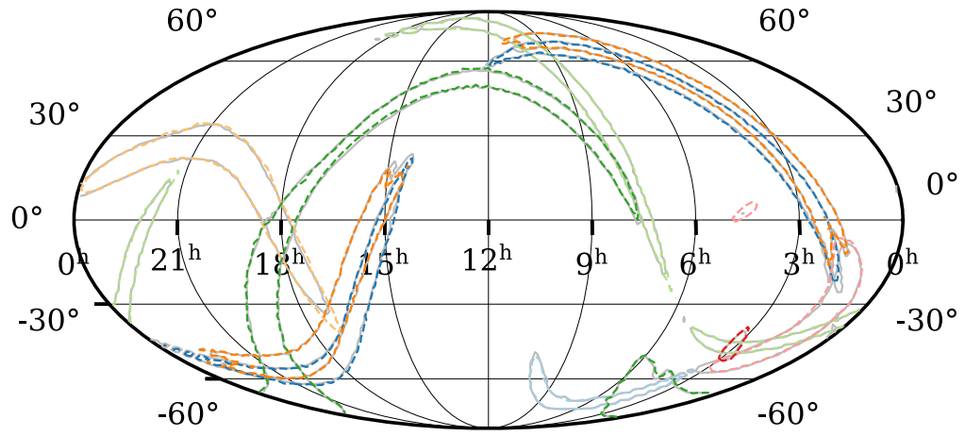


Real-time GW parameter estimation using machine learning

Stephen Green
Max Planck Institute for Gravitational Physics - Potsdam

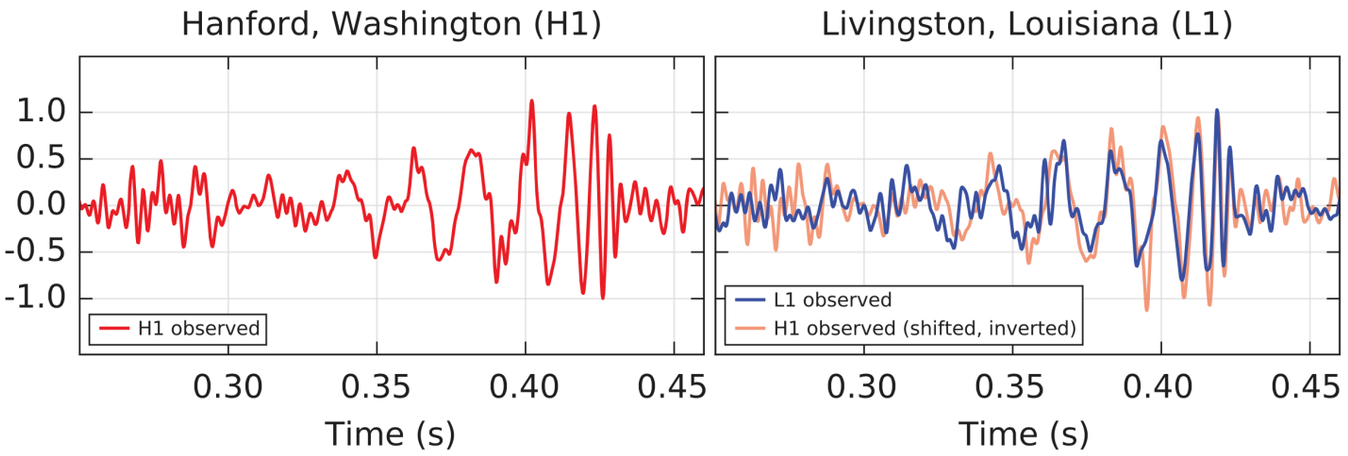
[with **M. Dax**, J. Gair, J. Macke, A. Buonanno, B. Schölkopf]

Workshop on Source inference and parameter estimation in GW Astronomy
IPAM, UCLA
November 17, 2021

Outline

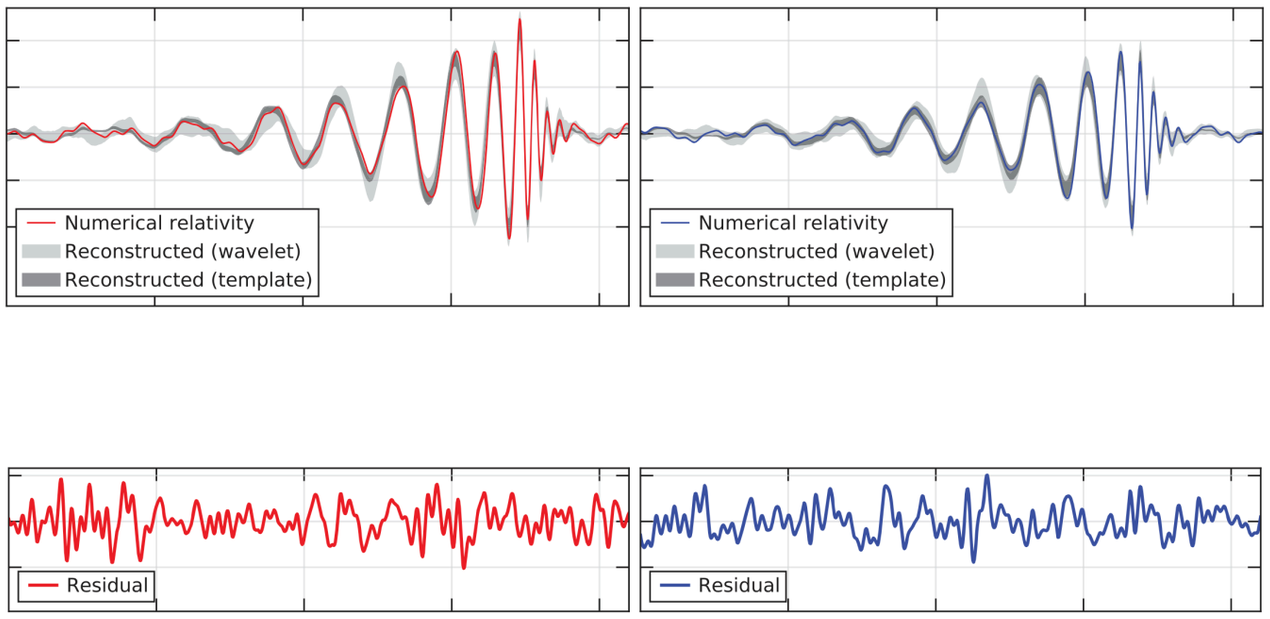
1. Parameter estimation review
2. Machine-learning approach: Neural posterior estimation
3. Application to real data
4. Validation of results
5. Outlook

Introduction to parameter estimation



data

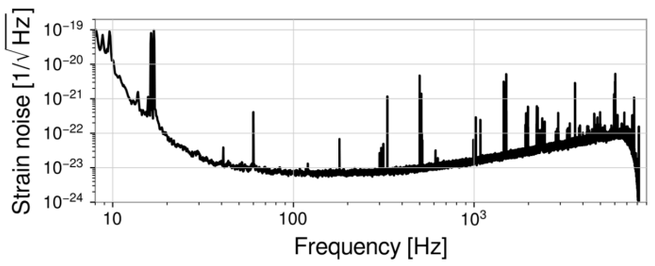
=



signal $h(\theta)$

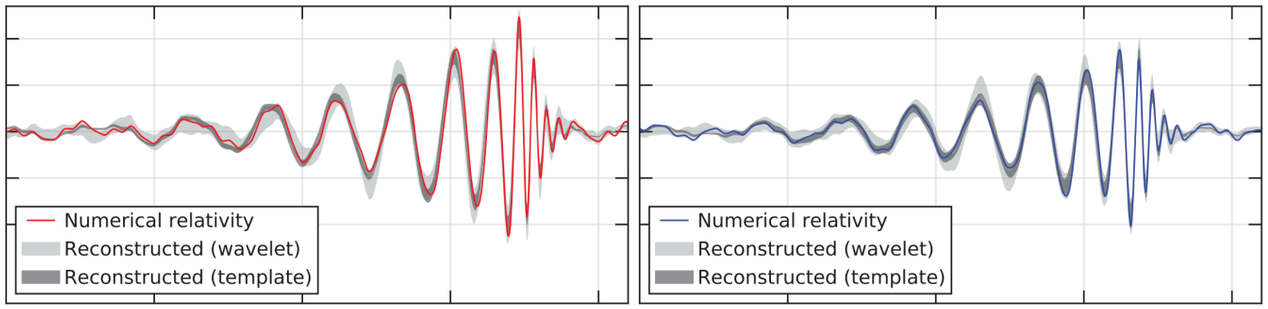
+

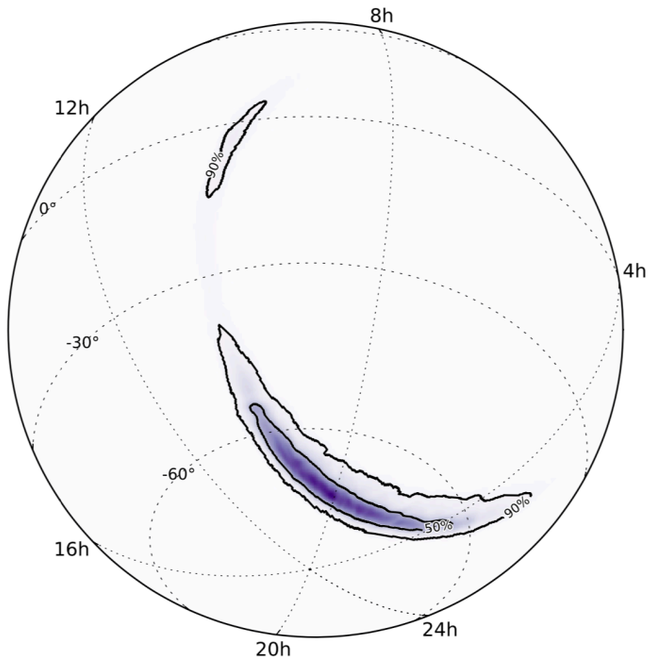
noise



Images: Abbott et al (2016)

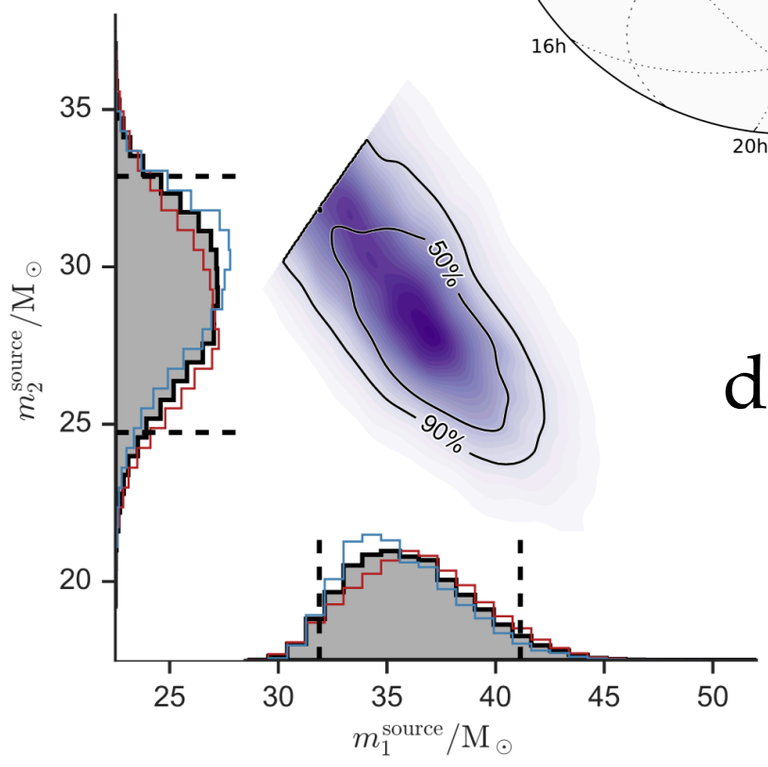
Introduction to parameter estimation





Bayes' theorem

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



posterior distribution

Introduction to parameter estimation

Posterior

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

Likelihood assumes **stationary Gaussian** noise

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

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

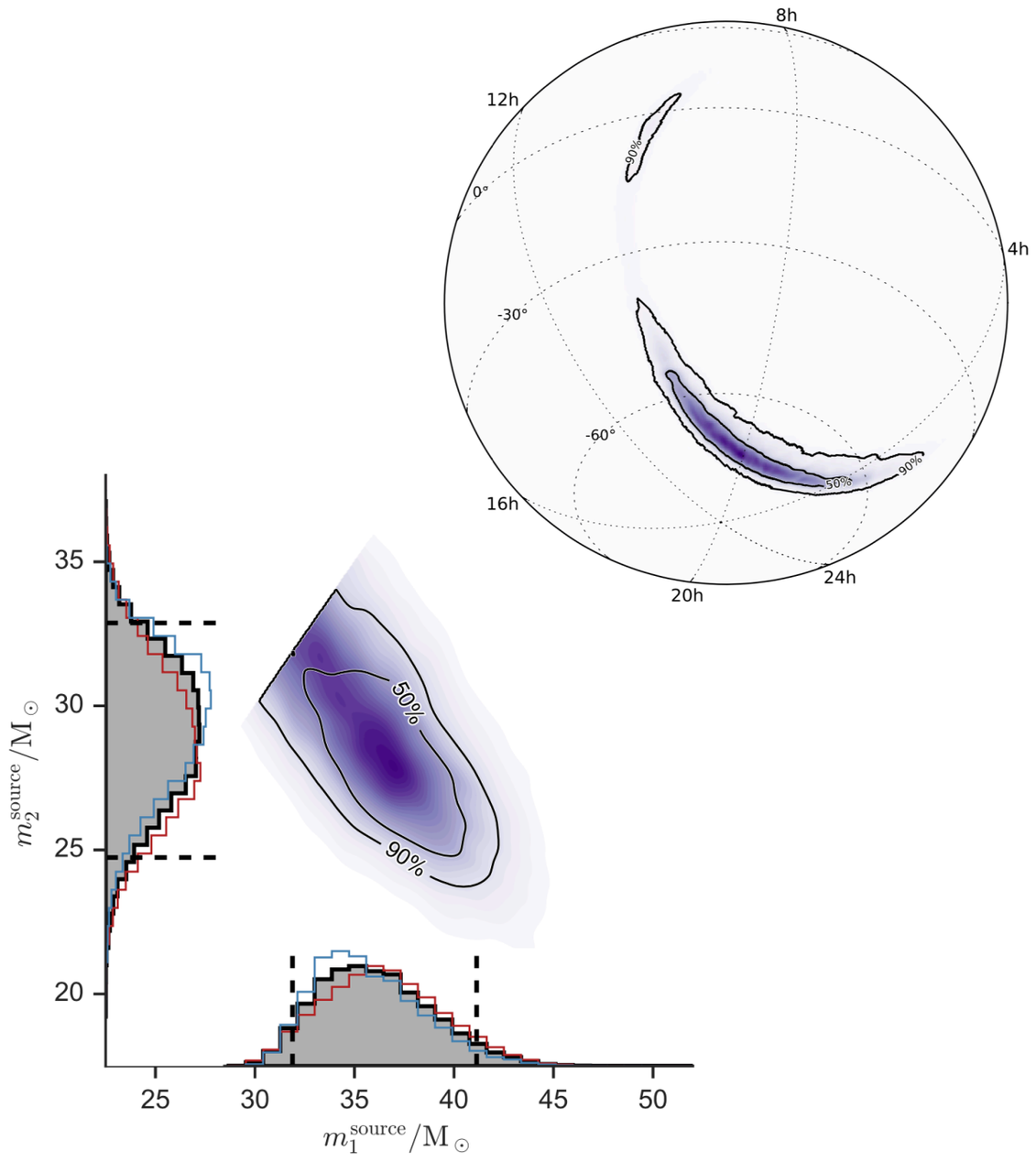
Prior

e.g., uniform in m_1, m_2 over some range,
uniformly distributed in sky,
etc.

Sampling

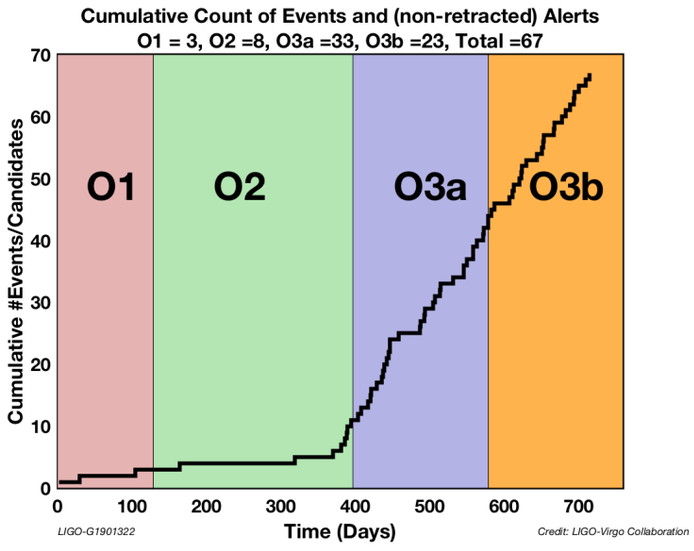
- **Stochastic:** Markov chain Monte Carlo (MCMC) or nested sampling

$$\theta \sim p(\theta \mid d)$$



Challenges

1. **Expensive!**
   Millions of likelihood evaluations
   $\Longrightarrow$ **days to weeks / event**



Simulation-based inference

2. **Restrictive assumptions!**
   Stationary Gaussian noise needed to evaluate likelihood

Simulation-based inference

- Two options for $p(d|\theta)$:

    1. **Evaluate** $p(d|\theta)$

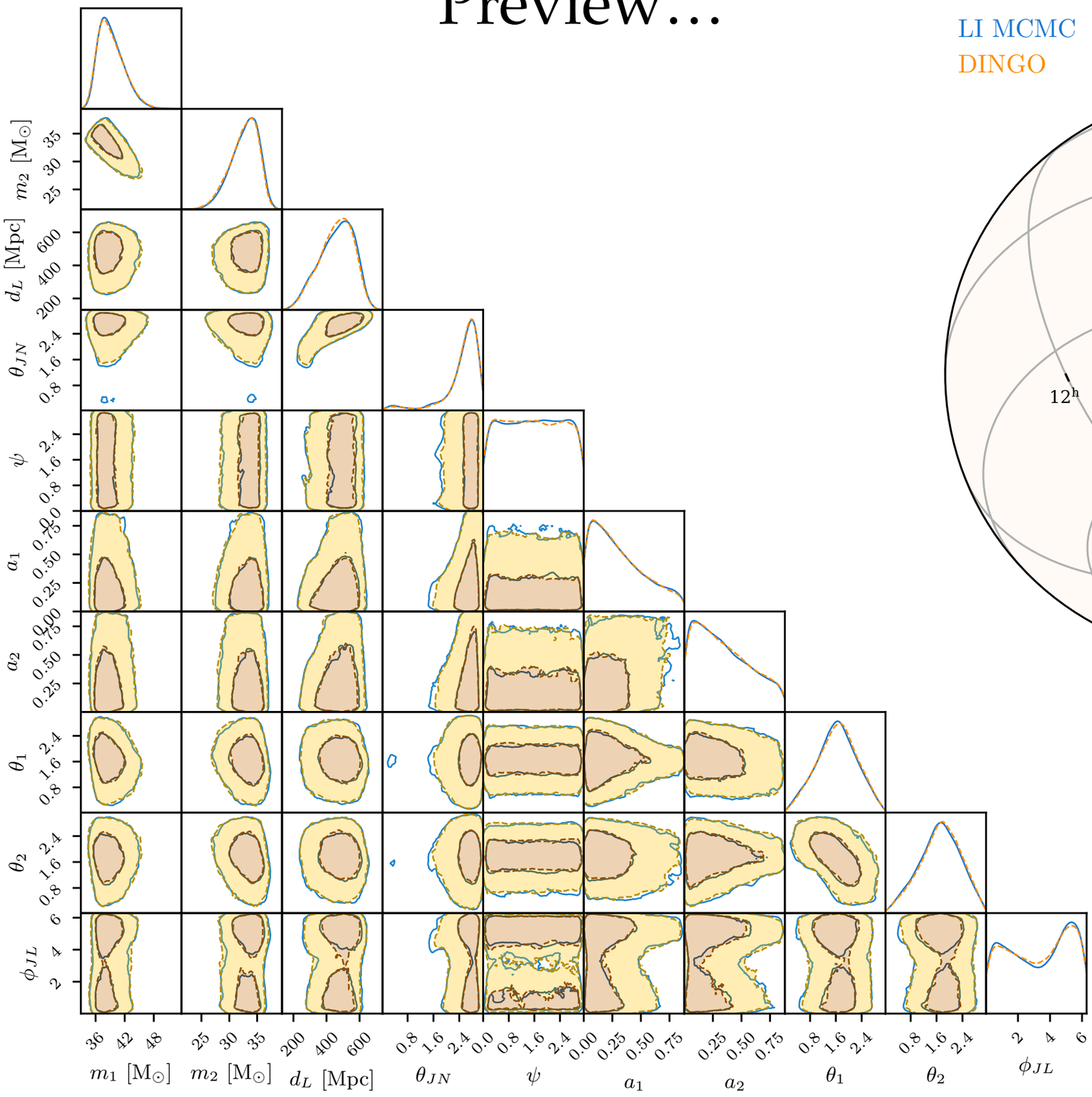
        - Likelihood-based inference

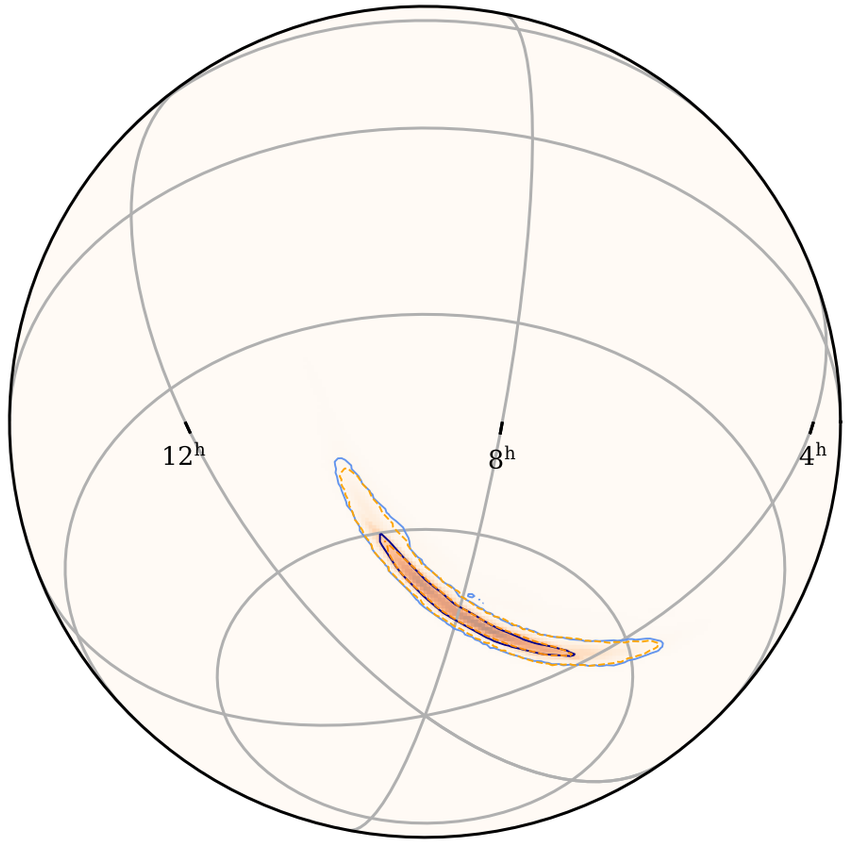
    2. **Sample** $d \sim p(d|\theta)$

        - Simulation-based inference / likelihood-free inference

- Both approaches use $p(d|\theta)$, just in different ways.

Preview…


m_1 [M$_\odot$]
m_2 [M$_\odot$]
d_L [Mpc]
θ_{JN}
ψ
a_1
a_2
θ_1
θ_2
ϕ_{JL}


LI MCMC
DINGO
12h
8h
4h

Neural posterior estimation

$q(\theta \mid d)$

Neural-network conditional probability distribution

- Extremely fast sampling and density estimation
- Amortized

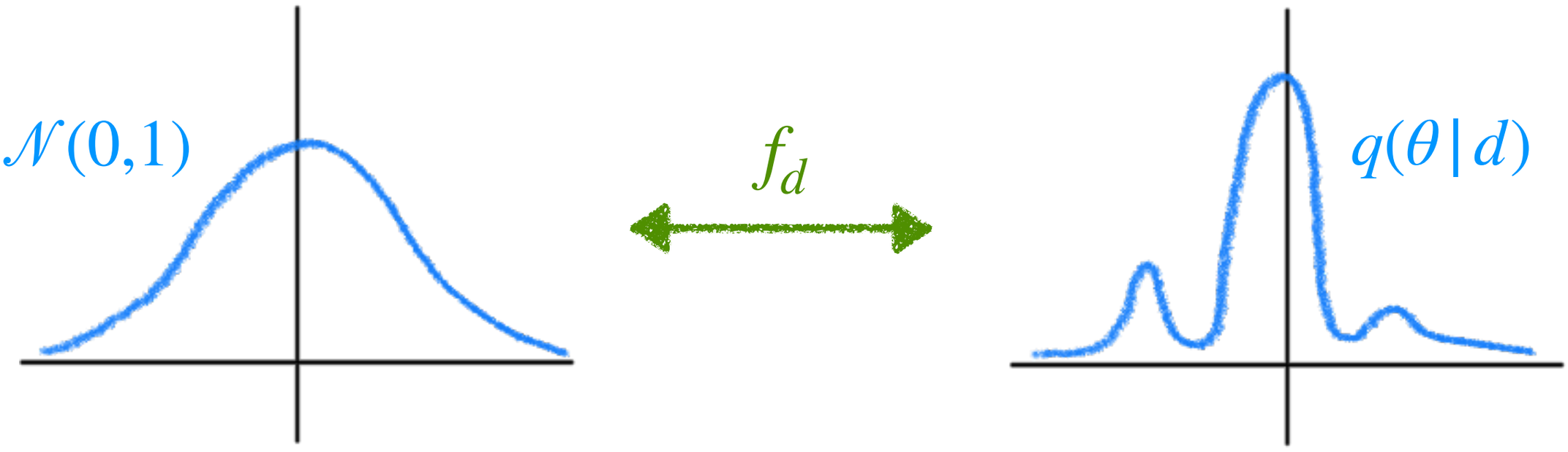
Train to model the posterior $q(\theta \mid d) \approx p(\theta \mid d)$

- Uses simulated data $(\theta, d) \sim p(\theta, d)$, but no actual posterior samples!

> "A surrogate for the posterior"

Normalizing flow

- A **normalizing flow** $f_d : u \mapsto \theta$ defines a complex distribution in terms of a simple one



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

Normalizing flow

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

- **Requirements:**
  1. Invertible
  2. Simple Jacobian determinant

- Parametrize f_d using **neural network.**

Fast to **evaluate** and **sample** from $q(\theta \mid d)$

Normalizing flow

- **Requirements:**

1. Invertible ✓
2. Simple Jacobian determinant ✓ $\det J_{f_d} = \prod_{i=\frac{D}{2}+1}^{D} c_i' \left(u_i; u_{1:\frac{D}{2}}, d\right)$

- Use a sequence of "coupling transforms":

$$c_{d,i}(u) = \begin{cases} u_i & \text{if } i \leq D/2 \\ c_i \left(u_i; u_{1:\frac{D}{2}}, d\right) & \text{if } i > D/2 \end{cases}$$

Hold fixed half of the components

Transform remaining components element-wise, conditional on other half and *s*.

- c_i should be differentiable and have analytic inverse with respect to u_i.

Normalizing flow

- Spline flow (Durkan et al, 2019):

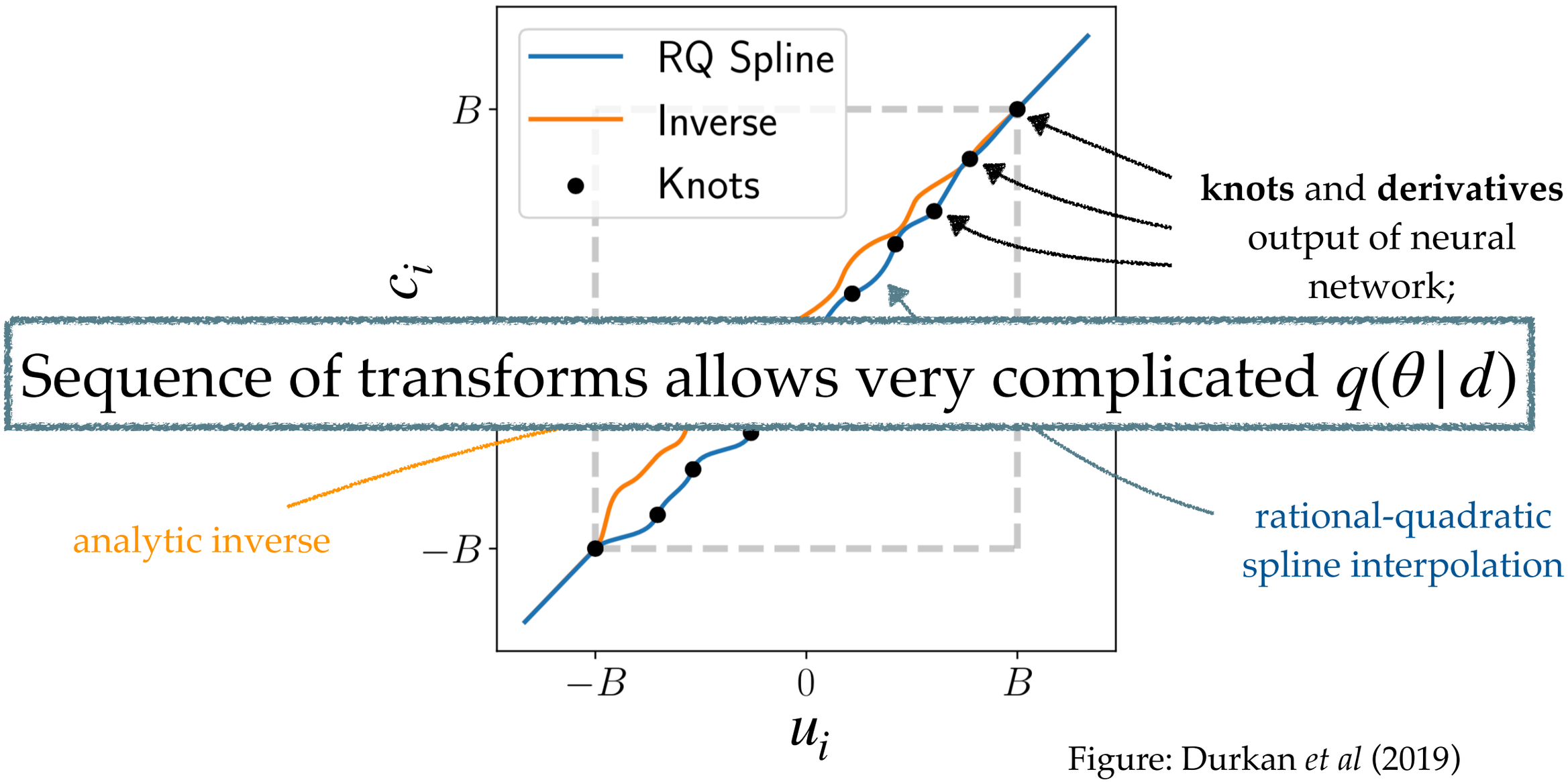


Figure: Durkan *et al* (2019)

Simulation-based training

- Minimize expectation over $d \sim p(d)$ of cross-entropy between $q(\theta|d)$ and $p(\theta|d)$

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

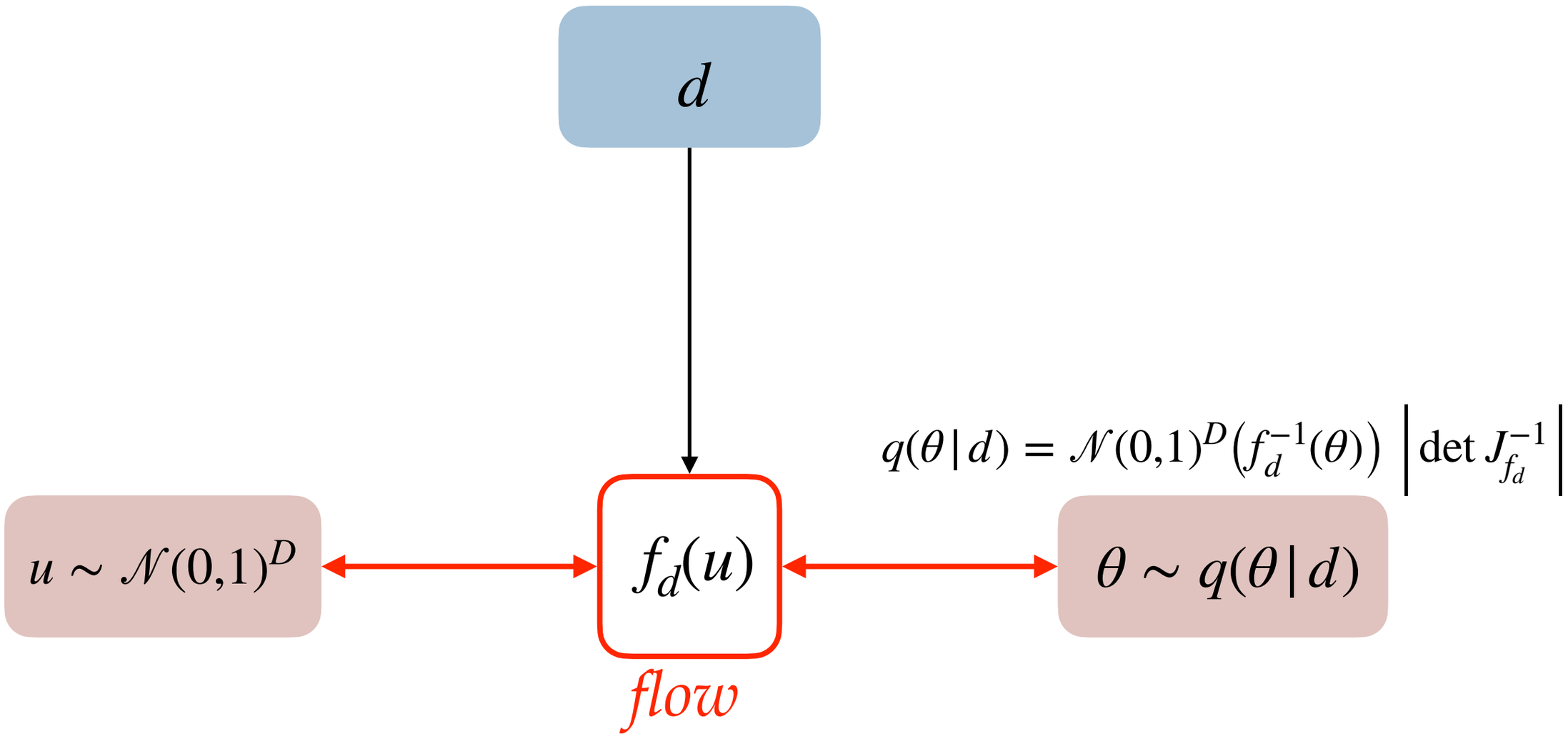
Much more tractable ordering!

Could also use real noise!

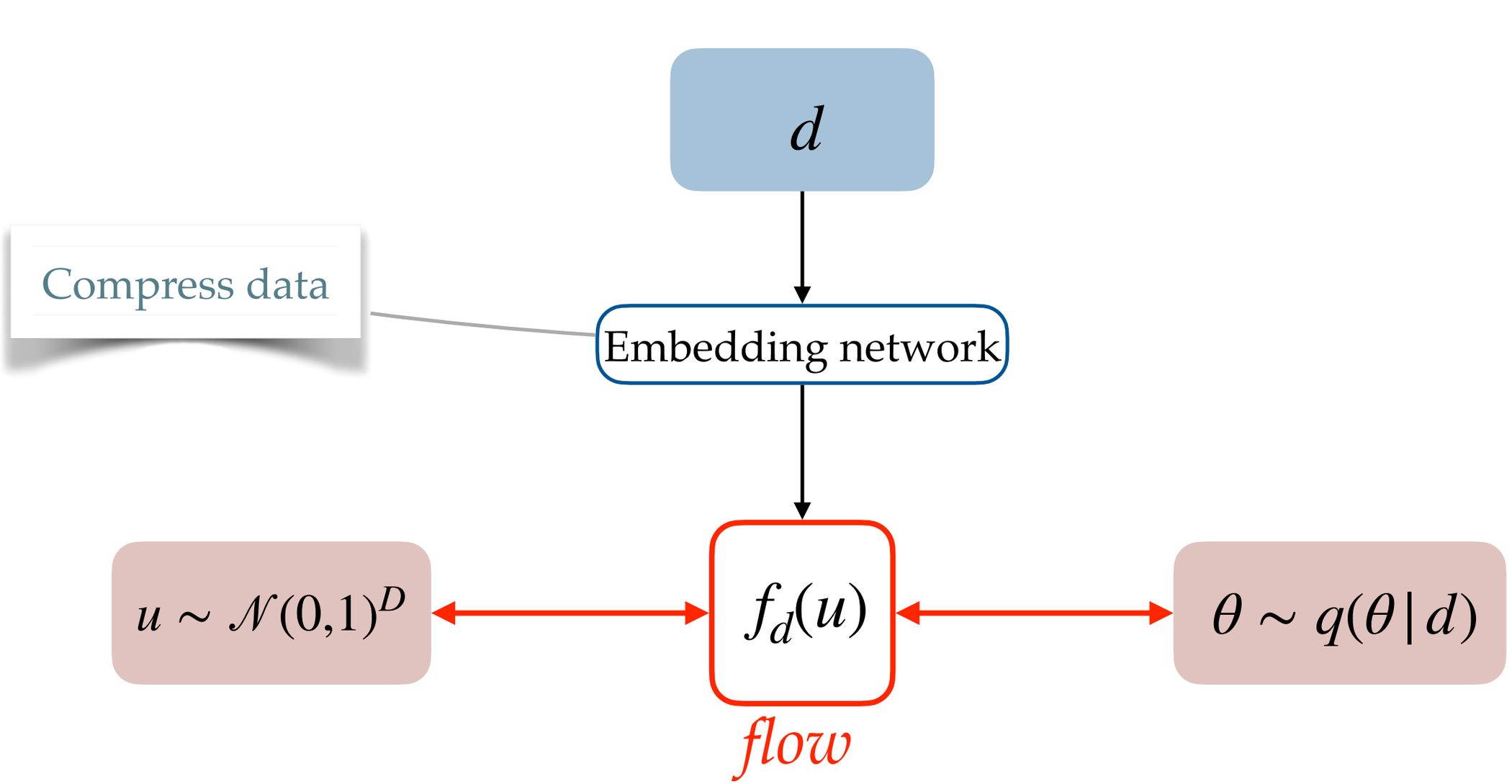
1. **Sample prior** $\theta^{(i)} \sim p(\theta),\ i = 1, \ldots, N$
2. **Simulate data** $d^{(i)} \sim p(d|\theta^{(i)});\ d^{(i)} = h(\theta^{(i)}) + n^{(i)}$ with $n^{(i)} \sim p_{S_n}(n)$
3. $L \approx \frac{1}{N} \sum_{i=1}^{N} \left[-\log q(\theta^{(i)}|d^{(i)}) \right]$

Normalizing flow

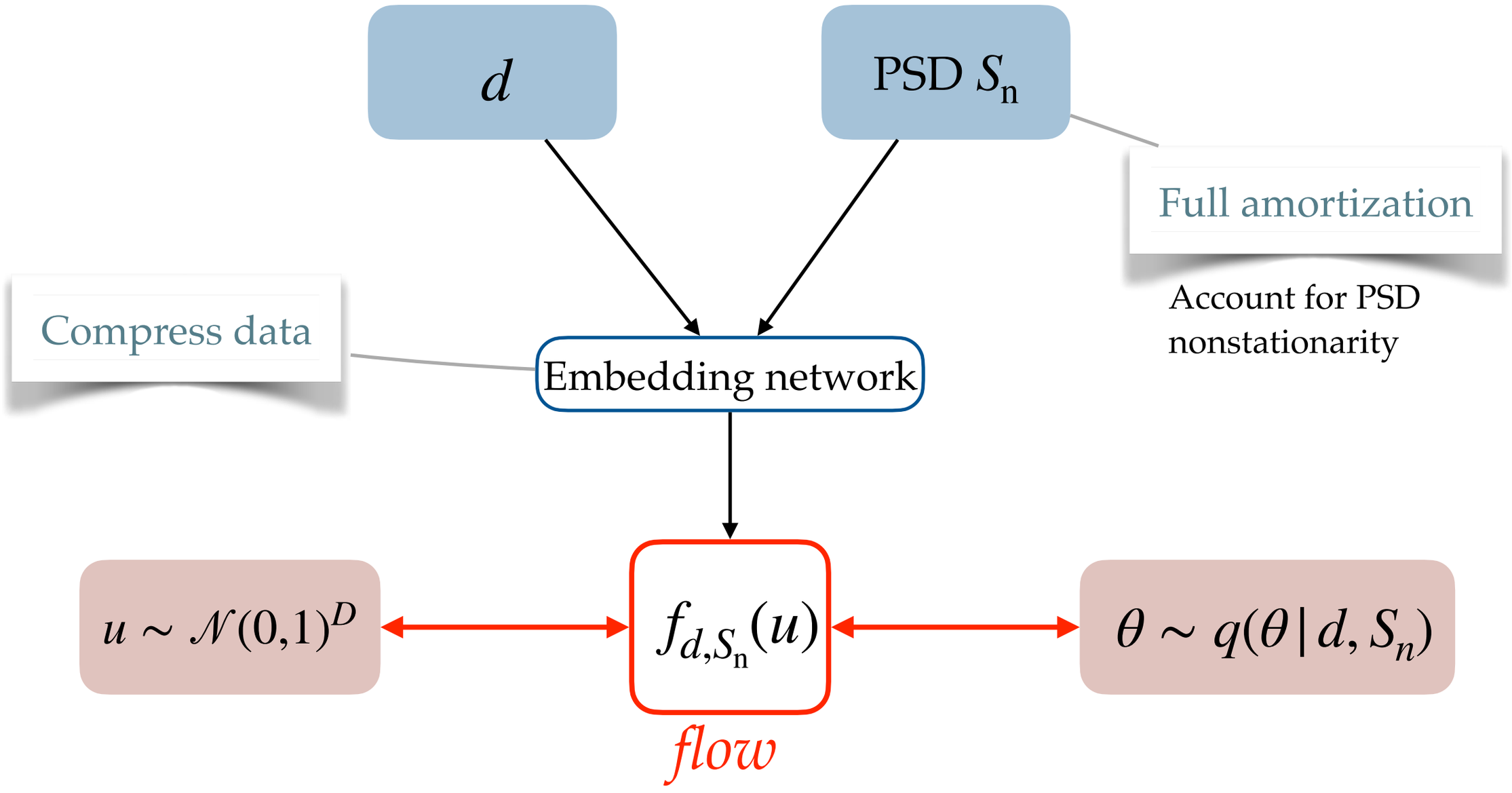
Express posterior in terms of a mapping from normally-distributed variables.



Normalizing flow

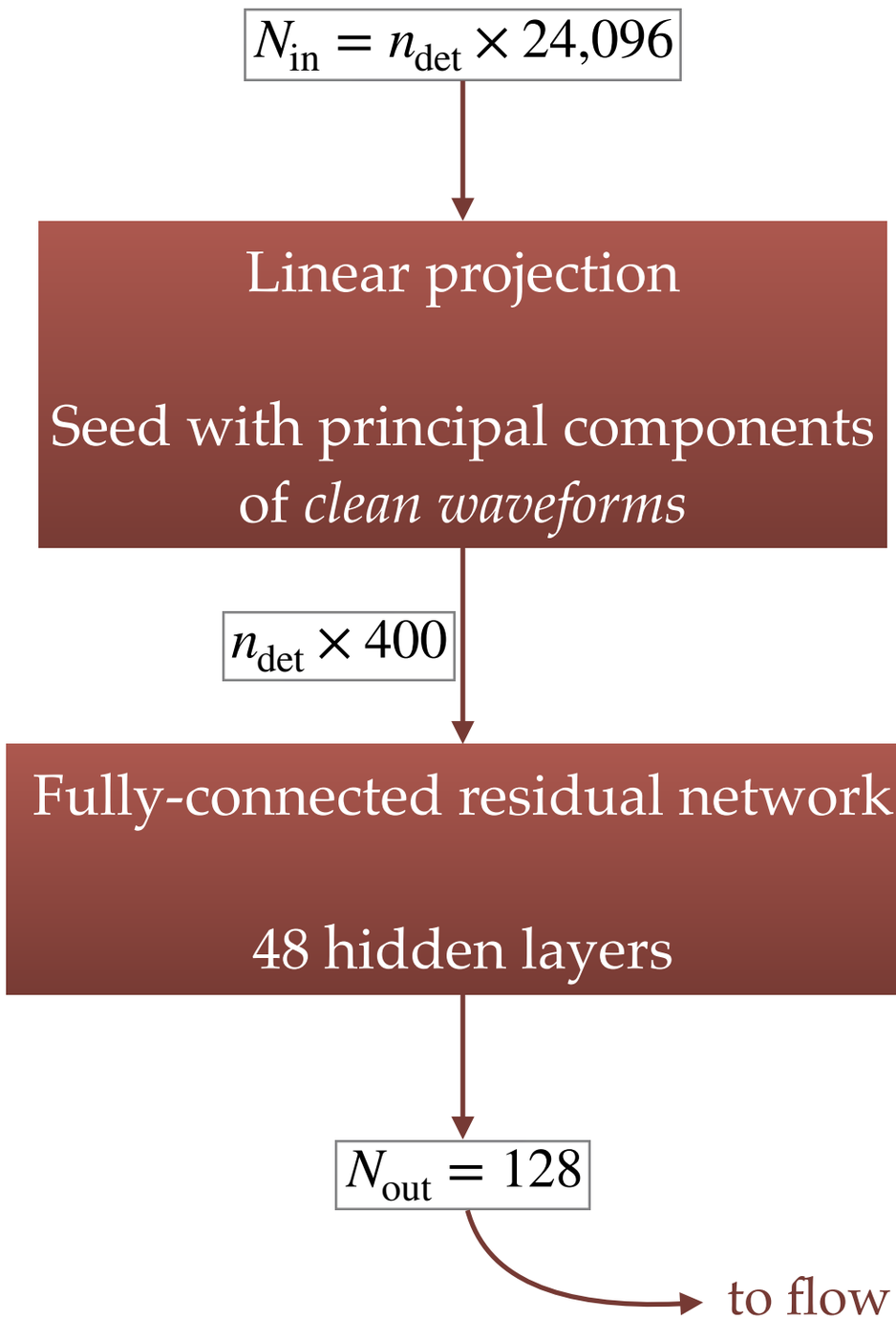


Normalizing flow



Big neural networks: ≈ 350 layers and 150 million parameters

Embedding network



Each detector has
(8 x 1024 frequency bins) x (Re + Im + PSD)

Inductive bias to recognize signal waveforms

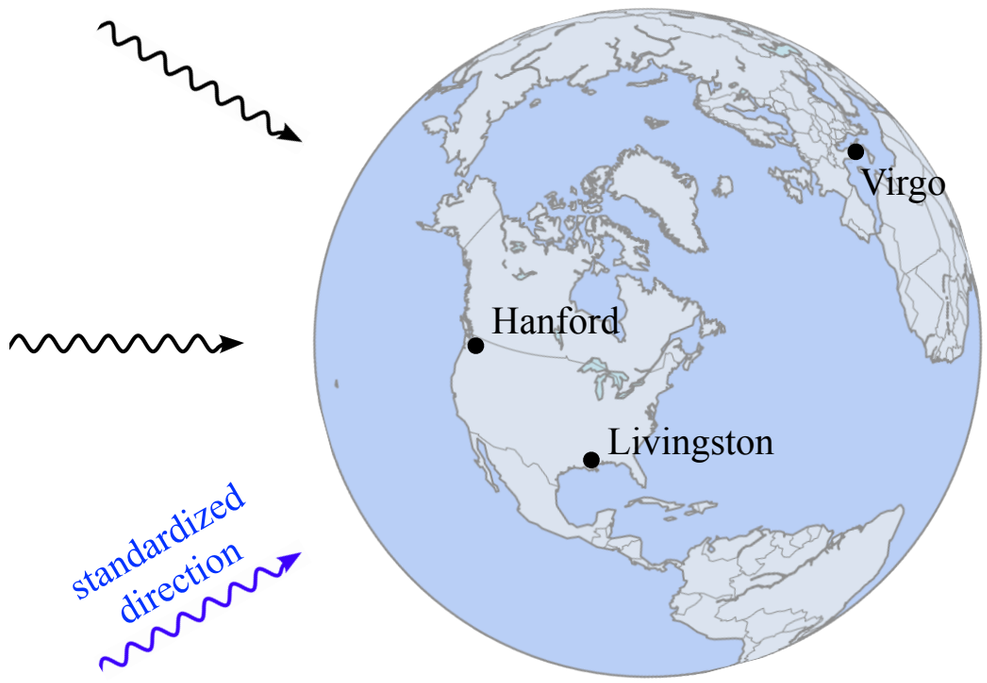
Nonlinear compression

Group equivariant neural posterior estimation

- Inferring sky position (α, δ) + coalescence time t_c
  - Network must learn to interpret different arrival times t_I in each detector

→ **hard**

  - If we knew (α, δ, t_c) could manually align the data to simplify analysis



→ **but we don't know this *a priori***

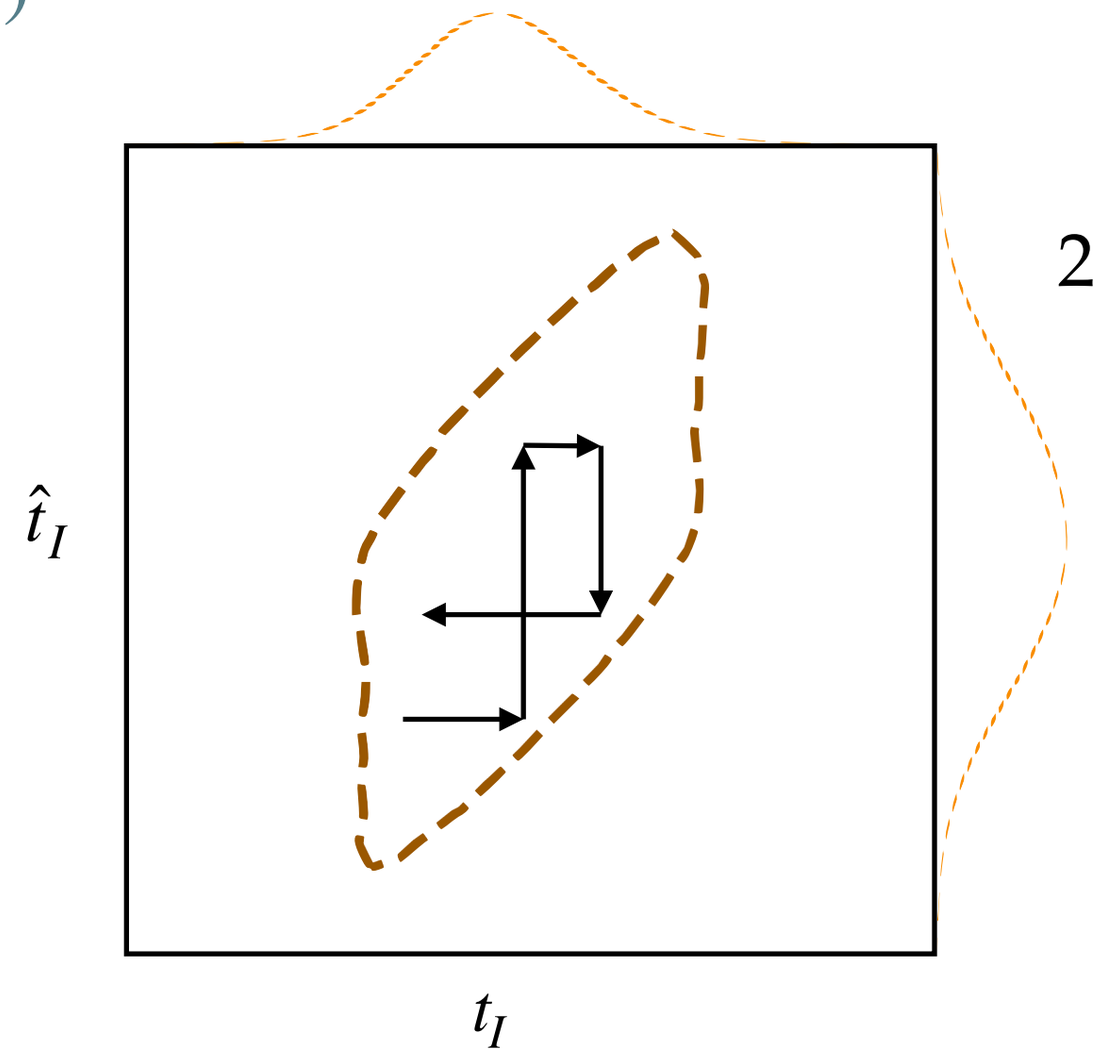
Group equivariant neural posterior estimation

- Expand parameter space to include **blurred** coalescence times $\hat{t}_I$

$$\longrightarrow \quad p(\theta, \hat{t}_I \,|\, d)$$

- **Gibbs sampling**

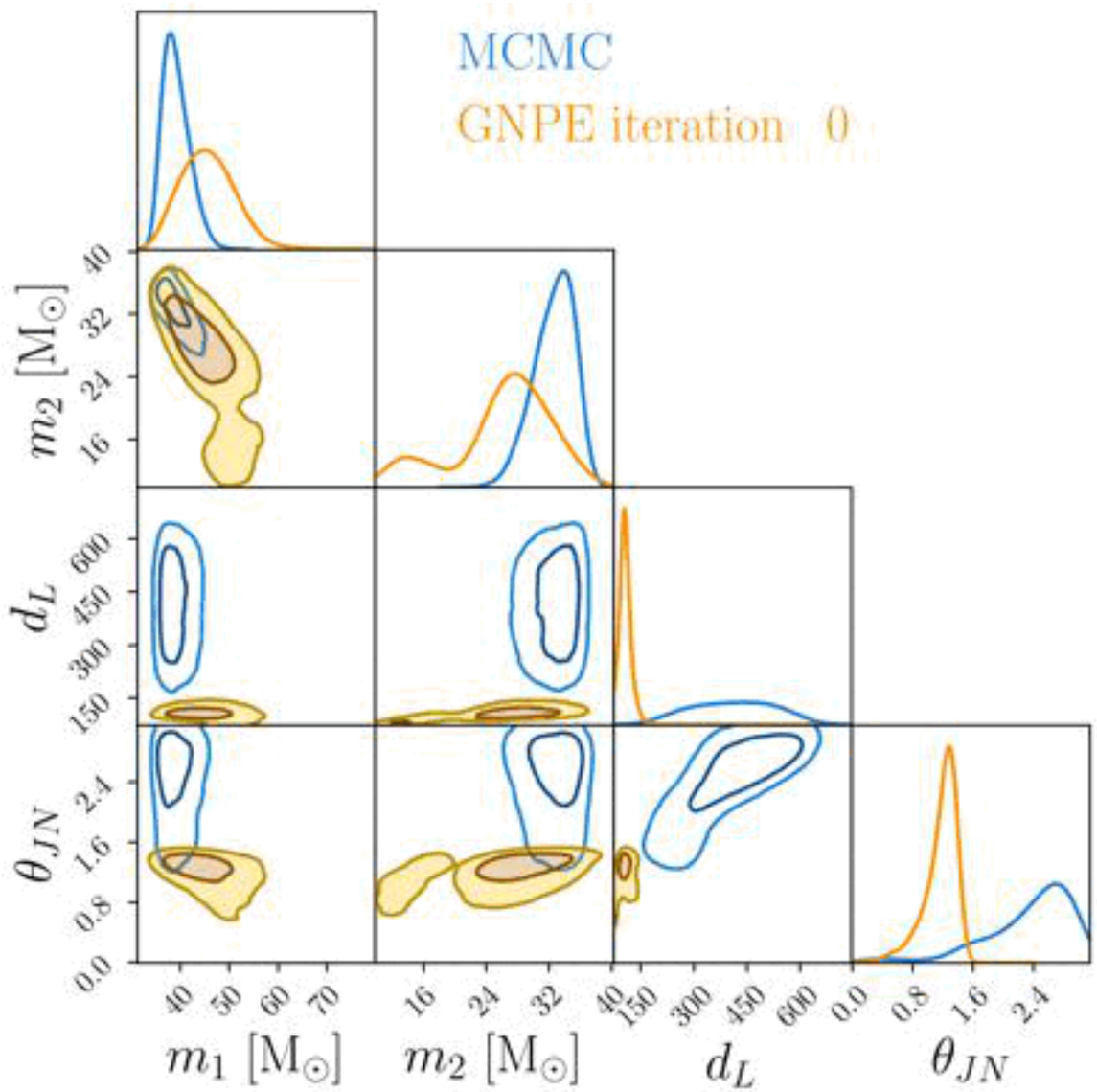
1. $\theta \sim q(\theta \,|\, T_{-\hat{t}_I}(d), \hat{t}_I)$
   - align data based on $\hat{t}_I$



2. $\hat{t}_I \sim p(\hat{t}_I \,|\, t_I)$
   - fixed kernel

Group equivariant neural posterior estimation

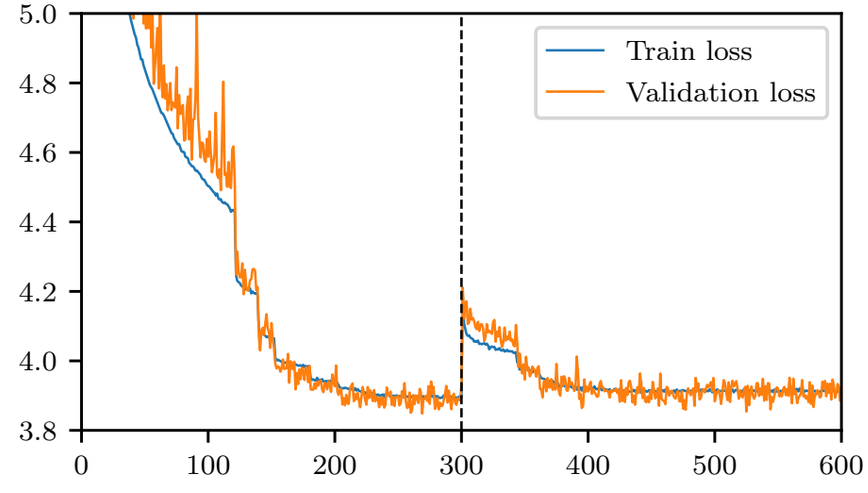
- **Real-time convergence** with $O(10)$ iterations



Simplified inference task by using symmetries

Training

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



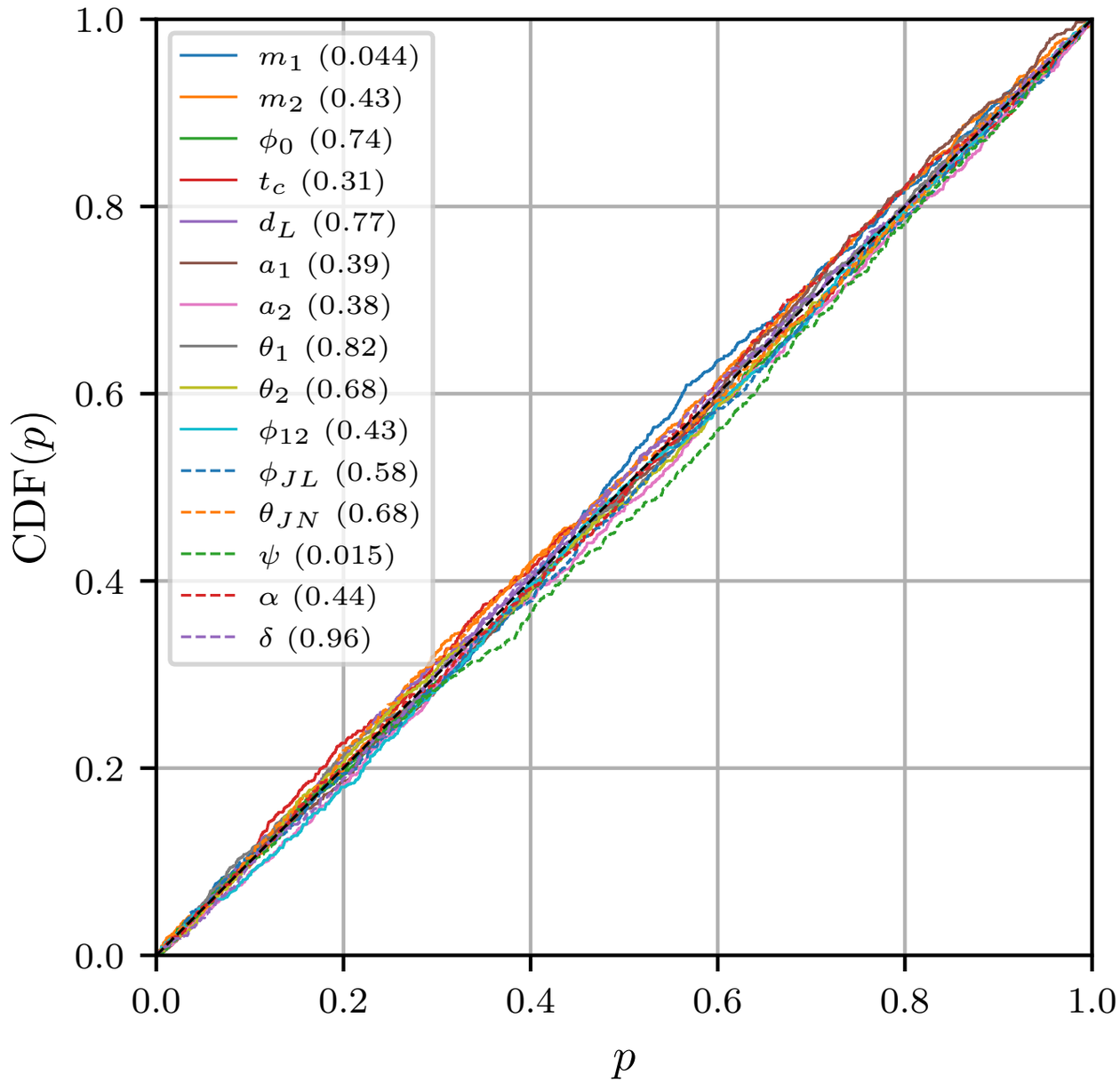
- + stationary Gaussian noise realizations consistent with given PSD

- 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]

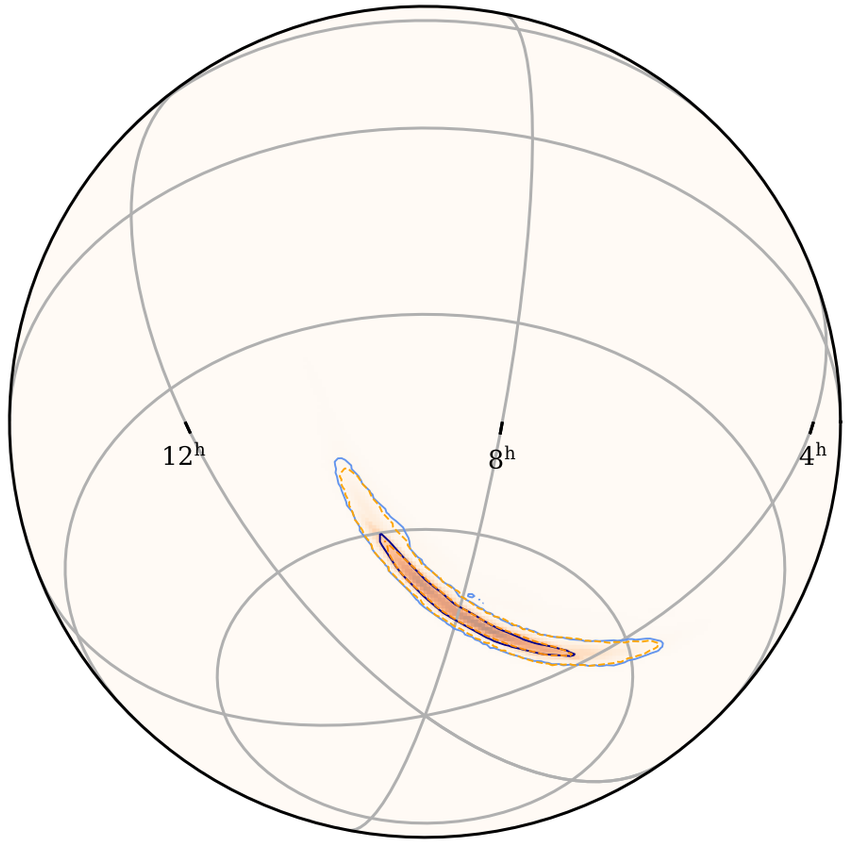
Importance of method validation

- Uncertainty estimates allow for **consistency checks**
  - "within-distribution"
- On individual events, **compare posteriors** against standard tools.
  - "out-of-distribution"

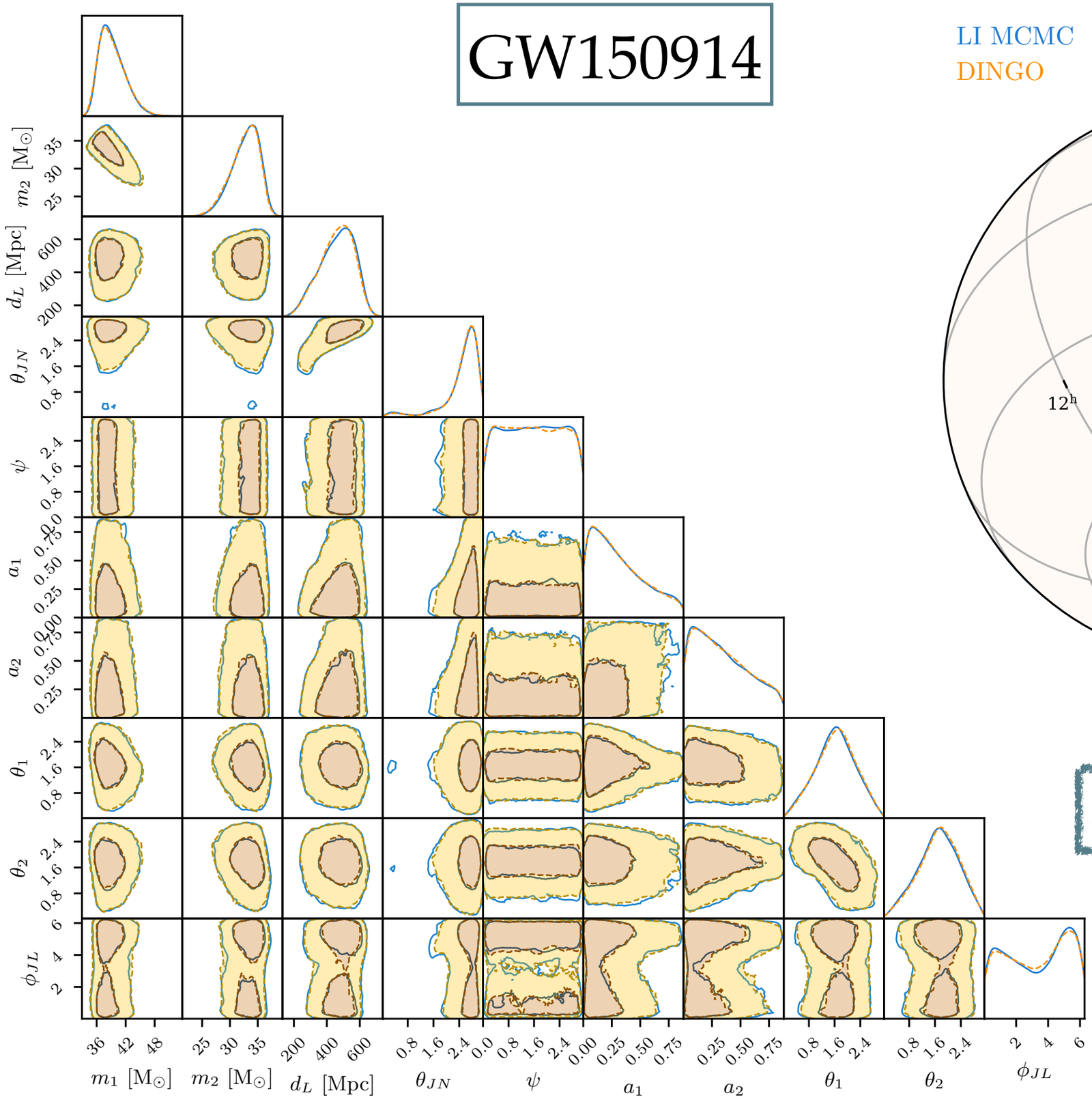


GW150914

LI MCMC
DINGO

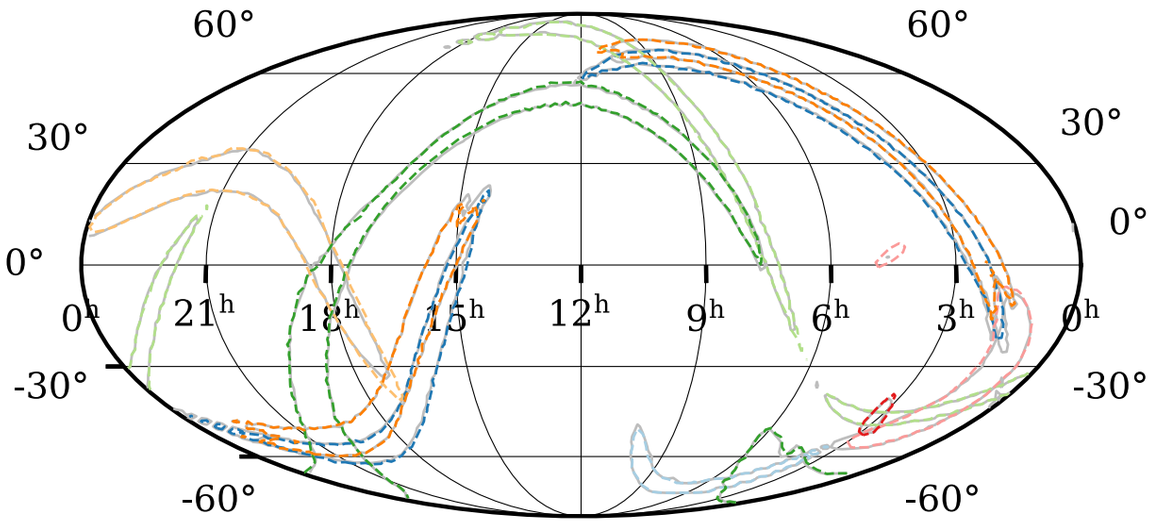


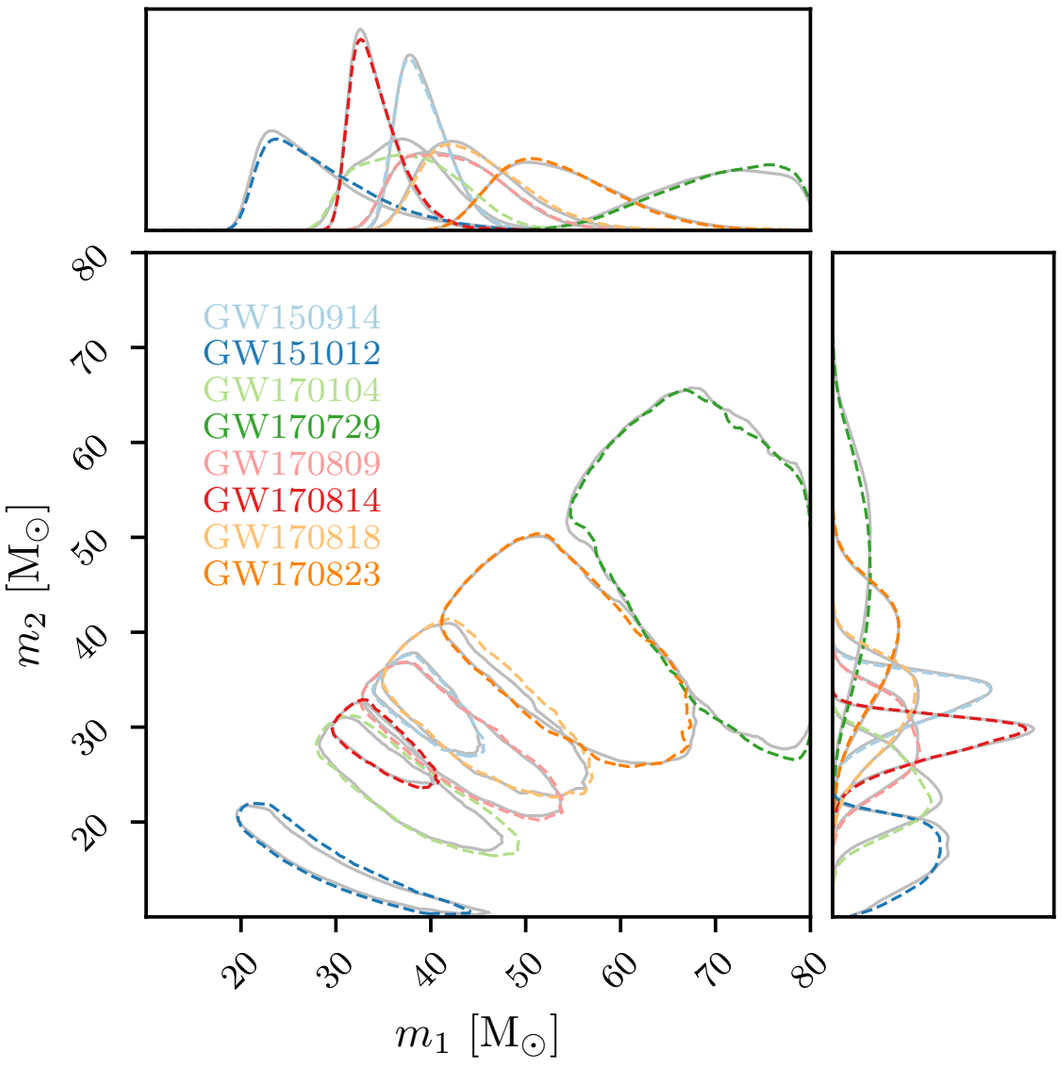
50,000 samples in ~ 20 s



Amortized inference

- Train once, analyze all events
- Account for detector nonstationarity from event to event

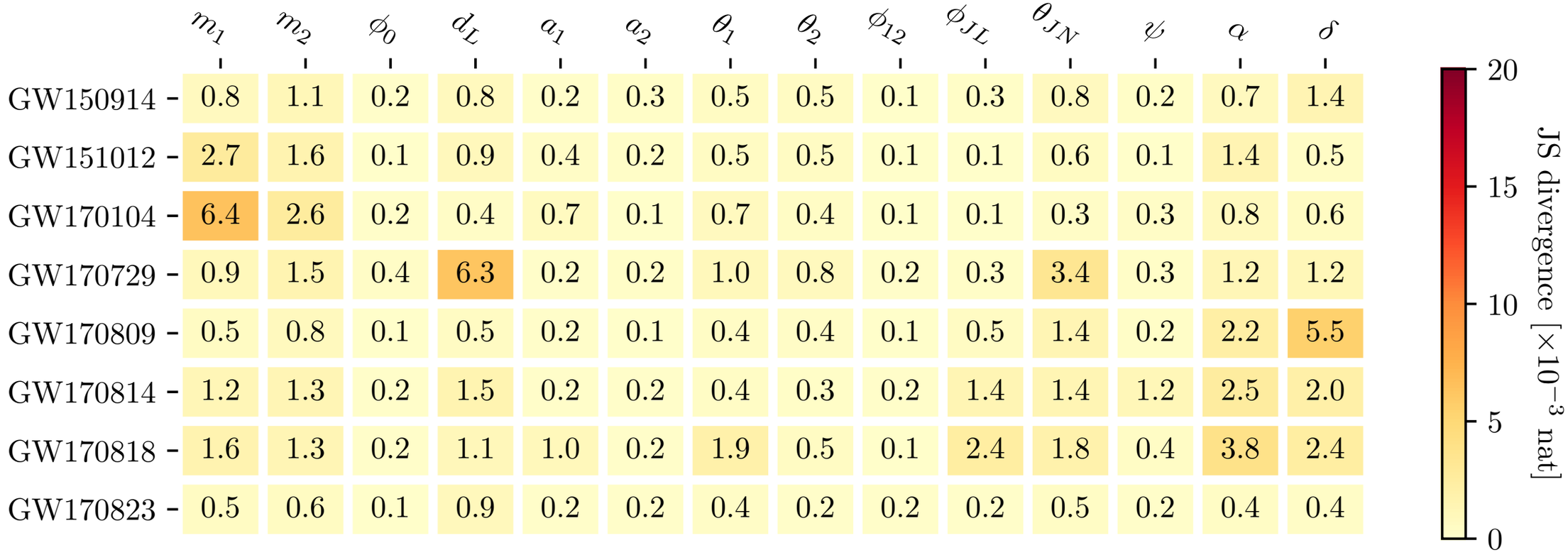




GWTC-1

Quantitative comparison vs standard sampler

- **JS divergence:** Measures a "distance" between probability distributions
- JSD $< 2 \times 10^{-3}$ nat "indistinguishable" (i.e., difference between LI runs)



Quantitative comparison vs standard sampler

- Median and 90% credible intervals

Event	$\mathcal{M}$ [$M_\odot$]	q	$\chi_{\rm eff}$	χ_p
GW150914	$31.0^{+1.5}_{-1.5}$	$0.85^{+0.13}_{-0.21}$	$-0.03^{+0.11}_{-0.12}$	$0.33^{+0.40}_{-0.27}$
	$31.1^{+1.5}_{-1.5}$	$0.85^{+0.13}_{-0.20}$	$-0.02^{+0.11}_{-0.12}$	$0.33^{+0.41}_{-0.27}$
GW151012	$18.2^{+1.1}_{-1.0}$	$0.60^{+0.35}_{-0.32}$	$0.01^{+0.20}_{-0.18}$	$0.30^{+0.40}_{-0.23}$
	$18.1^{+0.8}_{-0.7}$	$0.63^{+0.33}_{-0.34}$	$-0.00^{+0.21}_{-0.15}$	$0.30^{+0.40}_{-0.23}$
GW170104	$25.5^{+1.7}_{-1.8}$	$0.62^{+0.33}_{-0.23}$	$-0.06^{+0.16}_{-0.19}$	$0.39^{+0.34}_{-0.28}$
	$25.4^{+1.6}_{-1.6}$	$0.63^{+0.31}_{-0.22}$	$-0.07^{+0.15}_{-0.17}$	$0.38^{+0.34}_{-0.27}$
GW170729	$49.2^{+7.7}_{-8.1}$	$0.65^{+0.30}_{-0.24}$	$0.25^{+0.22}_{-0.26}$	$0.38^{+0.33}_{-0.26}$
	$49.6^{+7.6}_{-8.2}$	$0.68^{+0.28}_{-0.25}$	$0.27^{+0.22}_{-0.27}$	$0.38^{+0.32}_{-0.26}$
GW170809	$29.8^{+2.2}_{-1.9}$	$0.67^{+0.29}_{-0.24}$	$0.06^{+0.18}_{-0.16}$	$0.35^{+0.38}_{-0.27}$
	$29.9^{+2.1}_{-1.8}$	$0.68^{+0.28}_{-0.24}$	$0.07^{+0.17}_{-0.15}$	$0.35^{+0.38}_{-0.27}$
GW170814	$27.2^{+1.2}_{-1.2}$	$0.86^{+0.13}_{-0.24}$	$0.08^{+0.13}_{-0.12}$	$0.50^{+0.31}_{-0.38}$
	$27.1^{+1.1}_{-1.1}$	$0.86^{+0.13}_{-0.23}$	$0.08^{+0.12}_{-0.11}$	$0.52^{+0.29}_{-0.39}$
GW170818	$32.7^{+2.9}_{-2.8}$	$0.73^{+0.24}_{-0.27}$	$-0.02^{+0.21}_{-0.23}$	$0.51^{+0.30}_{-0.35}$
	$32.5^{+2.7}_{-2.6}$	$0.74^{+0.23}_{-0.27}$	$-0.05^{+0.20}_{-0.22}$	$0.53^{+0.28}_{-0.36}$
GW170823	$38.9^{+4.3}_{-4.1}$	$0.74^{+0.23}_{-0.28}$	$0.06^{+0.20}_{-0.20}$	$0.41^{+0.36}_{-0.31}$
	$38.9^{+4.3}_{-3.9}$	$0.73^{+0.24}_{-0.28}$	$0.06^{+0.20}_{-0.20}$	$0.41^{+0.36}_{-0.31}$

Summary comparison

	Likelihood-based inference	Neural posterior estimation
Requirements	Likelihood	**Simulated data**
Up-front costs (T = waveform generation time)	**None**	Data generation: ~ T Training: ~ 10 days
Inference time	Hours to weeks ~ T	**Seconds**
Sample quality	Correlated	**Independent**

Conclusions

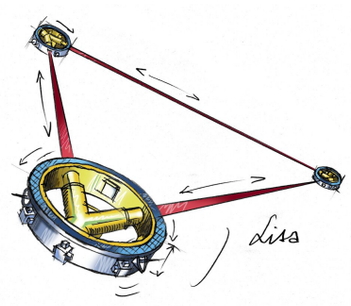
- Machine-learning algorithms can produce **full posterior distributions** including uncertainties.

- For generic BBH events, results are nearly **indistinguishable** from those of standard codes.
  - **Much faster:** ~ 20 s / event
  - Accounts for detector **noise nonstationarity** from event to event
  - User-friendly software under development (**“Dingo”**)

- Simulation-based inference is more flexible: Does **not** require a likelihood!

Outlook

- **LVK inference:**
  - Extensions to binary neutron stars, models with precession and higher radiation multipoles.
  - Real detector noise (nonstationary, non-Gaussian)
- **LISA:** Can this be adapted to disentangle overlapping events?



- **Populations:** Can simulation-based inference be used to speed up analyses? Are there problems involving intractable likelihoods?
  - Happy to discuss!

Thank You!