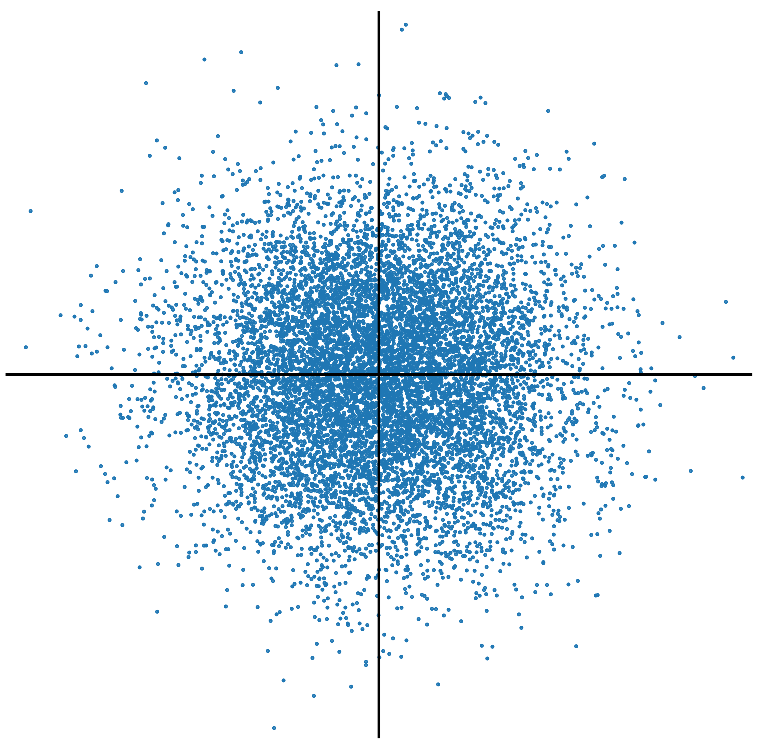
  Typically, take $\pi(u)$ to be a simple base distribution, e.g., multivariate standard normal.

Normalizing flows for gravitational waves

- To model a gravitational-wave posterior, take $x \rightarrow \theta$, and condition the flow f on strain data s.

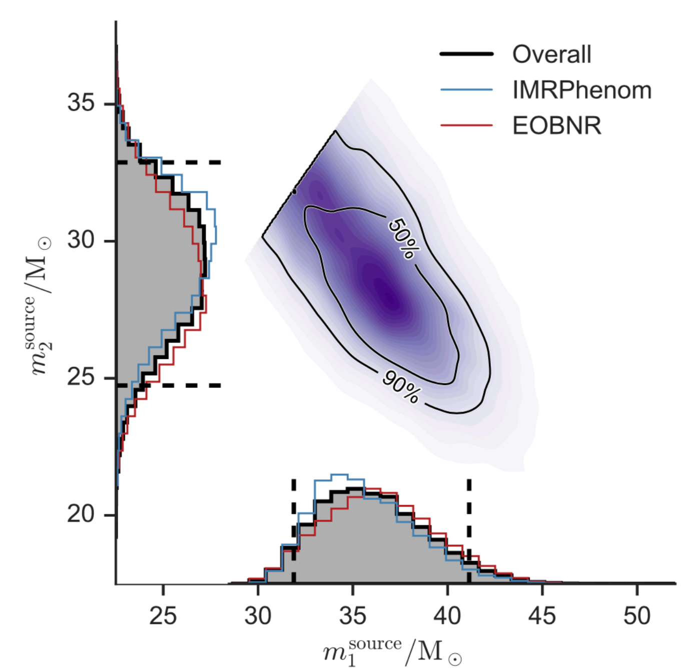


$u \sim \mathcal{N}(0,1)^n$



$\theta = f(u, s)$

$$\theta \sim p(\theta|s) = \mathcal{N}(0,1)^n(f^{-1}(\theta))|\det J_f^{-1}|$$



(hopefully)

Masked autoregressive flow

Papamakarios et al (2017)

- By the product rule, an arbitrary probability distribution $p(x)$ may be decomposed as

$$p(x) = \prod_{i=1}^{n} p(x_i | x_{1:i-1})$$

- Define an autoregressive model by restricting the form of each factor,

$$p(x_i | x_{1:i-1}) = \mathcal{N}(\mu_i(x_{1:i-1}), \exp(2\alpha_i(x_{1:i-1})))$$

i.e., if $u \sim \mathcal{N}(0,1)^n$, and we set $x_i = \mu_i(x_{1:i-1}) + u_i \exp \alpha_i(x_{1:i-1})$, then $x \sim p(x)$.

- The mapping $f : u \mapsto x$ defines a normalizing flow.

Masked autoregressive flow

Papamakarios et al (2017)

- f satisfies properties of a normalizing flow:

1. $f : u \mapsto x$ $$x_i = \mu_i(x_{1:i-1}) + u_i \exp \alpha_i(x_{1:i-1})$$

Forward map recursive

2. $f^{-1} : x \mapsto u$ $$u_i = \left[x_i - \mu_i(x_{1:i-1})\right] \exp\left(-\alpha_i(x_{1:i-1})\right)$$

Inverse map nonrecursive

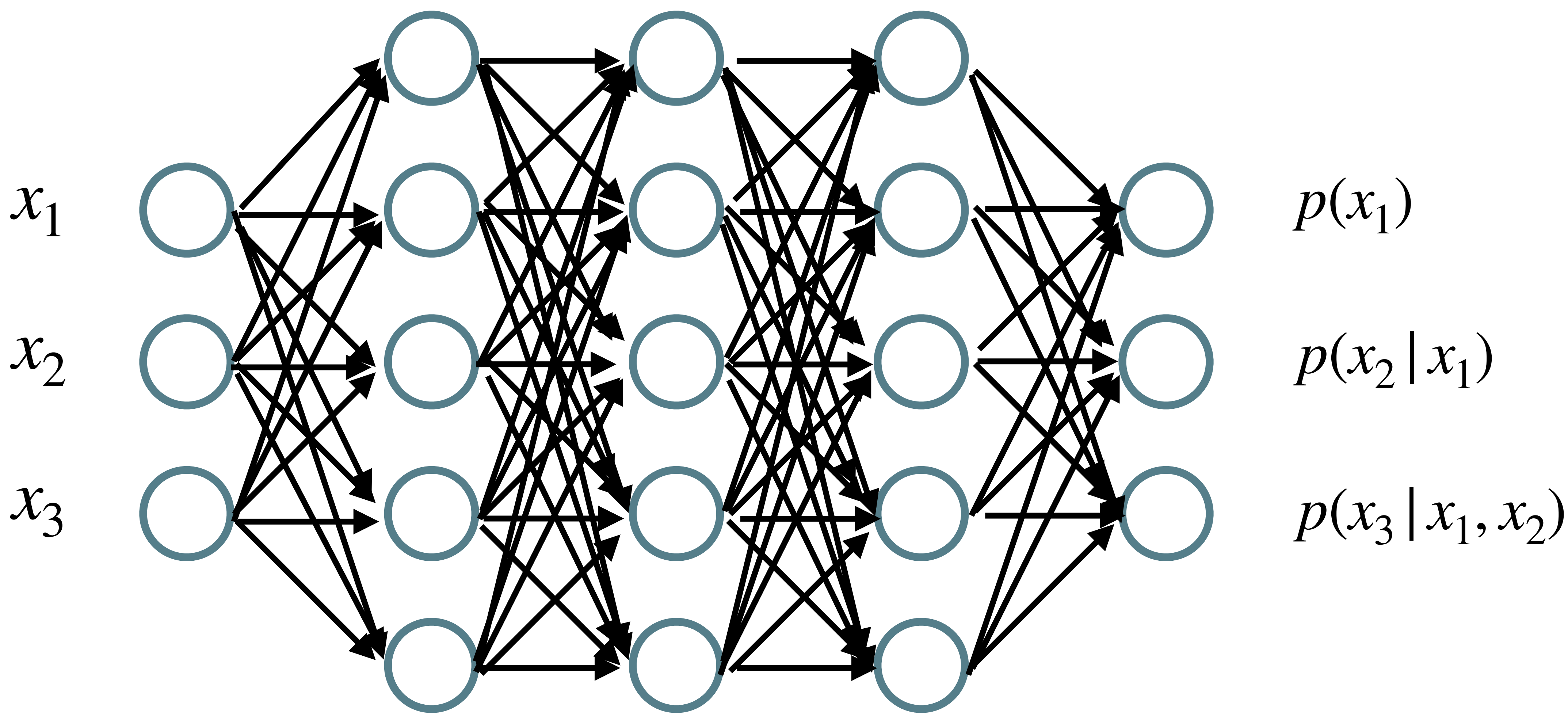
3. $$\left| \det \frac{\partial(f_1^{-1}, \ldots, f_n^{-1})}{\partial(x_1, \ldots, x_n)} \right| = \exp\left(-\sum_{i=1}^{n} \alpha_i(x_{1:i-1})\right)$$

Simple Jacobian determinant

Masked autoregressive flow

Papamakarios et al (2017)

- Can be implemented with a neural network by masking certain connections that violate autoregressive property [MADE network, Germain et al (2015)]

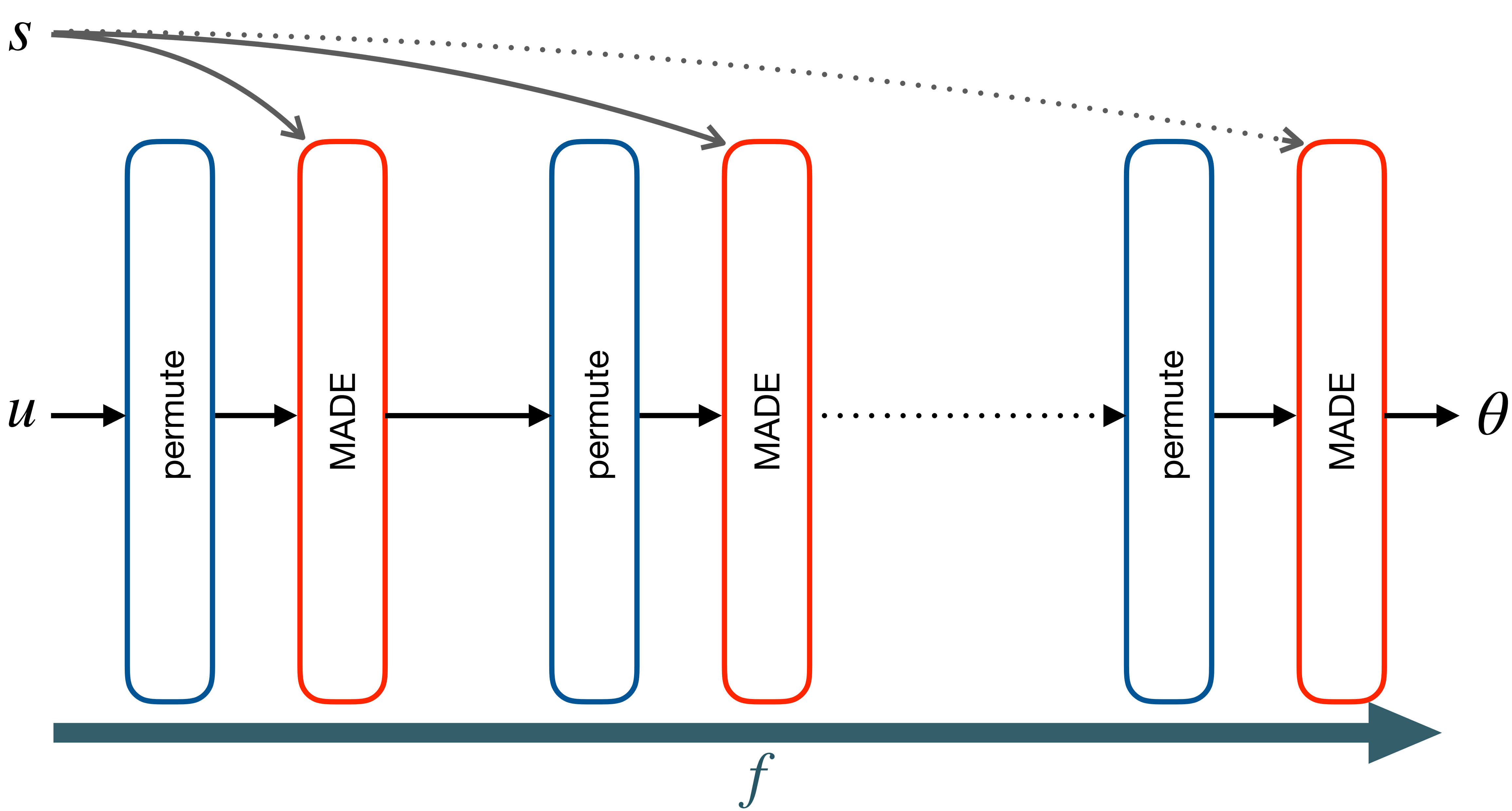


- Forward flow requires n passes.

Masked autoregressive flow

Papamakarios et al (2017)

- To achieve further generality, stack several MADE blocks, permuting components in between.

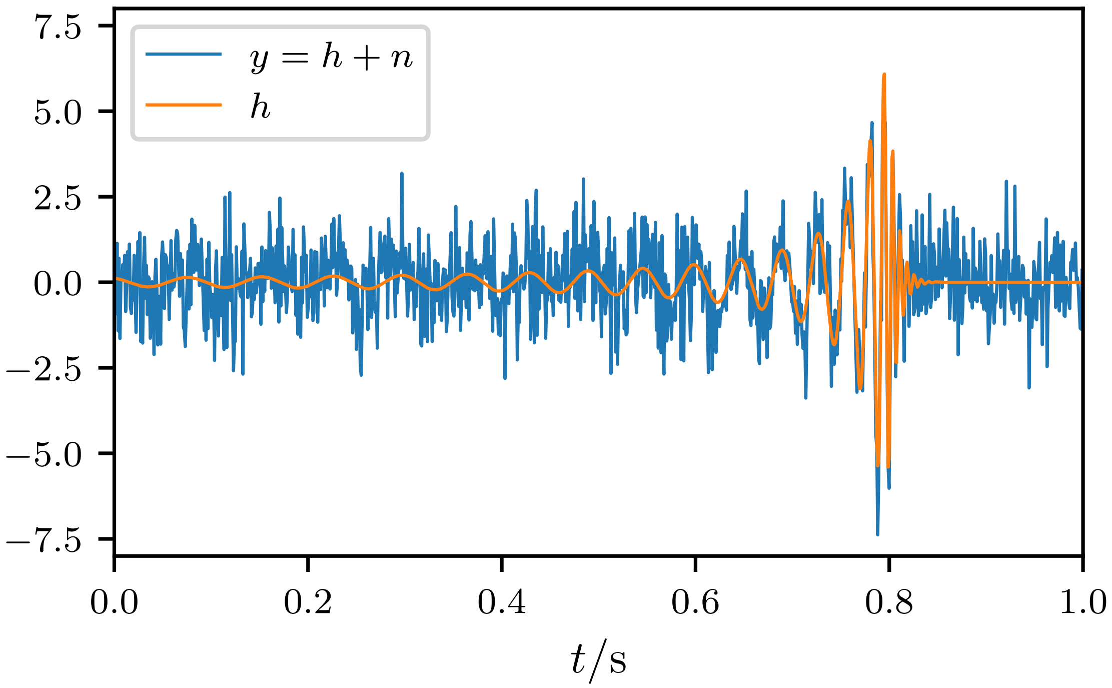


Training

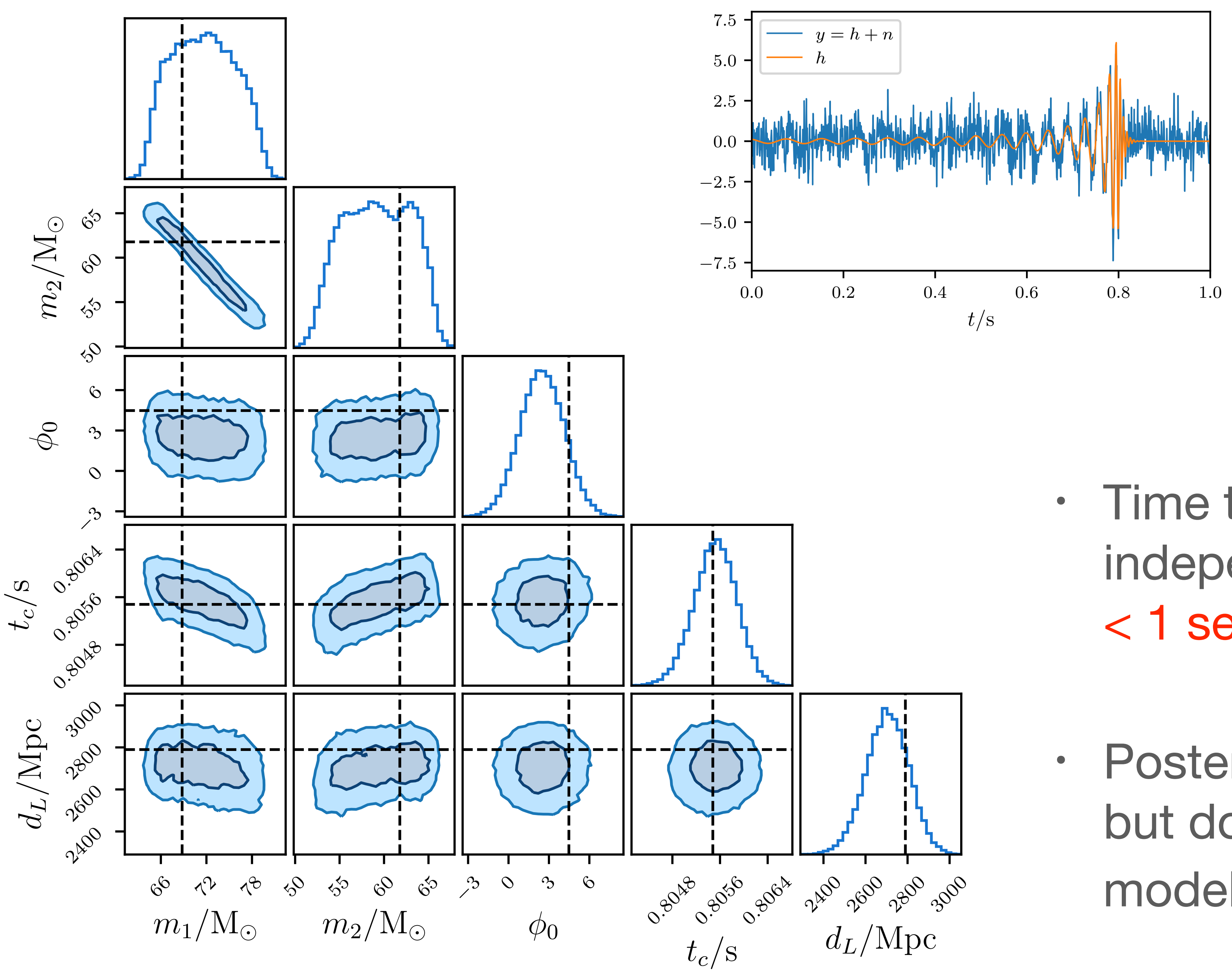
[same approach as Gabbard et al (2019)]

- Train on (θ, s) pairs:
  - $\theta \sim p(\theta)$, 10^6 samples
  - $s \sim p(s\,|\,\theta)$; $s = h(\theta) + n$
    - $h(\theta)$: 1 second long whitened (fixed PSD) inspiral-merger-ringdown waveforms at1024 Hz, stored in training set
    - n : stationary Gaussian noise sampled at train time
- Training time ~ 6 hours

$$
\begin{aligned}
35\ \mathrm{M}_\odot \leq\ & m_1, m_2 \leq 80\ \mathrm{M}_\odot, \\
1000\ \mathrm{Mpc} \leq\ & \quad d_L \quad \leq 3000\ \mathrm{Mpc}, \\
0.65\ \mathrm{s} \leq\ & \quad t_c \quad \leq 0.85\ \mathrm{s}, \\
0 \leq\ & \quad \phi_0 \quad \leq 2\pi,
\end{aligned}
$$



Sample posterior: MAF



- Time to draw 10,000 independent samples < 1 second.
- Posterior pretty good, but does not properly model ϕ_0

Variational autoencoder

Kingma and Welling (2013)

- To increase flexibility further, introduce latent variables z. These must be marginalized over to obtain posterior.

$$p(\theta \mid s) = \int p(\theta \mid z, s) p(z \mid s) \, \mathrm{d}z$$

Both described by neural networks

- This mixture of distributions is more general. To sample

  (i) draw latent variable from variational prior $z \sim p(z \mid s)$.

  (ii) draw parameters $\theta \sim p(\theta \mid z, s)$.

Variational autoencoder

Kingma and Welling (2013)

$$p(\theta \,|\, s) = \int p(\theta \,|\, z, s) p(z \,|\, s) \, \mathrm{d}z$$

- To train, would like to evaluate the posterior. But integral is intractable.

- Variational autoencoder introduces third model, the recognition model $q(z \,|\, \theta, s)$, which is an approximation to the variational posterior $p(z \,|\, \theta, s)$.

$$\begin{tikzcd} z \arrow[r, bend left, "p(\theta \,|\, z, s)"] & \theta \arrow[l, bend left, "q(z \,|\, \theta, s)"] \end{tikzcd}$$

- Training maximizes the variational lower bound on $p(\theta \,|\, s)$, namely

$$\mathcal{L} = \underbrace{\mathbb{E}_{q(z|\theta,s)} \log p(\theta|z,s)}_{\text{reconstruction loss}} - \underbrace{D_{\mathrm{KL}} \left(q(z|\theta,s) \| p(z|s) \right)}_{\text{KL loss}}$$

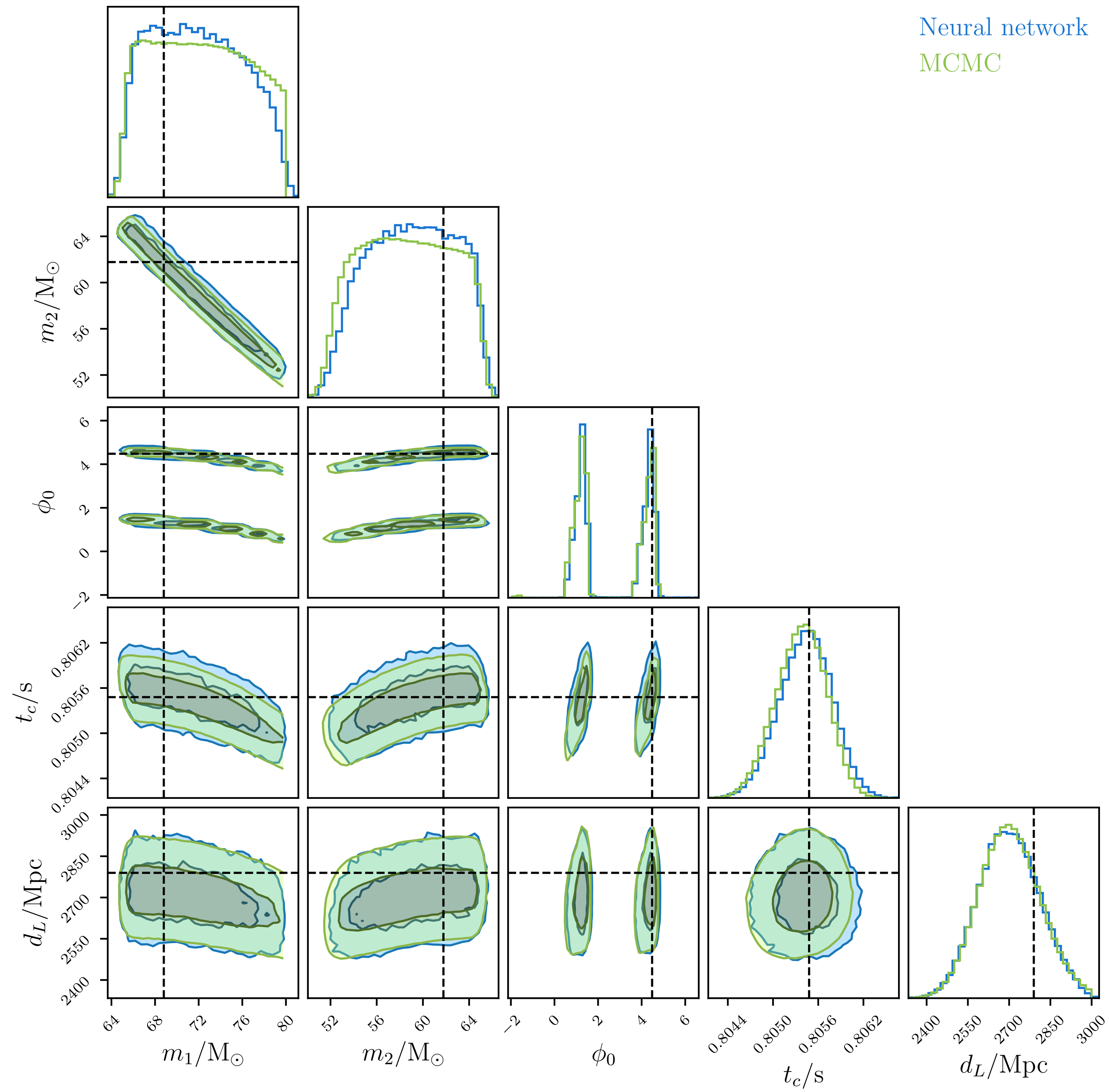
- Applied by Gabbard et al (2019) to gravitational waves: With all 3 networks Gaussian, obtained similar performance to MAF.

Variational autoencoder with normalizing flows

- $p(\theta \mid z, s)$
  $p(z \mid s)$
  $q(z \mid \theta, s)$
  all taken to be MAFs.

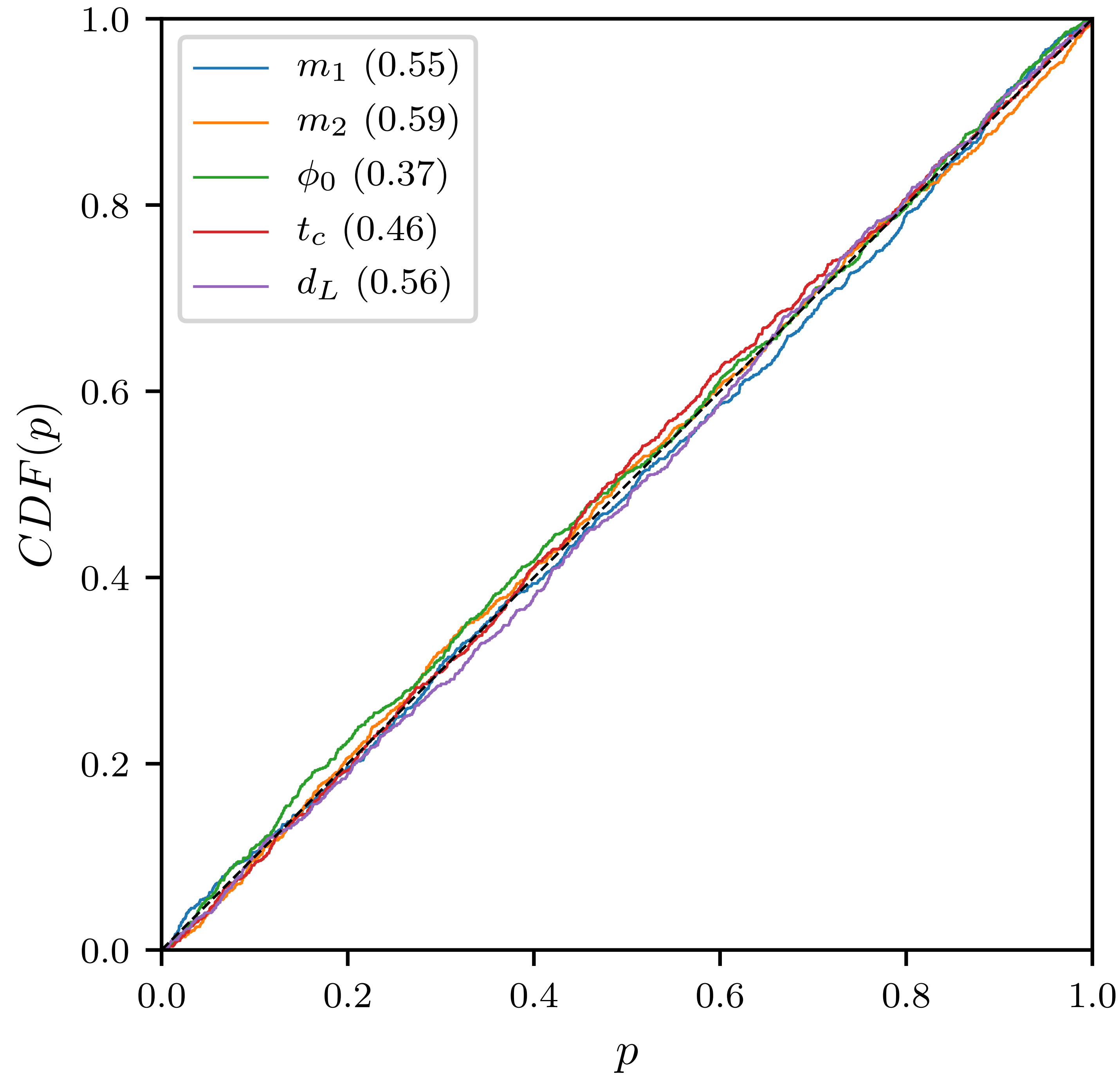
- Training time ~ 15 hours

- Posterior comparable to MCMC.

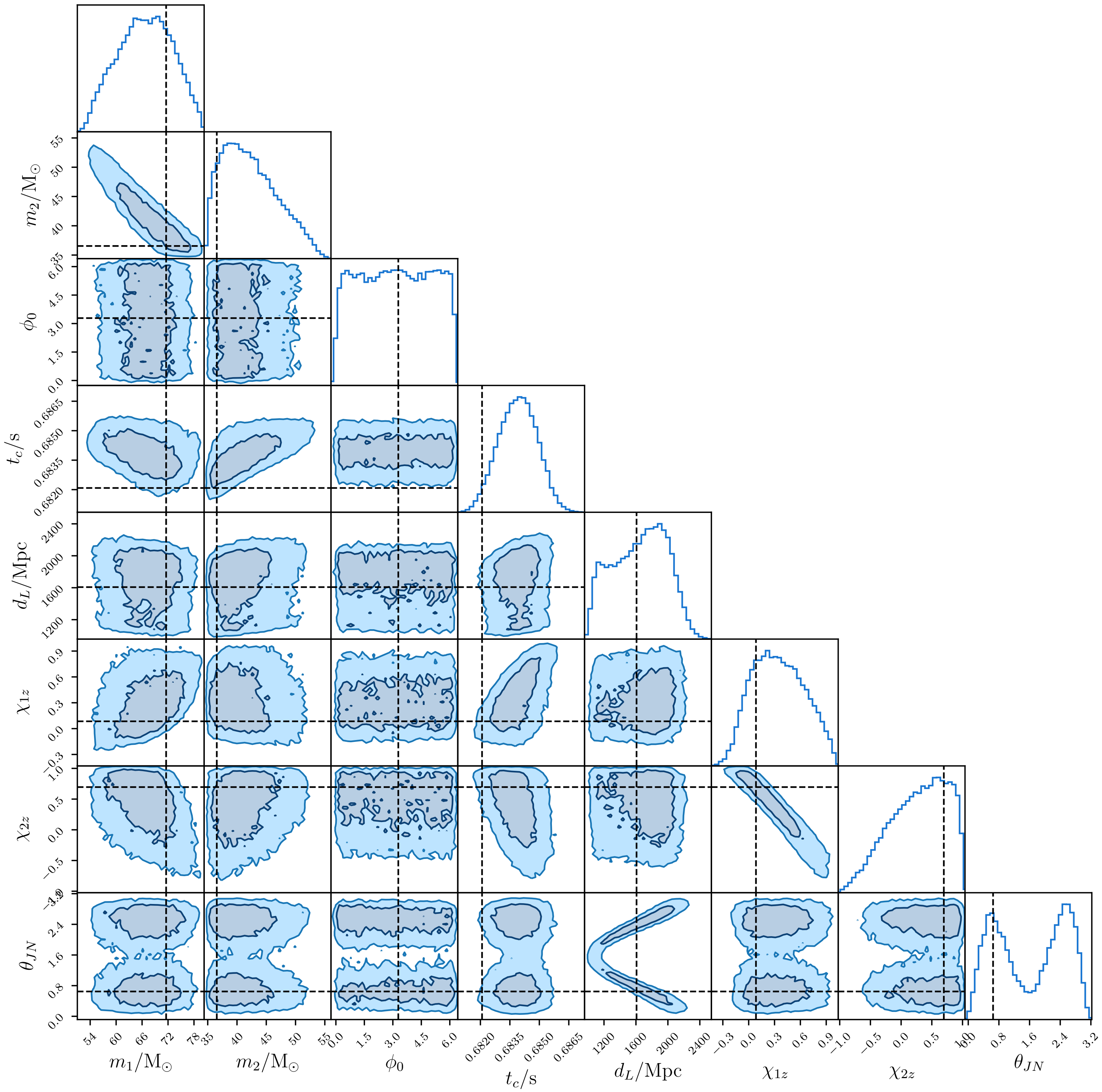


P—P plot

- For each one-dimensional marginalized posterior, study distribution of percentile values of true parameters.
- 1000 different waveforms + noise realizations.



Adding aligned spins and inclination



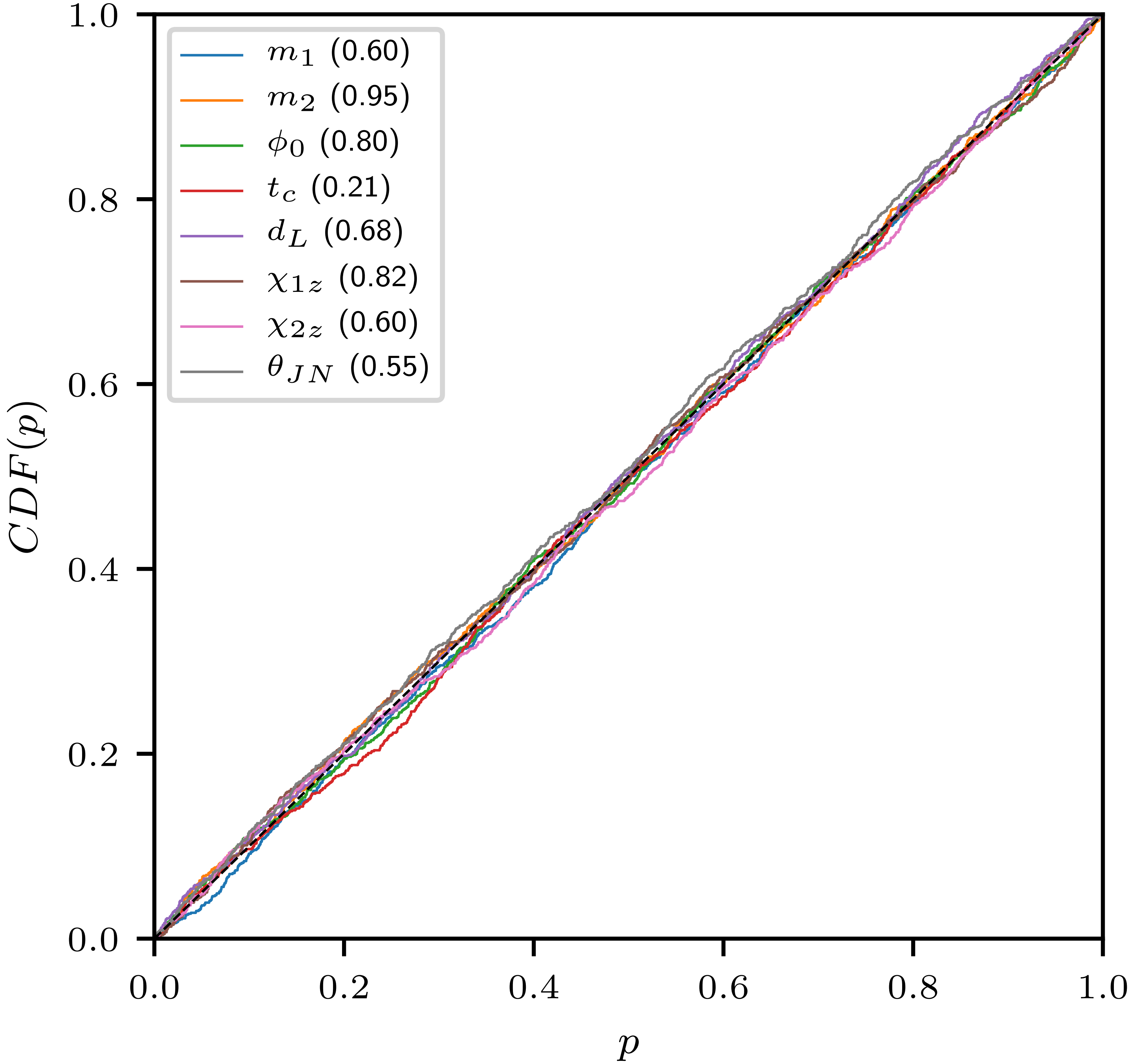
- Prior ranges

$$\begin{aligned} 35\ \mathrm{M}_\odot \le m_1, m_2 &\le 80\ \mathrm{M}_\odot, \\ 1000\ \mathrm{Mpc} \le \quad d_L \quad &\le 3000\ \mathrm{Mpc}, \\ 0.65\ \mathrm{s} \le \quad t_c \quad &\le 0.85\ \mathrm{s}, \\ 0 \le \quad \phi_0 \quad &\le 2\pi, \\ -1 \le \chi_{1z}, \chi_{2z} &\le 1, \\ 0 \le \quad \theta_{JN} \quad &\le \pi. \end{aligned}$$

- Slightly larger network

- Sampling time now ~ 2 seconds for 10,000 samples.

P—P plot



~ 30 minutes to generate all samples

Next steps

- Expand to full 15D parameter space: multiple detectors, sky position, non-aligned spins.
- Allow the noise PSD to vary from event to event.
- Waveform “compression” to allow lower mass BBH, and BNS events. These involve longer waveforms, and higher sampling frequency.
- Try to reduce size of training set.

Conclusions

- For single detector, aligned spin binaries, neural networks are capable of modeling multimodal $p(\theta | s)$.
    - Training is likelihood-free, requiring only (θ, s) pairs from the data generative process.
    - After training, < 2 seconds to produce 10,000 independent samples. Compares to days for standard methods.
- Model with CVAE *and* MAF has best performance:
    - Successfully models all parameters, including degeneracies.
    - Posterior comparable to MCMC.
    - Passes P—P plot statistical tests.
- Ongoing work to develop into a complete parameter estimation tool.

THANK YOU