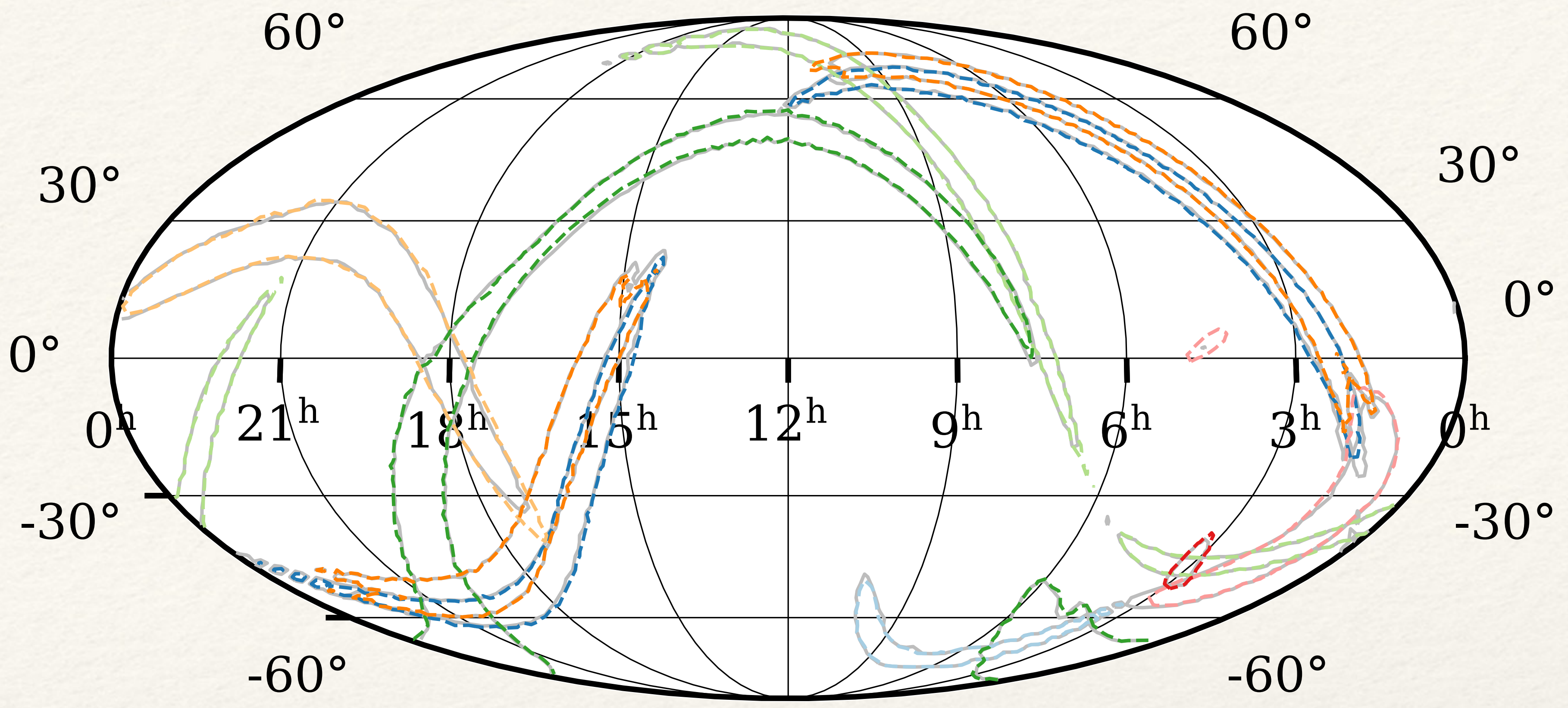


Bayesian Deep Learning for Cosmology and Time-Domain Astrophysics
June 22, 2022

Simulation-based inference for gravitational waves

Stephen R. Green

Max Planck Institute
for Gravitational Physics

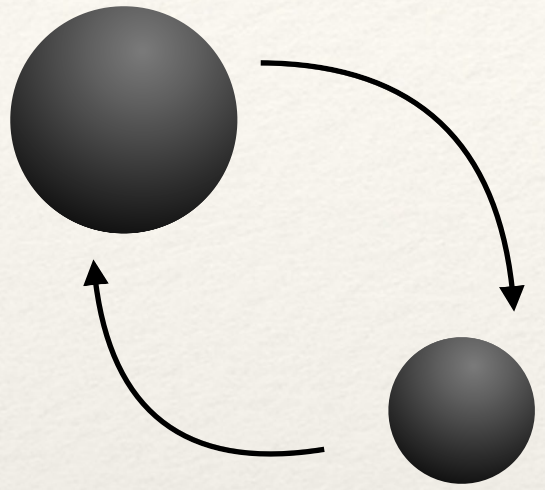
with
Maximilian Dax
Jonathan Gair
Jakob Macke
Alessandra Buonanno
Bernhard Schölkopf

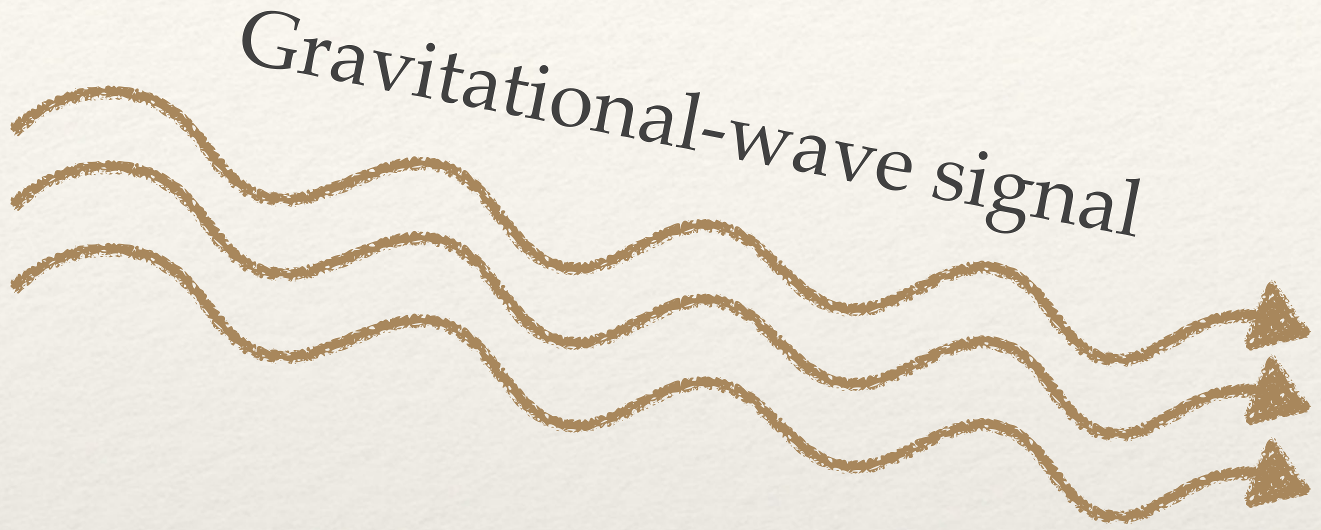
Outline

- Gravitational-waves: Why use Bayesian deep learning?
- Neural posterior estimation
- Using symmetries to simplify data
- Validating / improving results with importance sampling

Gravitational-wave astronomy

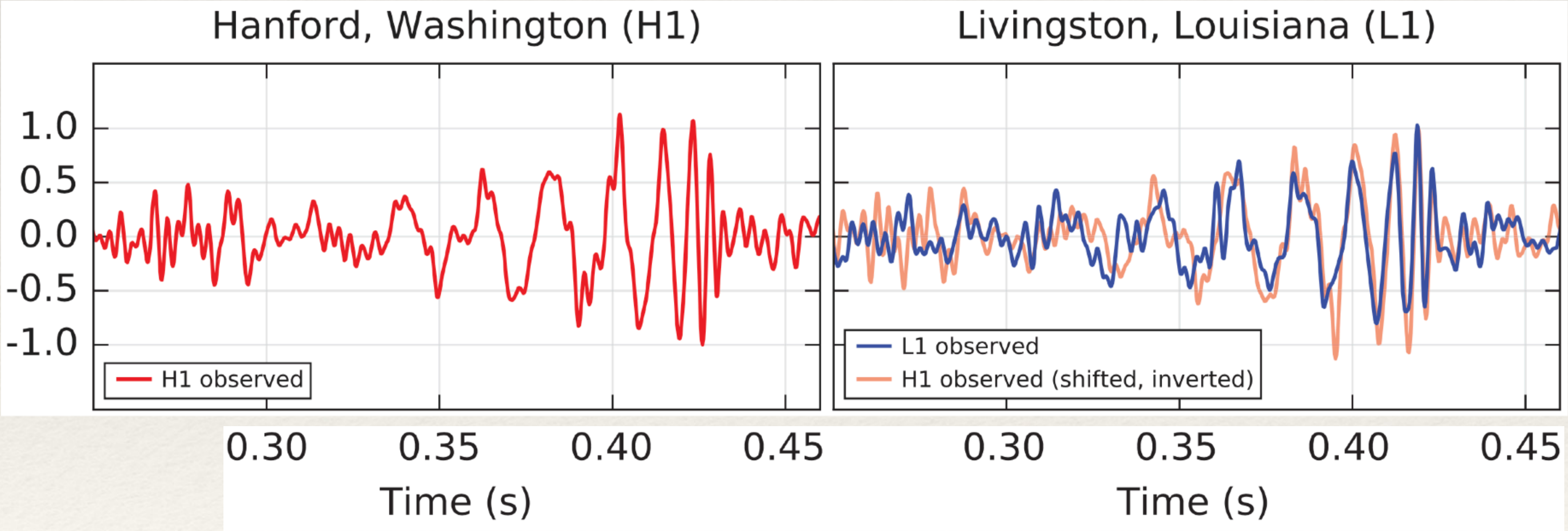
Merging black hole and/or neutron star binaries

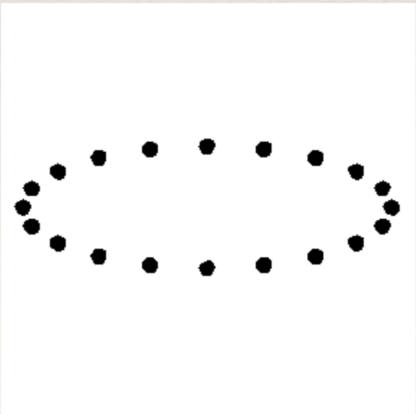


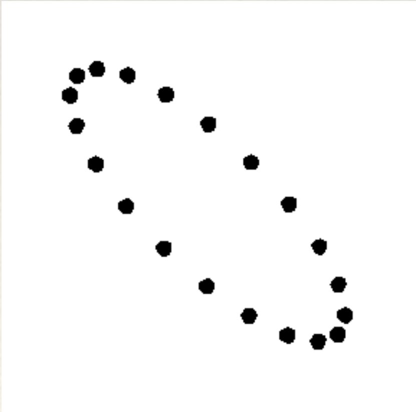


Gravitational-wave signal

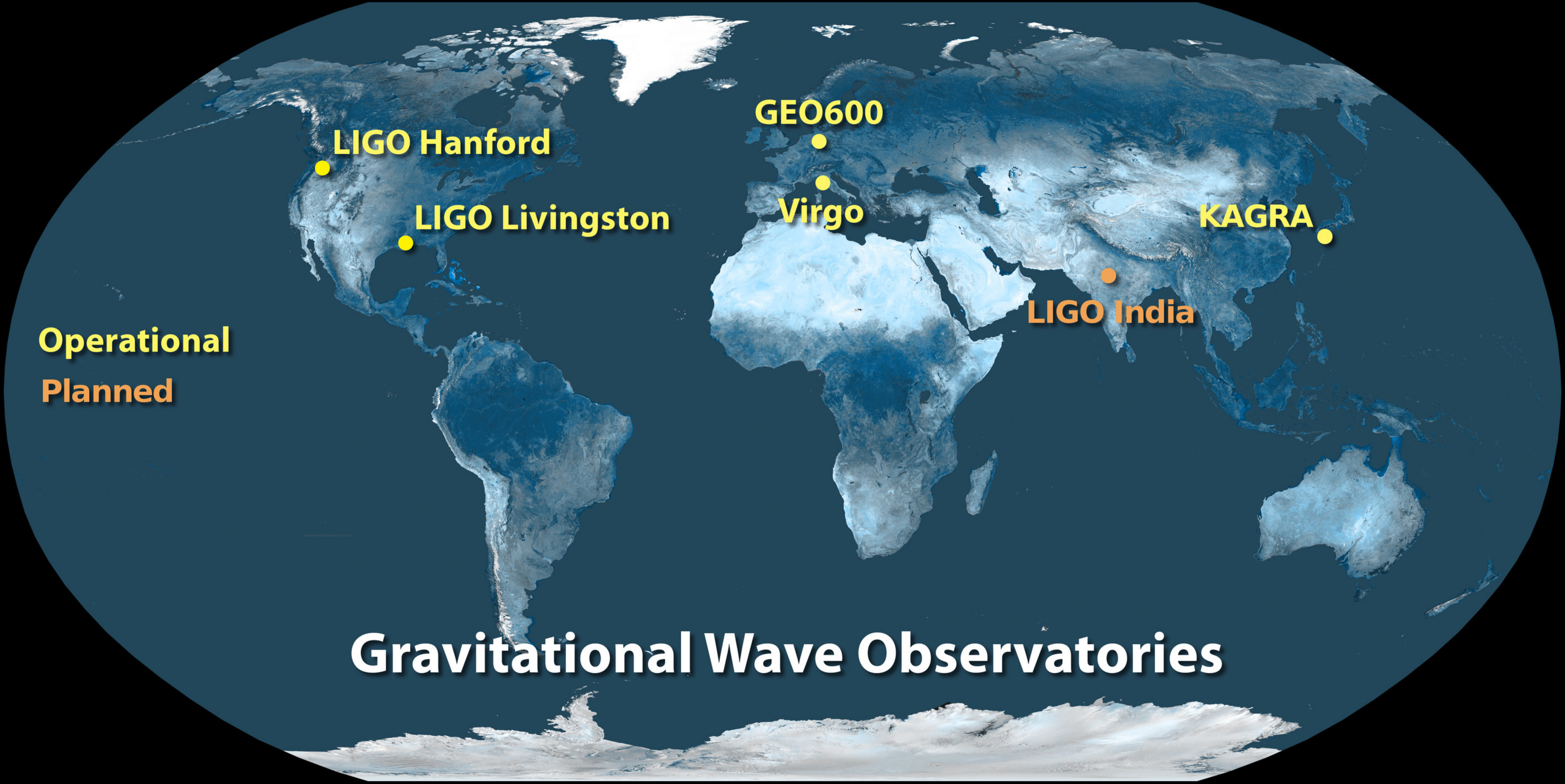
LIGO and Virgo observatories







Gravitational Wave Observatories



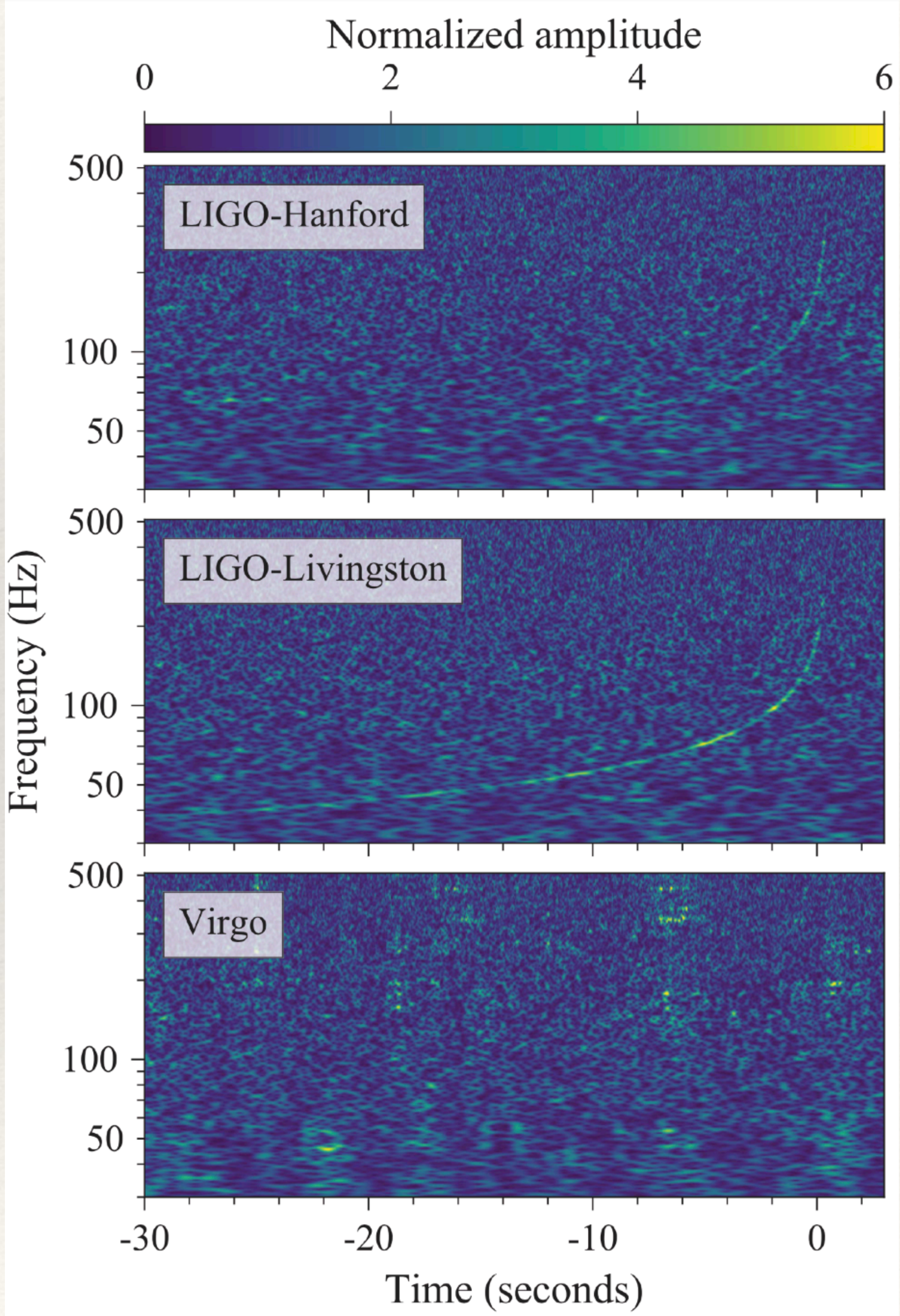
Masses in the Stellar Graveyard

From these results, fit Bayesian Hierarchical Models to study populations, cosmology

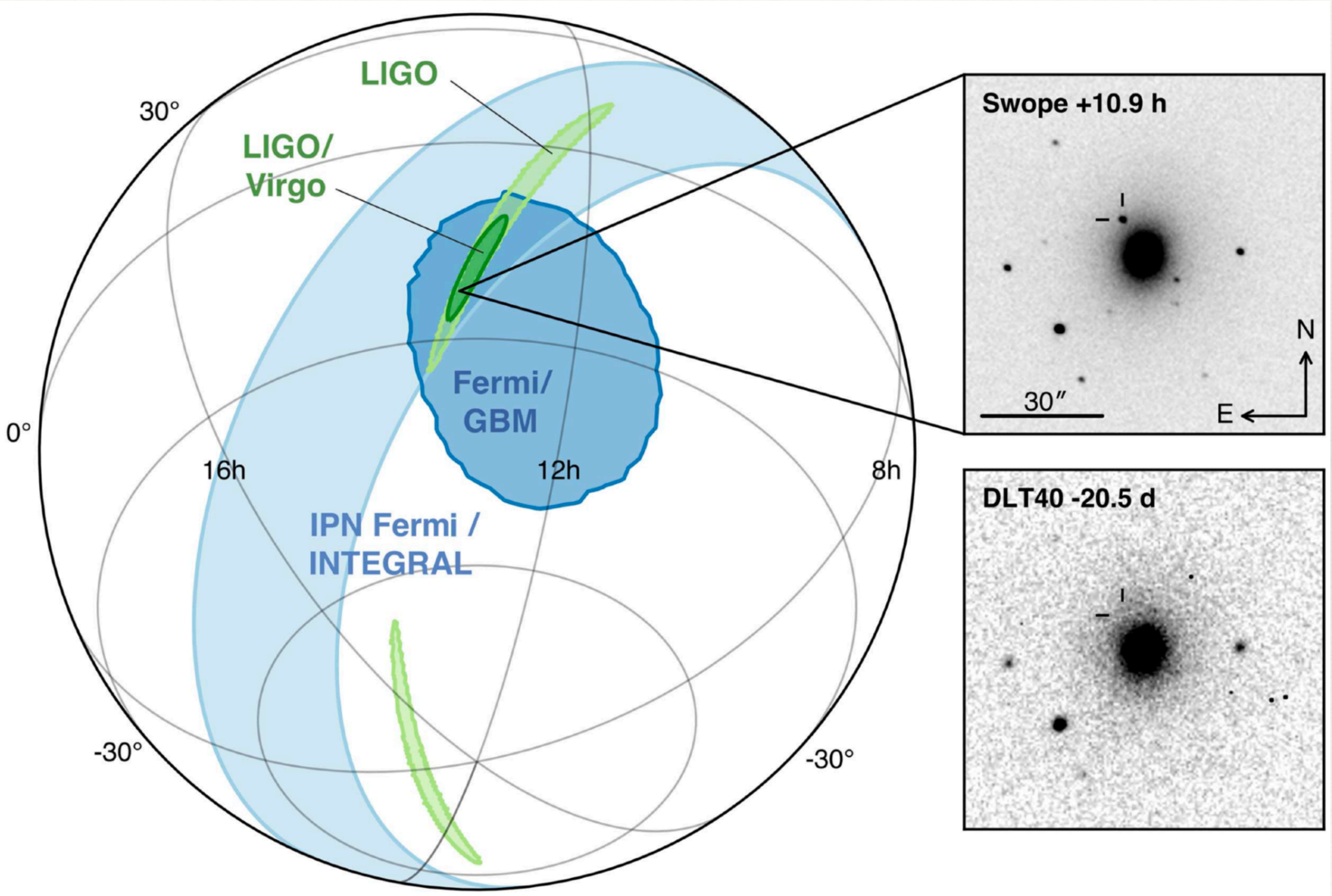
LIGO-Virgo-KAGRA | Aaron Geller | Northwestern

Binary neutron star mergers

- GW170817: Sky localization enables multimessenger astrophysics



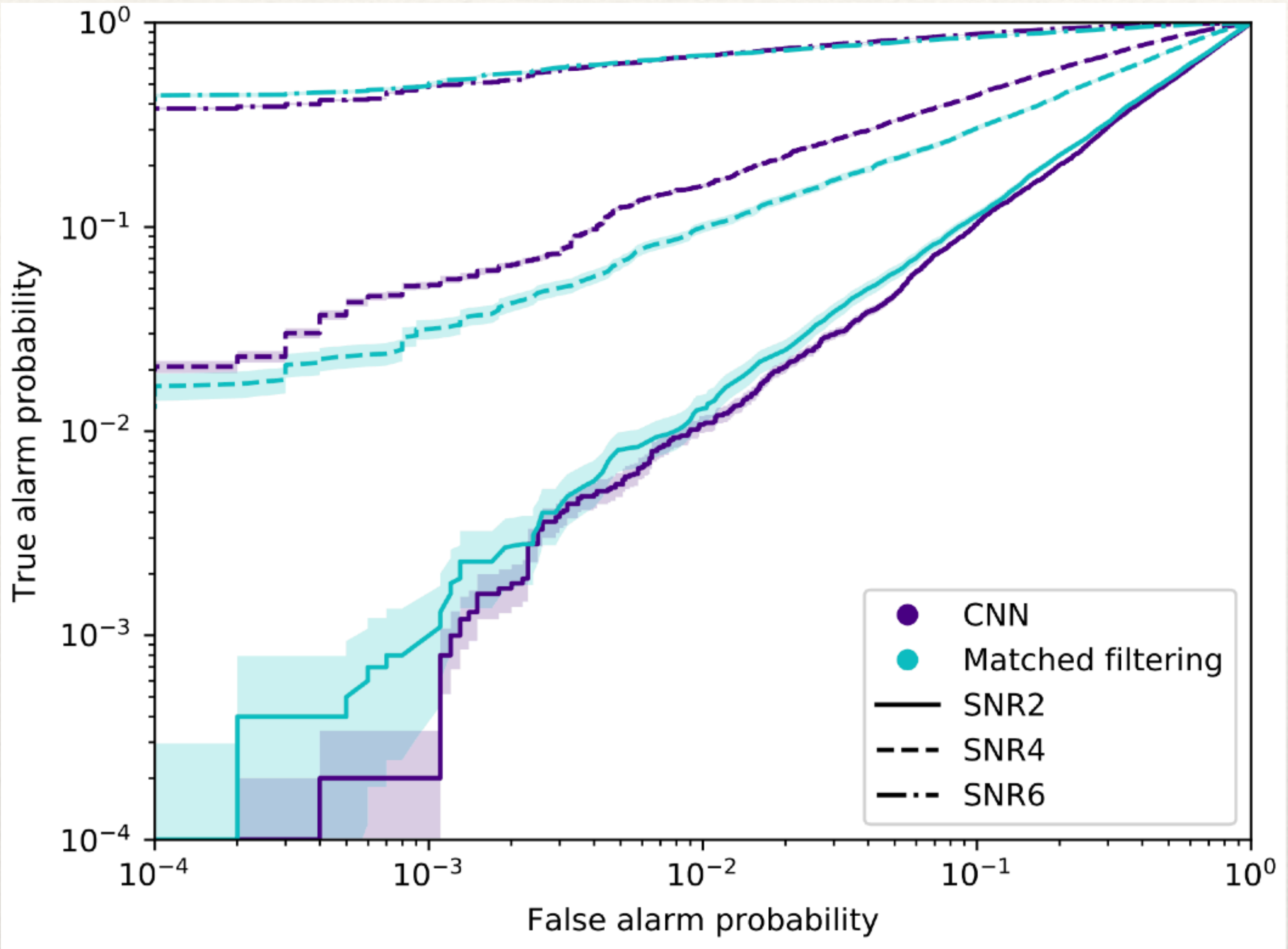
PRL 119, 161101 (2017)



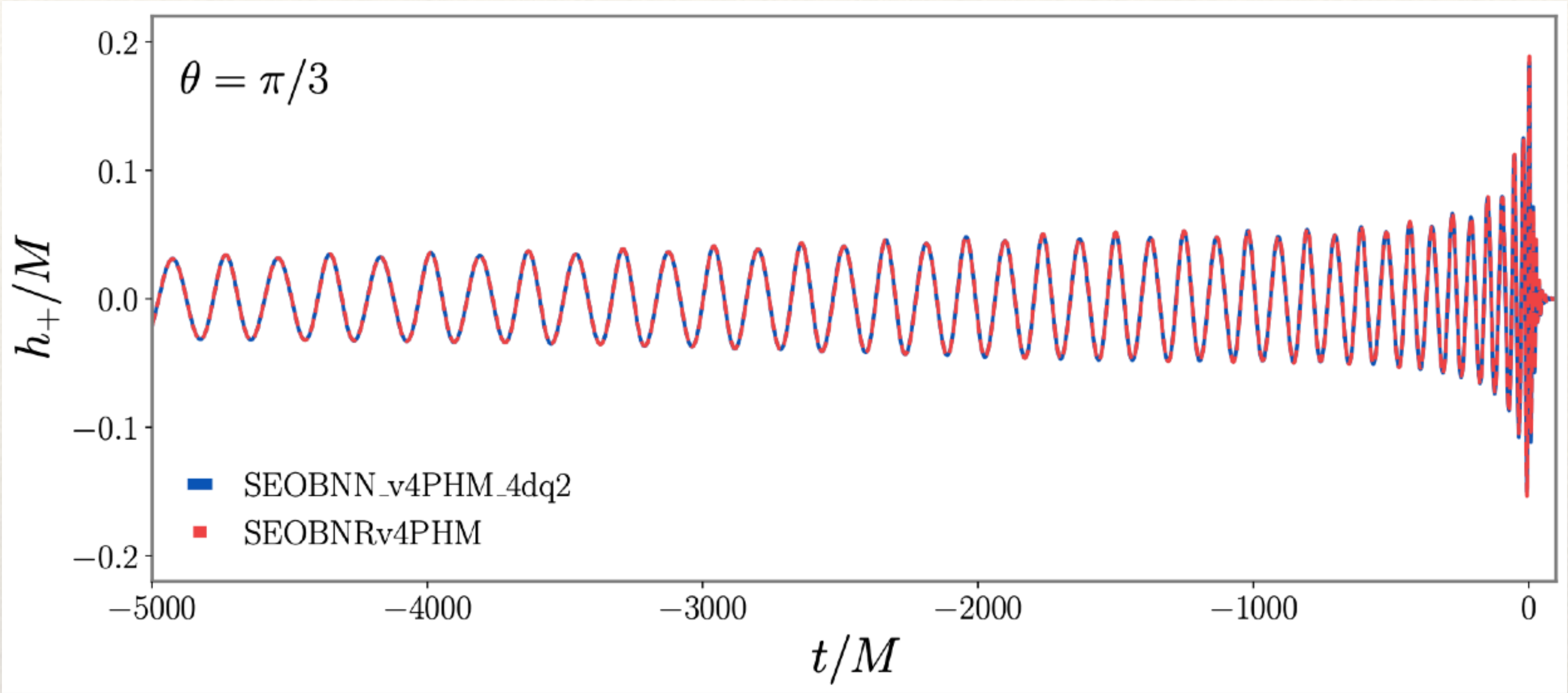
ApJ Lett 848:L12 (2017)

Applications of deep learning in gravitational waves

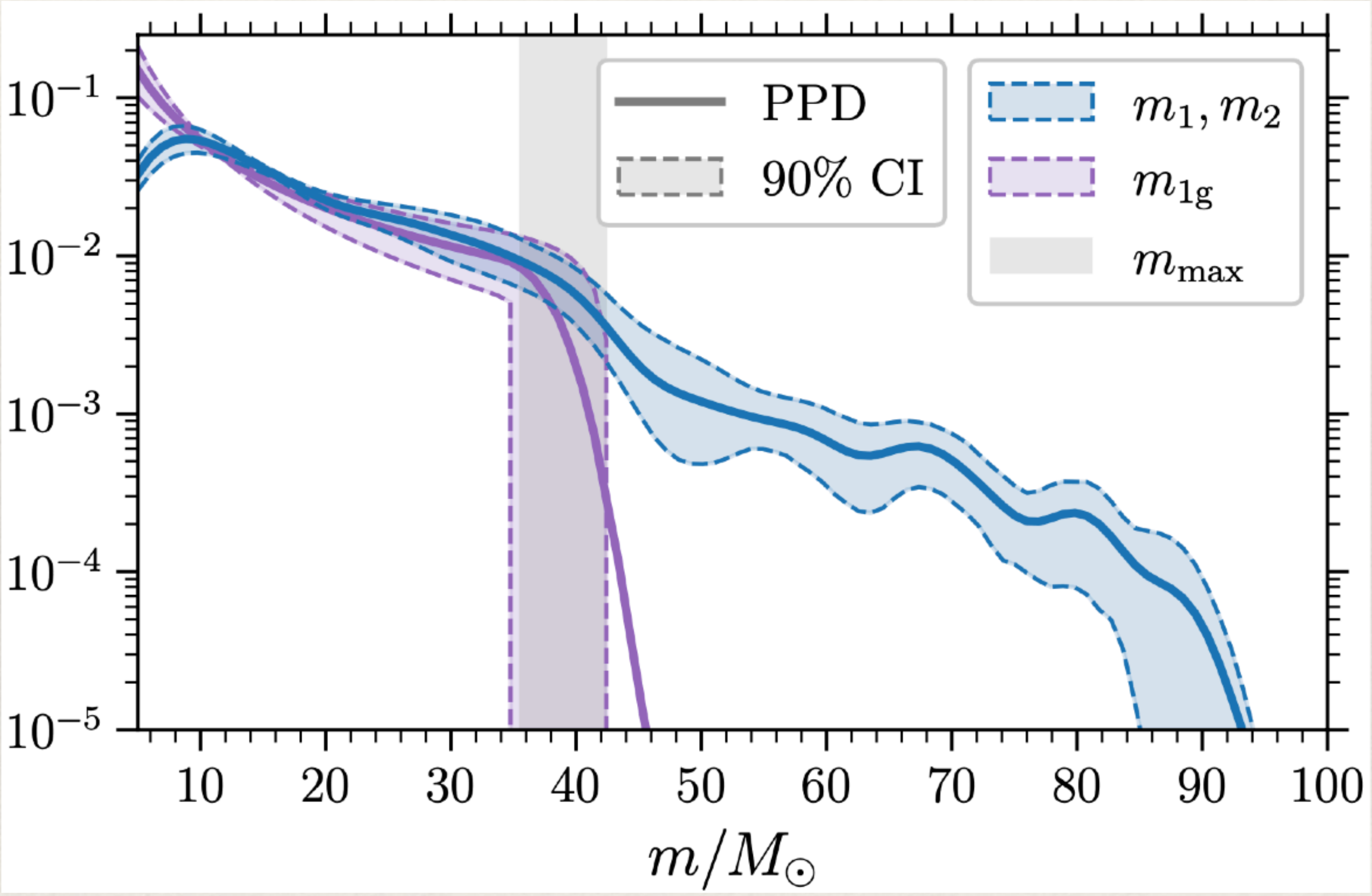
Signal detection



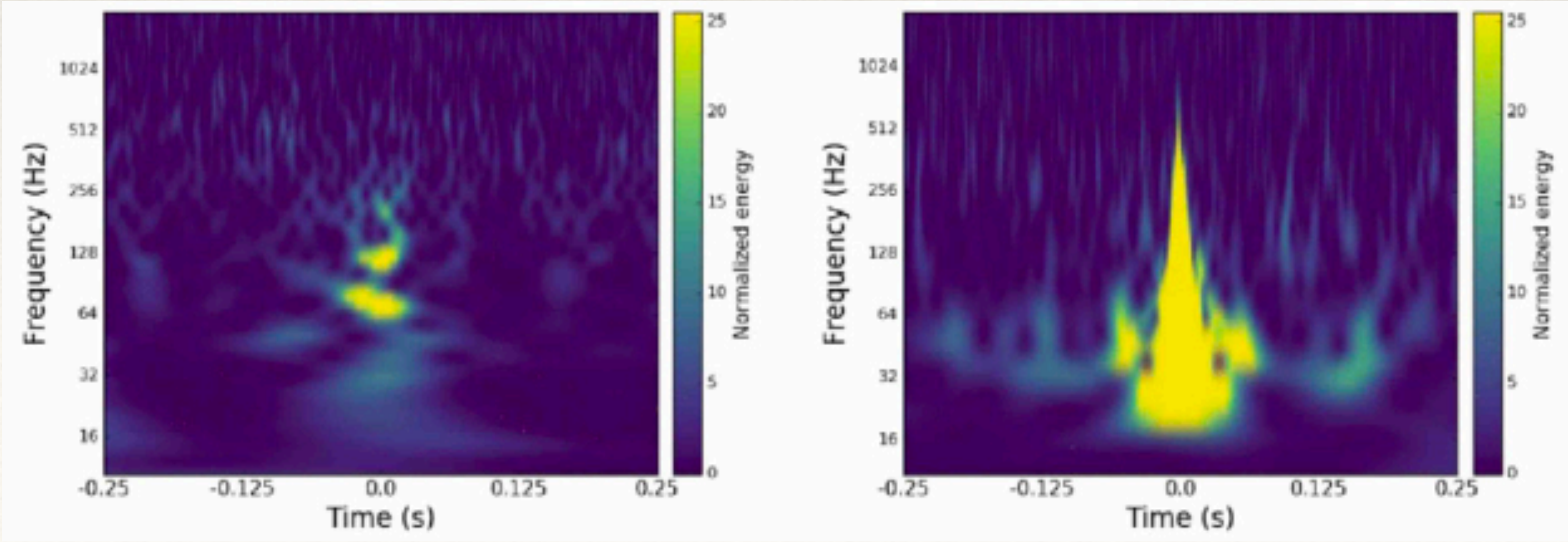
Waveform modeling



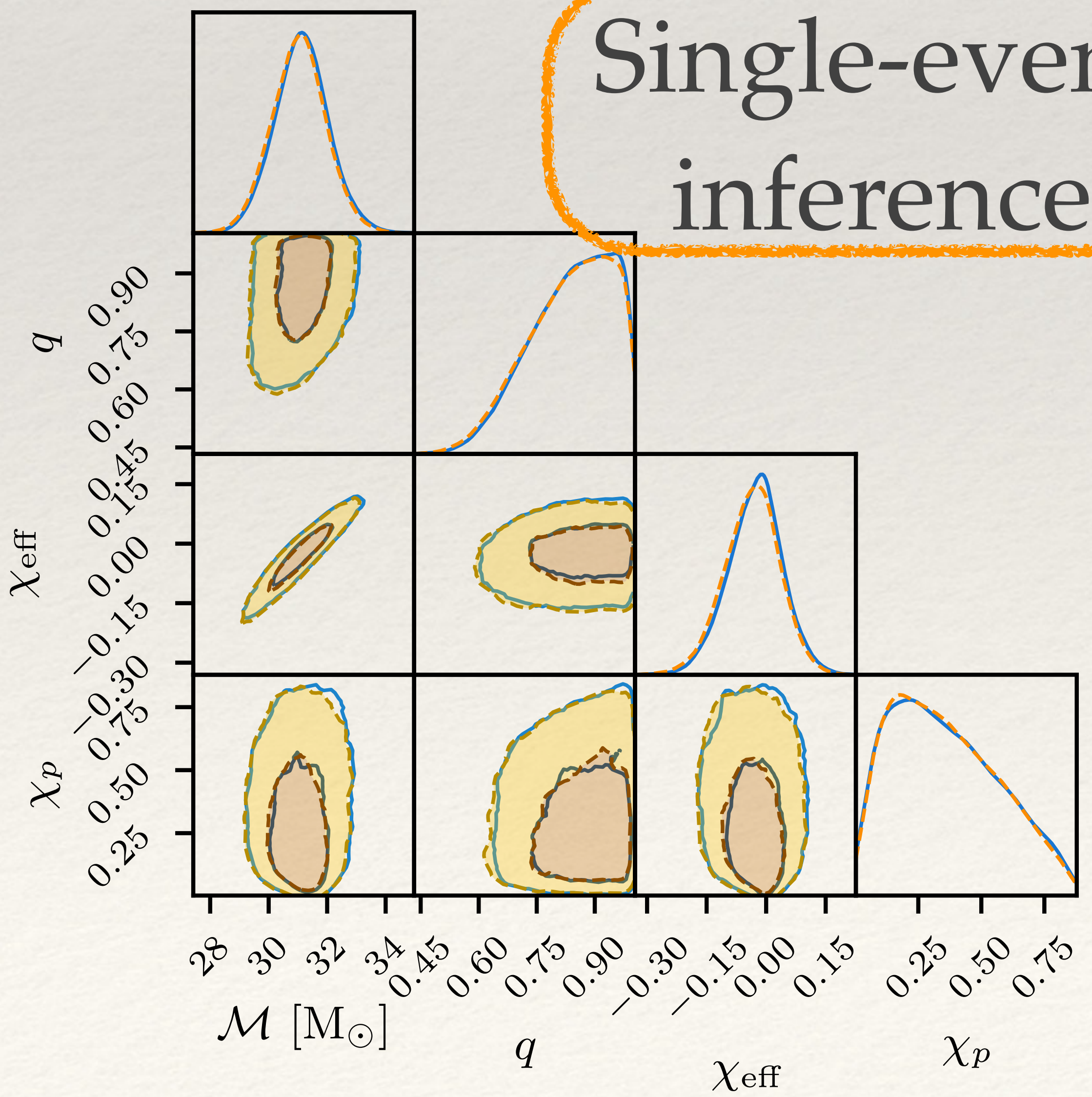
Population inference



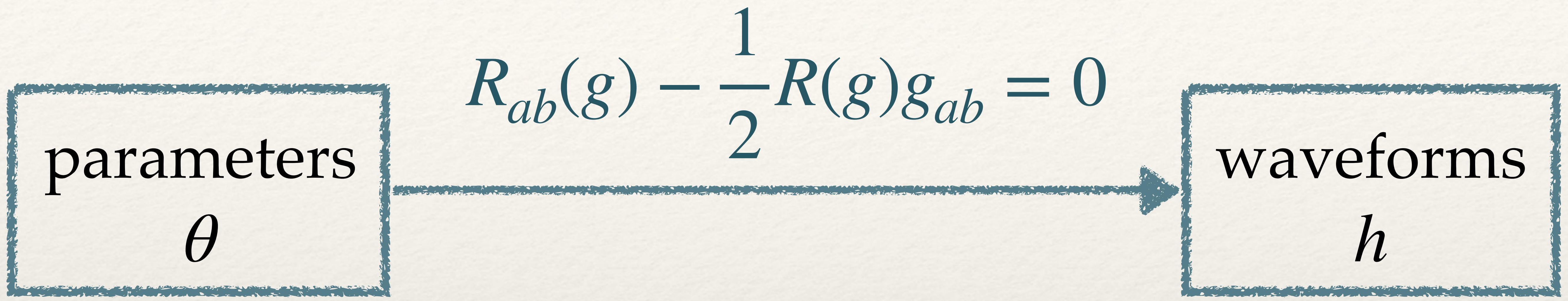
Glitch classification



Single-event inference



Binary black hole signal models



parameters θ

$$R_{ab}(g) - \frac{1}{2}R(g)g_{ab} = 0$$

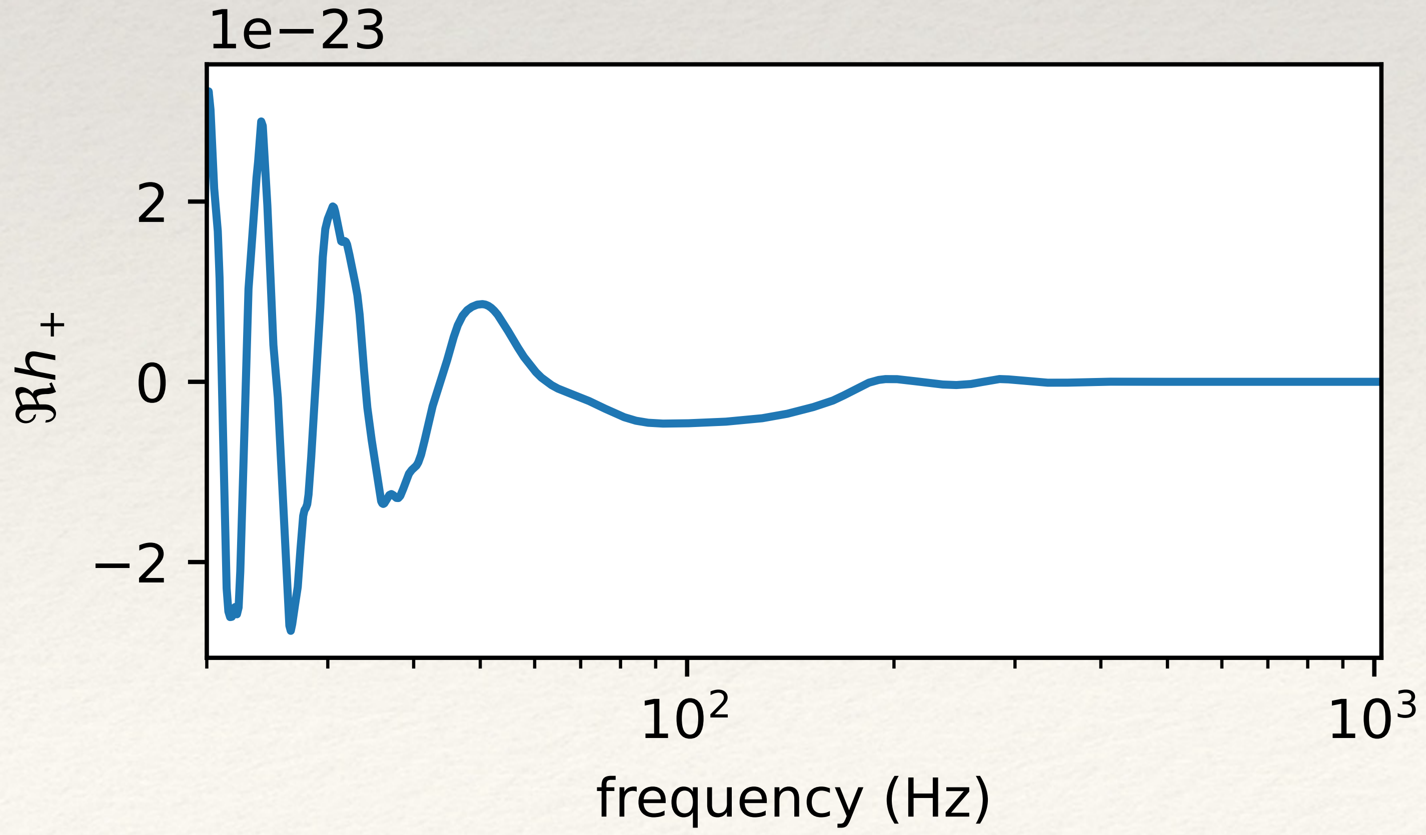
waveforms h

15 for binary black holes

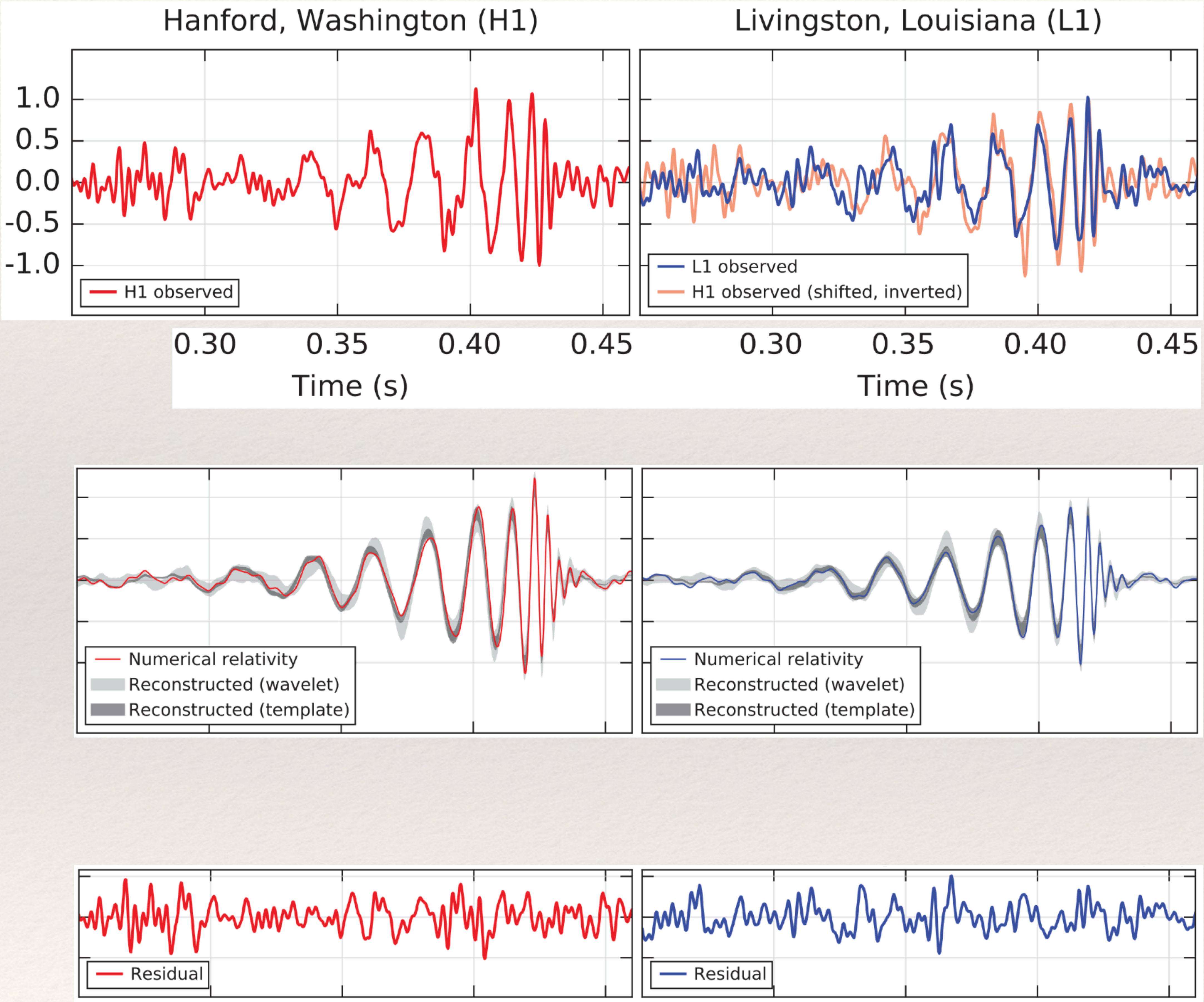
- masses (m_1, m_2)
- spins $(a_1, a_2, \theta_1, \theta_2, \phi_{12}, \phi_{JL})$
- distance d_L
- sky position (α, δ)
- coalescence time t_c
- reference phase ϕ_c
- polarization ψ

$\approx 10^4 - 10^5$ **dimensions**

- $f \in [20, 1024]$ Hz
- $T = 8$ s
- 2 or 3 detectors



Data are noisy



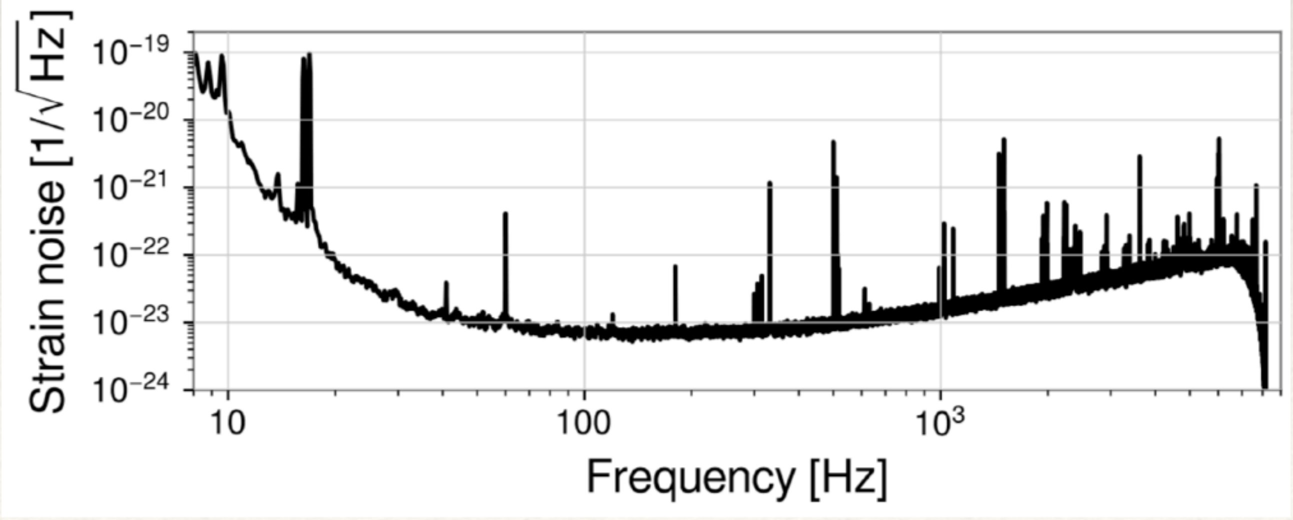
data

=

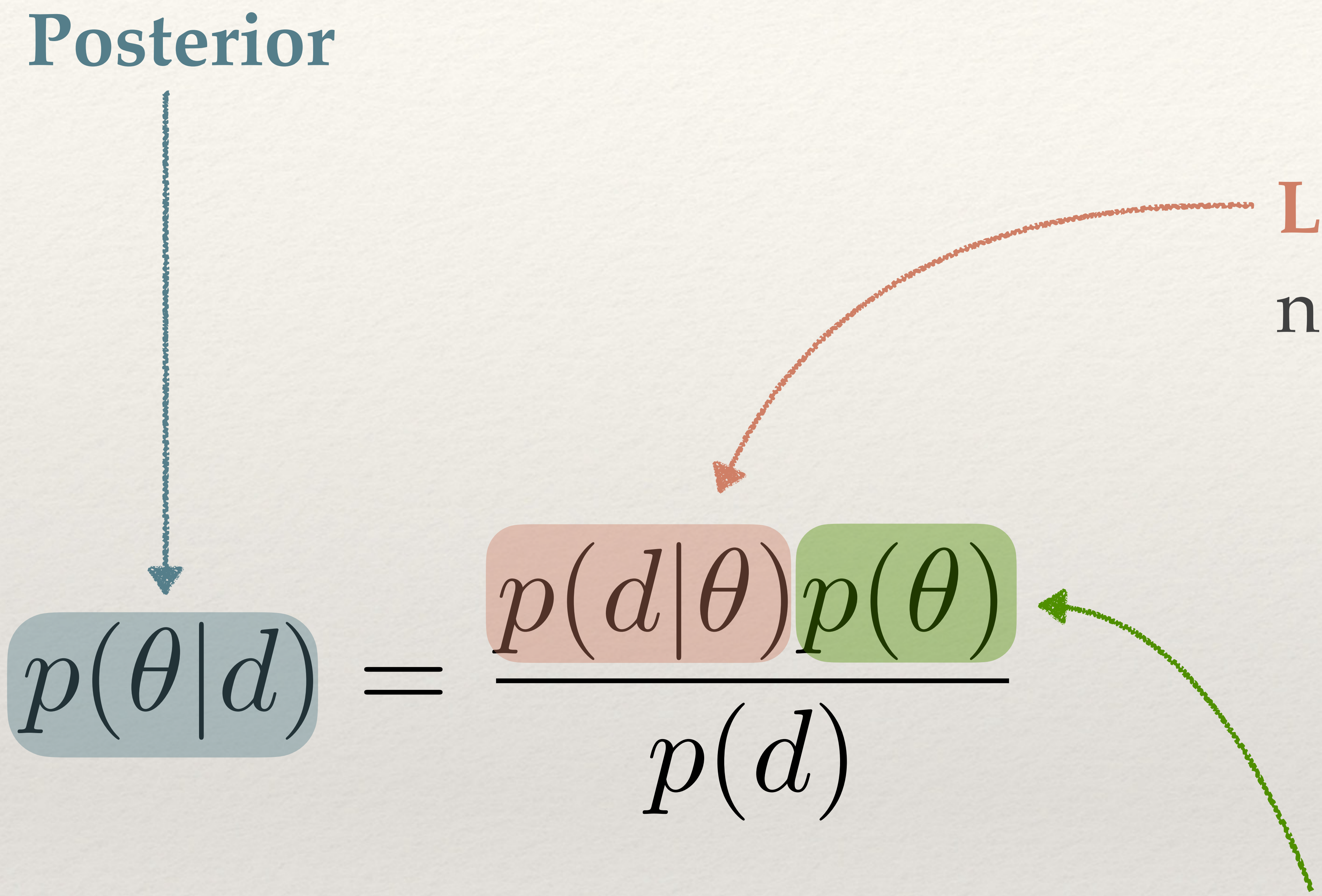
signal $h(\theta)$

+

noise



Model



Posterior

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

Likelihood assumes **stationary Gaussian** detector noise

$$p(d\,|\,\theta) = \mathcal{N}(h_I(\theta), S_{\mathrm{n},I})$$

waveform model

power spectral density

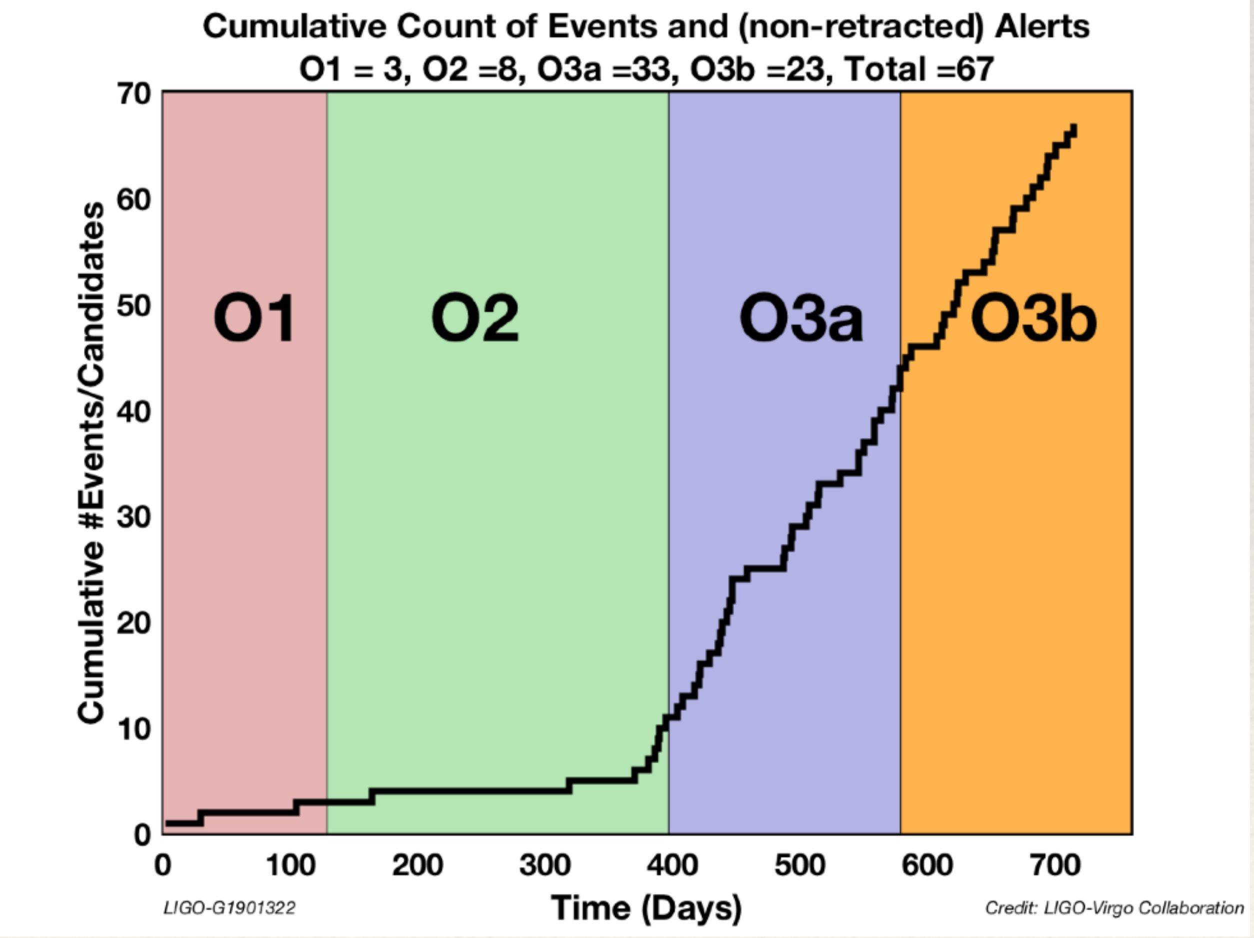
Prior

e.g., uniform in masses, spins

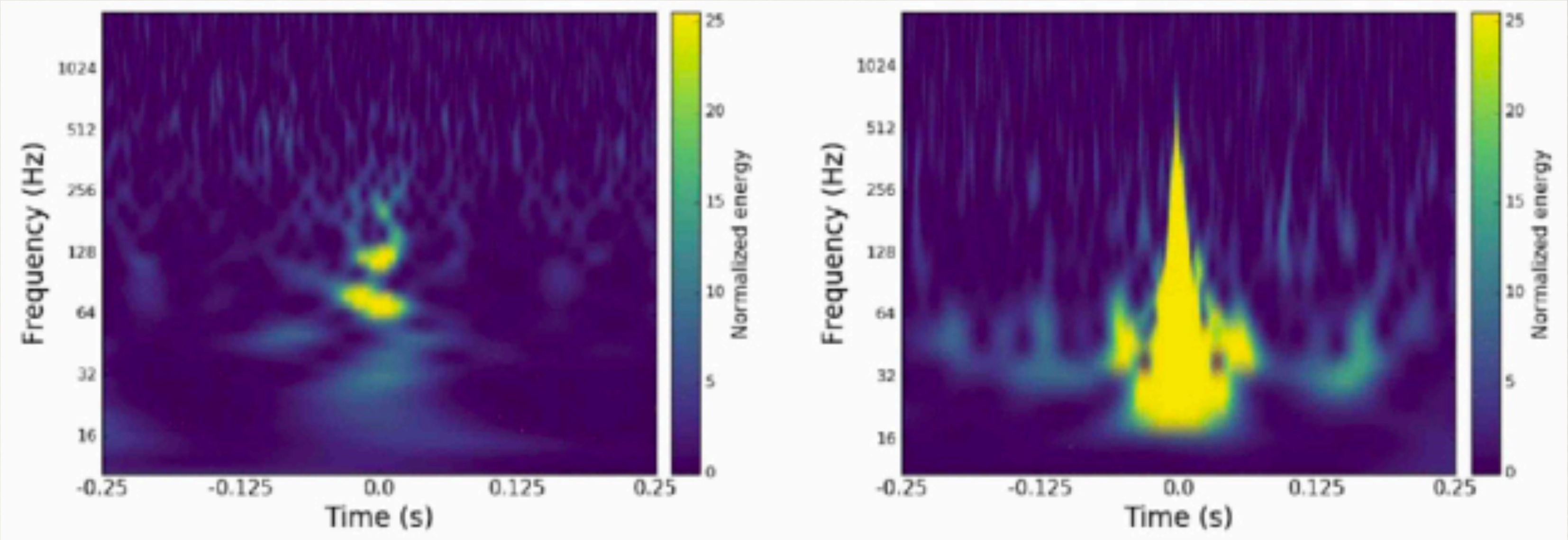
uniformly sky position and orientation

Why deep learning for inference?

- **Speed / amortization**
  - Model evaluation is expensive, with classical inference taking hours to weeks.
  - Rapid electromagnetic alerts
  - Large number of events



- **Likelihood-free**
  - The stationary-Gaussian likelihood is an approximation.
  - Noise transients (glitches) could be included.



  - Train with real noise.

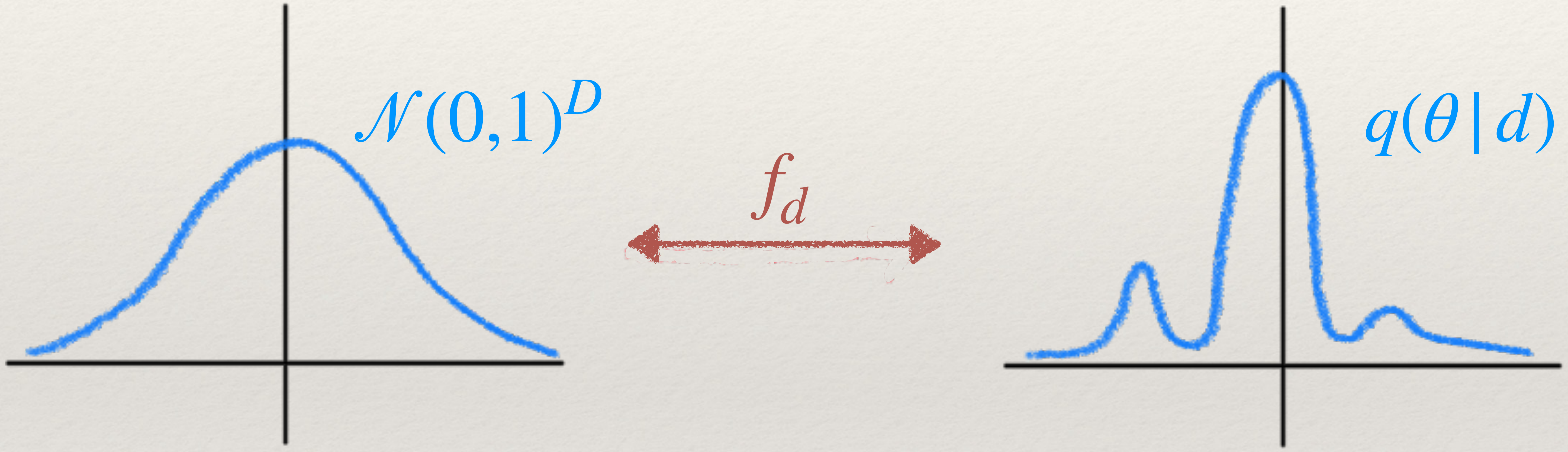
This talk

Outline

- Gravitational-waves: Why use Bayesian deep learning?
- Neural posterior estimation
- Using symmetries to simplify data
- Validating / improving results with importance sampling

Neural posterior estimation

- Train a **conditional normalizing flow** to approximate the Bayesian posterior.



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

- **Properties of** f_d
    1. invertible
    2. simple Jacobian determinant

$q(\theta \,|\, d)$ has **fast sampling and density evaluation**

Simulation-based inference

- Loss function

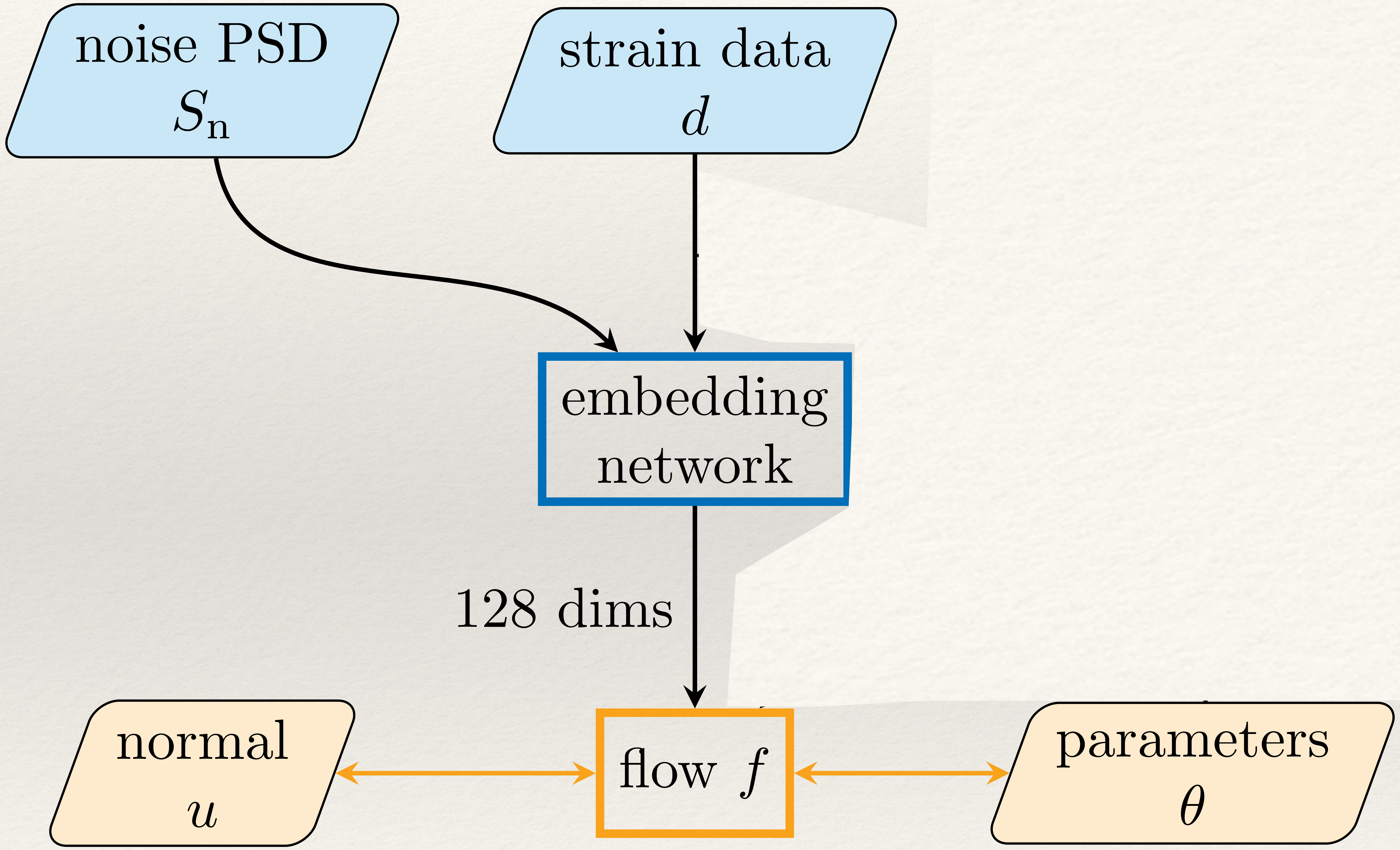
$$
\begin{aligned}
L &= \mathbb{E}_{p(d)} D_{\mathrm{KL}}(p\|q) \\
&= \int dd\, p(d) \int d\theta\, p(\theta|d) \log \frac{p(\theta|d)}{q(\theta|d)} \\
&\simeq \int d\theta\, p(\theta) \int dd\, p(d|\theta) \left[-\log q(\theta|d)\right] \\
&\approx \sum_{\substack{\theta^{(i)}\sim p(\theta) \\ d^{(i)}\sim p(d|\theta^{(i)})}} -\log q(\theta^{(i)}|d^{(i)})
\end{aligned}
$$

Generate training data by
(1) sampling from the prior,
(2) simulating a signal + noise.

Neural Posterior Estimation Network

Dax, SRG+ (PRL 2021)

Detector noise characteristics vary from event to event



noise PSD S_n

strain data d

embedding network

128 dims

normal u

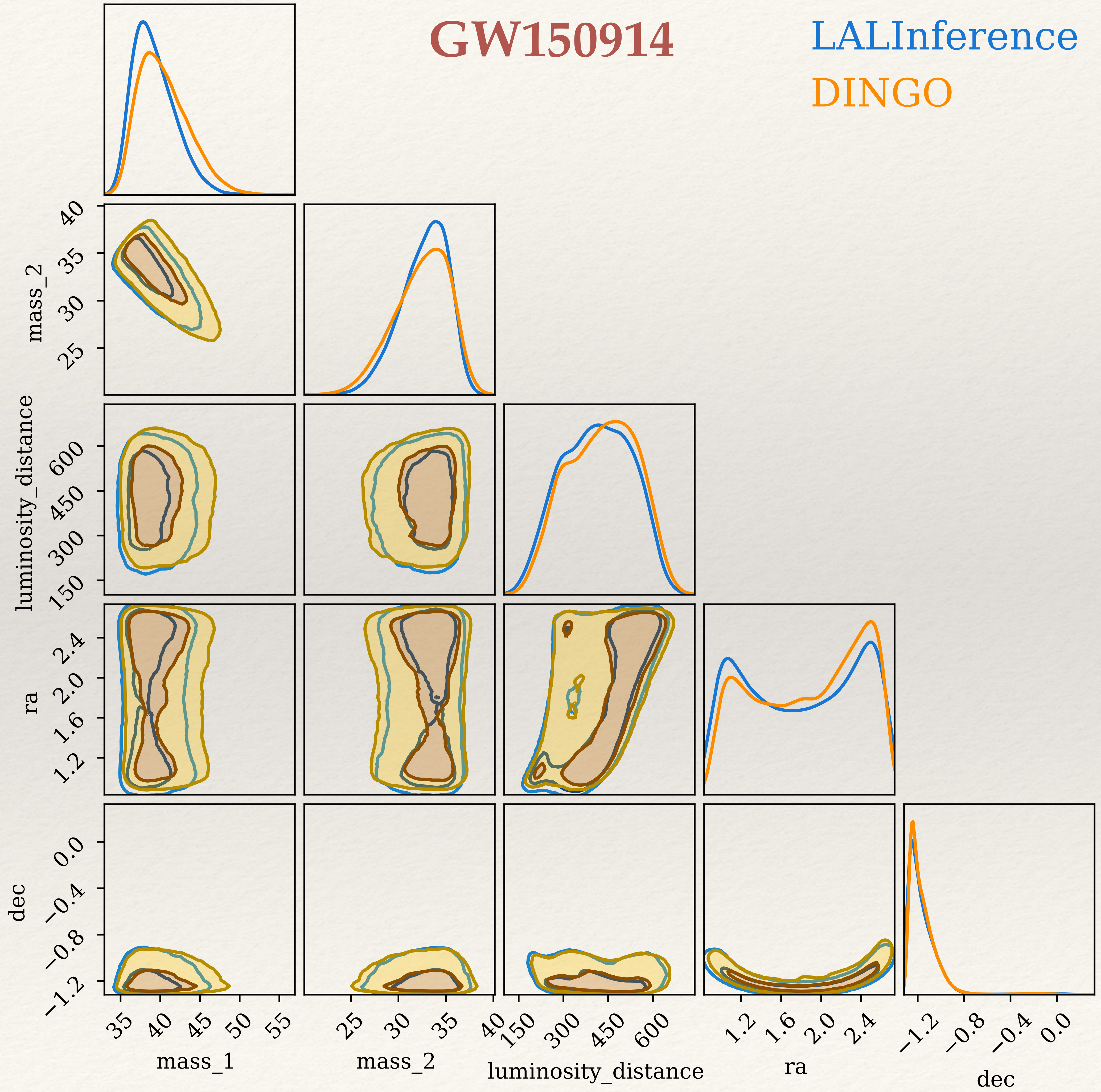
flow f

parameters θ

(spline)

Results

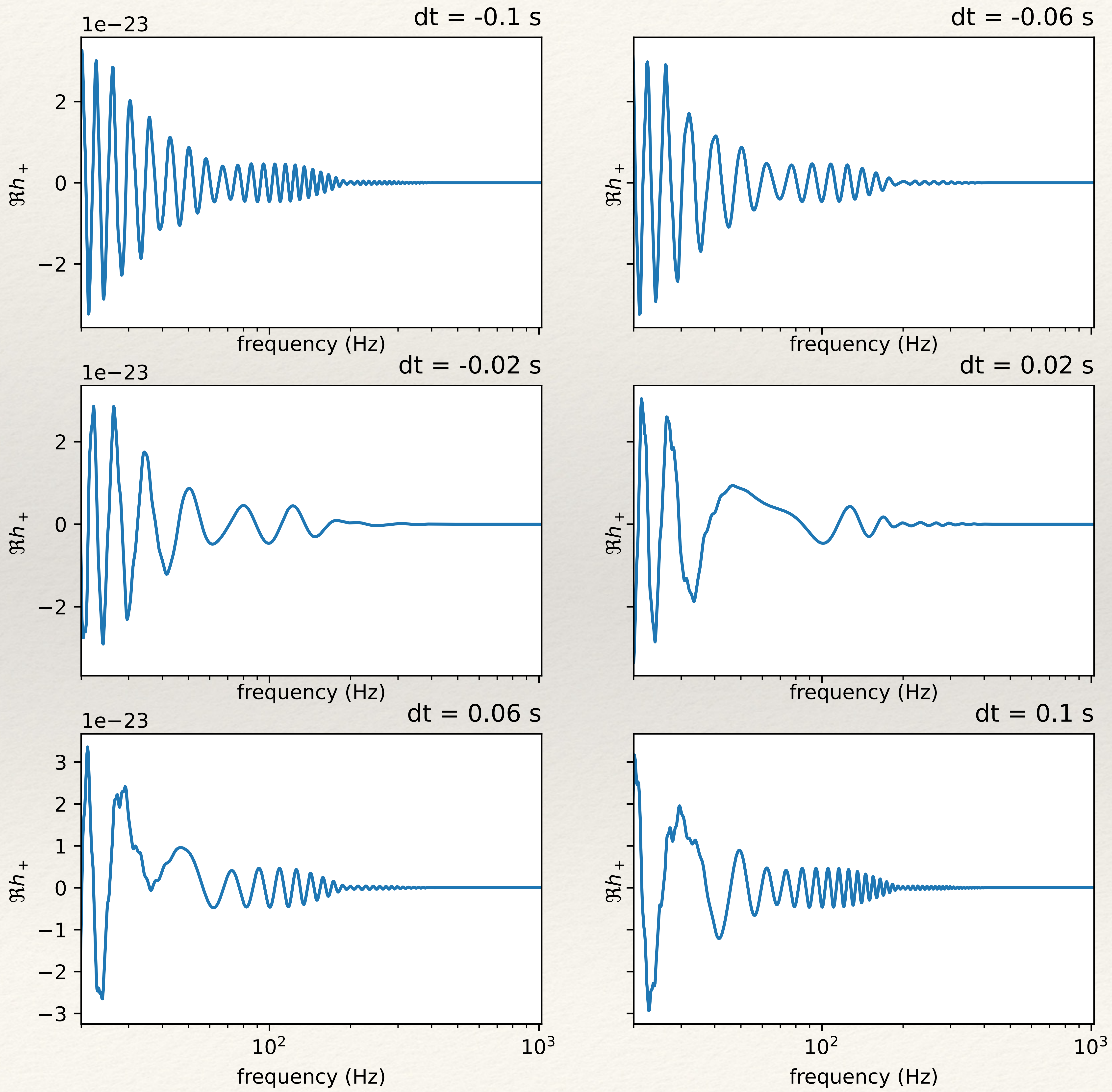
- Train for a week…
- Results are not bad… but clear deviations from MCMC.



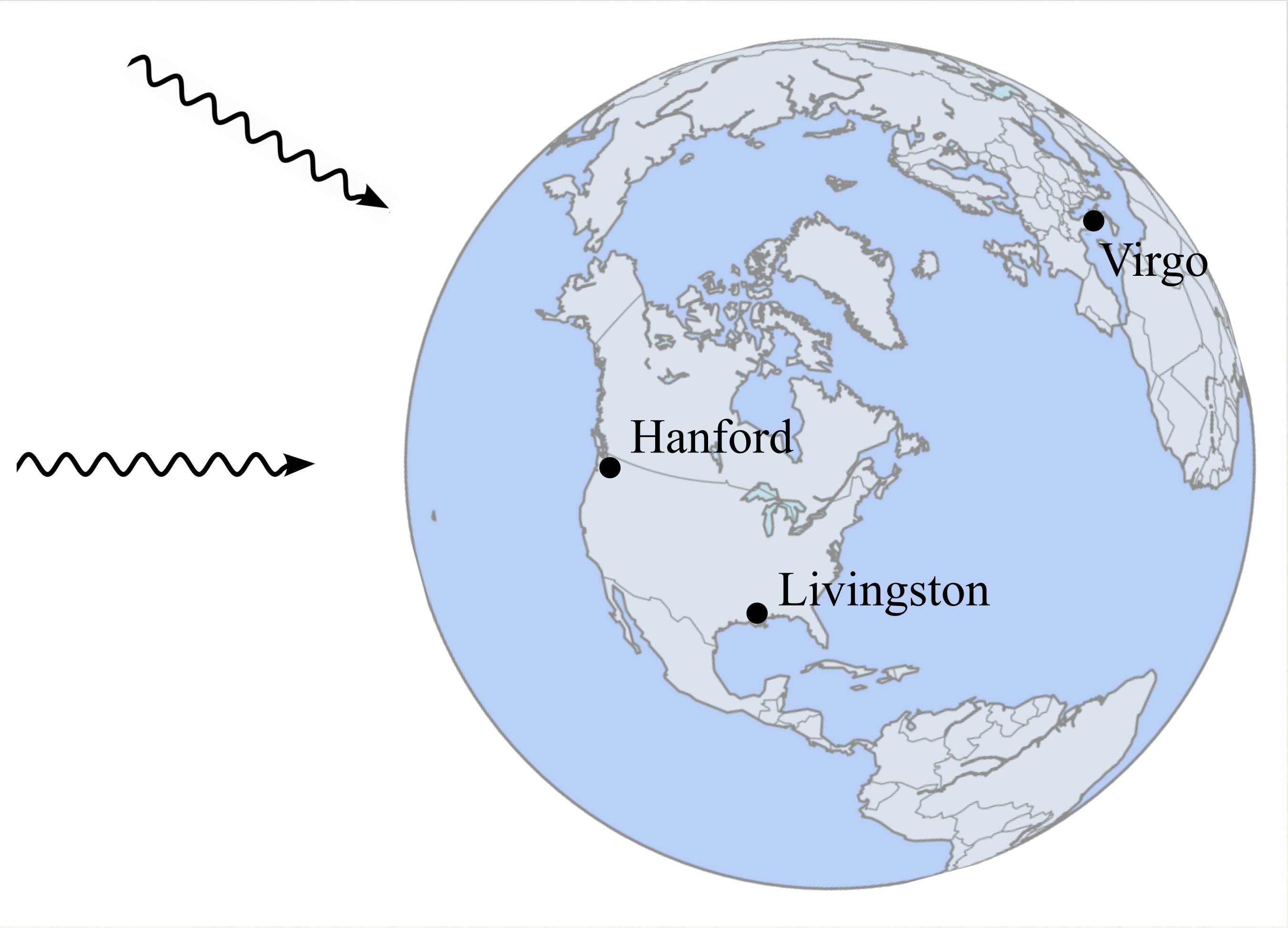
Outline

- Gravitational-waves: Why use Bayesian deep learning?
- Neural posterior estimation
- Using symmetries to simplify data
- Validating / improving results with importance sampling

Time shifted signals



- Frequency-domain data
  - Corresponds to multiplication by $e^{-2\pi i f \delta t}$
- Variation in sky position + overall coalescence time
  - Time shifts in each detector $\delta t \approx 0.1$ s



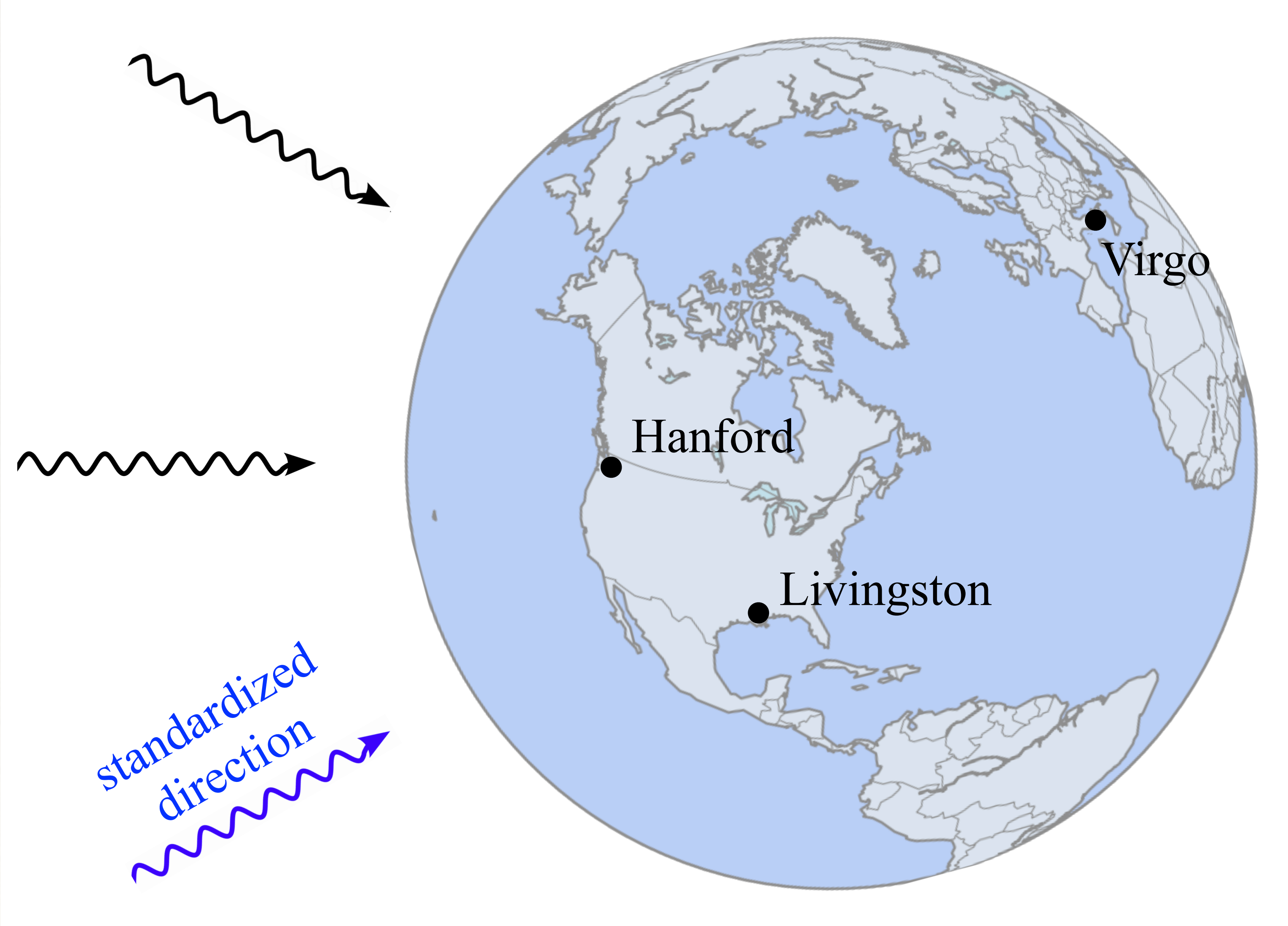
Hard to learn!

Equivariance in deep learning

- **Want to enforce a covariant posterior**

$$\boxed{p(t_c \,|\, d) = p(t_c + \delta t \,|\, d * e^{-2\pi i f \delta t})}$$

- **Equivariant architectures** commonly used to incorporate symmetries:
  - E.g., **convolutional network** (translational symmetry)
    — cannot propagate equivariance through normalizing flow
  - E.g., **equivariant flows**
    — not suitable for joint transformation of data and parameters
- **Our approach: Iterative transformation to "standardized" data**



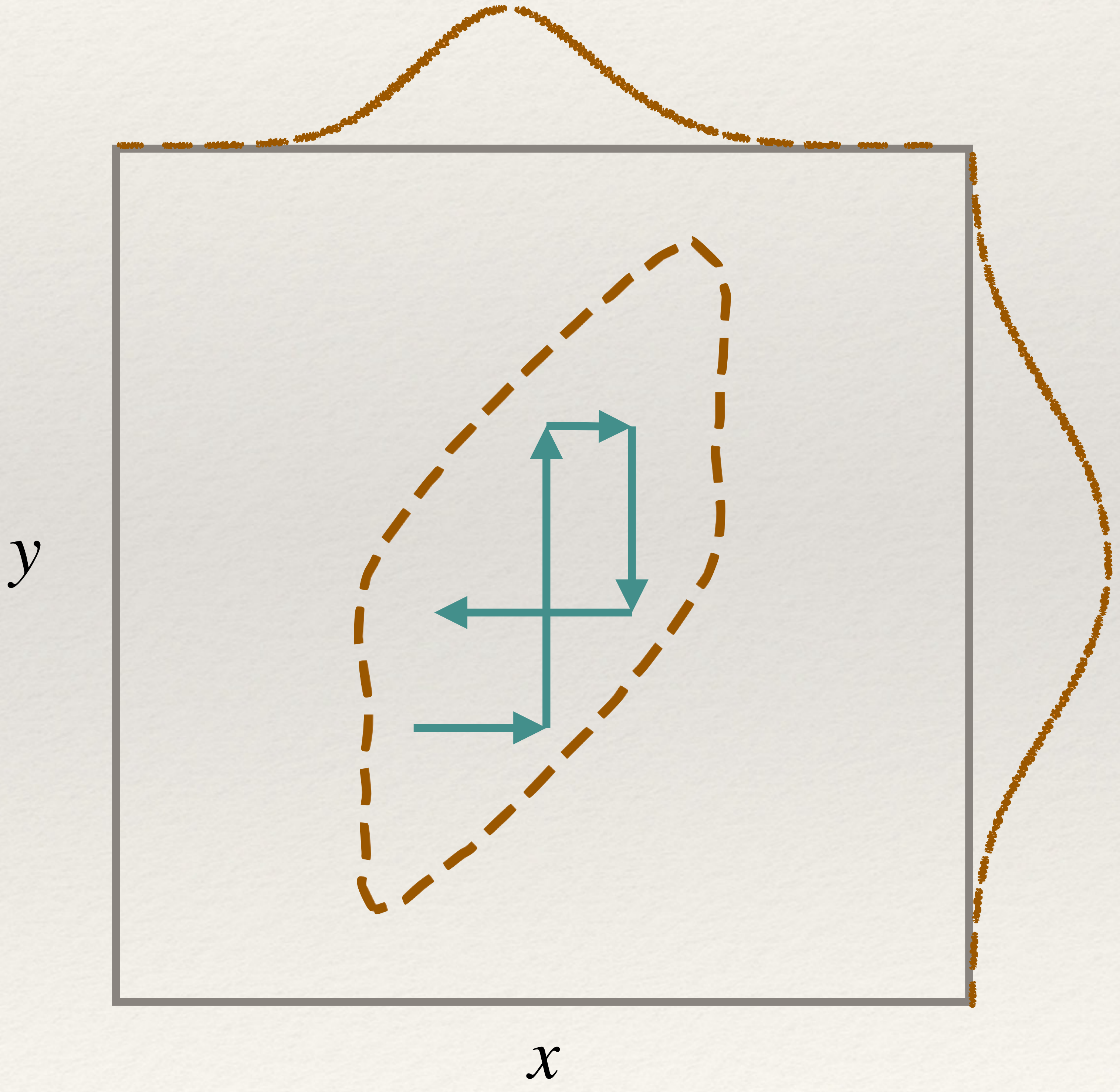
Gibbs sampling

- Sample a joint (x, y) distribution by alternately sampling $x|_y$ and $y|_x$

$$p(x, y) \longrightarrow \begin{cases} p(x|y) \\ p(y|x) \end{cases}$$

(1) $x \sim p(x|y)$

(2) $y \sim p(y|x)$



Gibbs sampling + normalizing flows

Dax, SRG+ (ICLR, 2022)

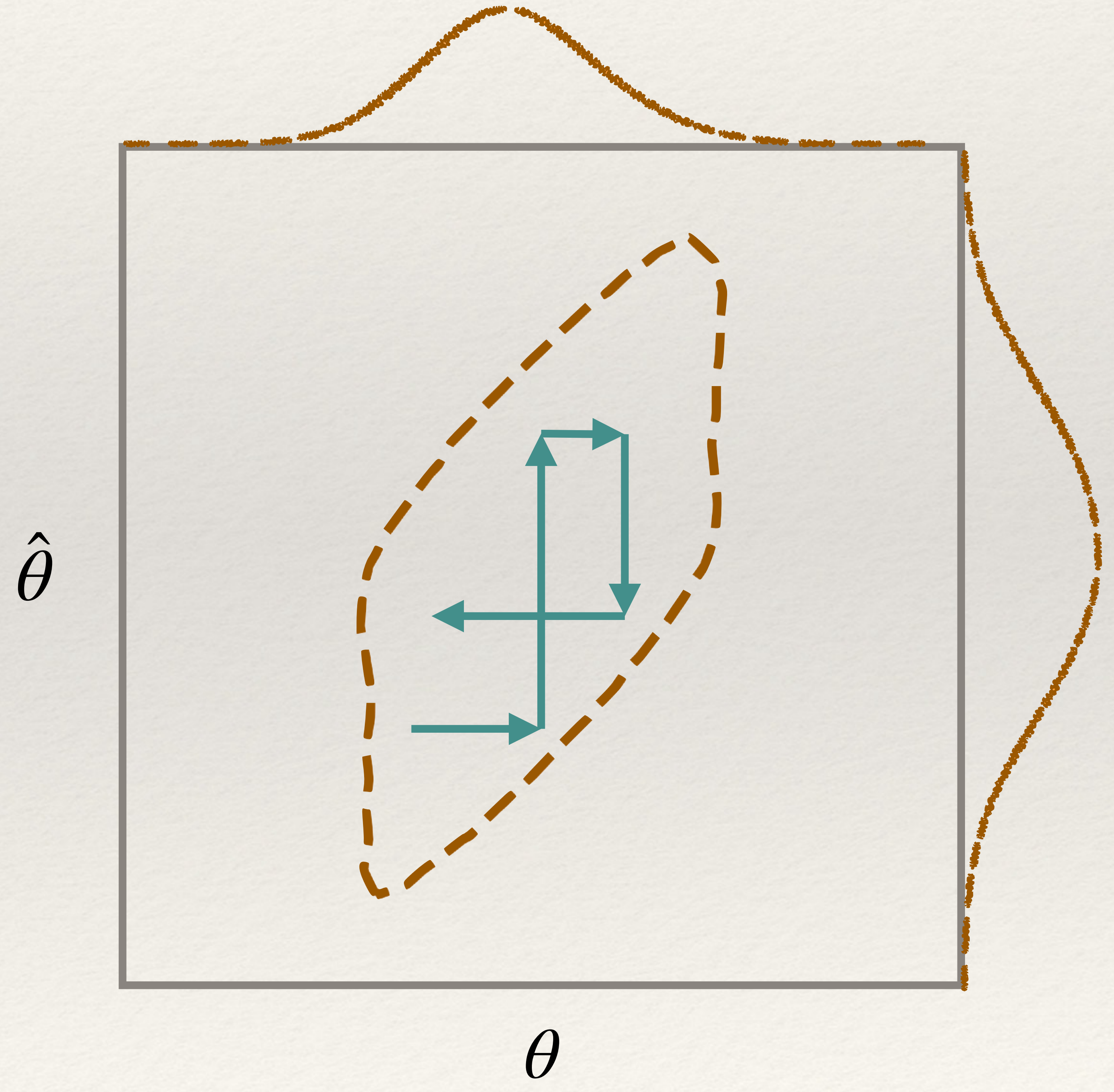
- Take $x \rightarrow \theta$ (e.g., time of coalescence)

  $y \rightarrow \hat{\theta}$ “blurred” parameters

- Joint distribution

Transform based on $\hat{\theta}$

$$p(\theta, \hat{\theta}|d) \longrightarrow \begin{cases} p(\theta|\boxed{d}, \hat{\theta}) & \text{normalizing flow} \\ p(\hat{\theta}|\theta) & \text{fixed kernel} \end{cases}$$



Group Equivariant NPE

Dax, SRG+ (ICLR, 2022)

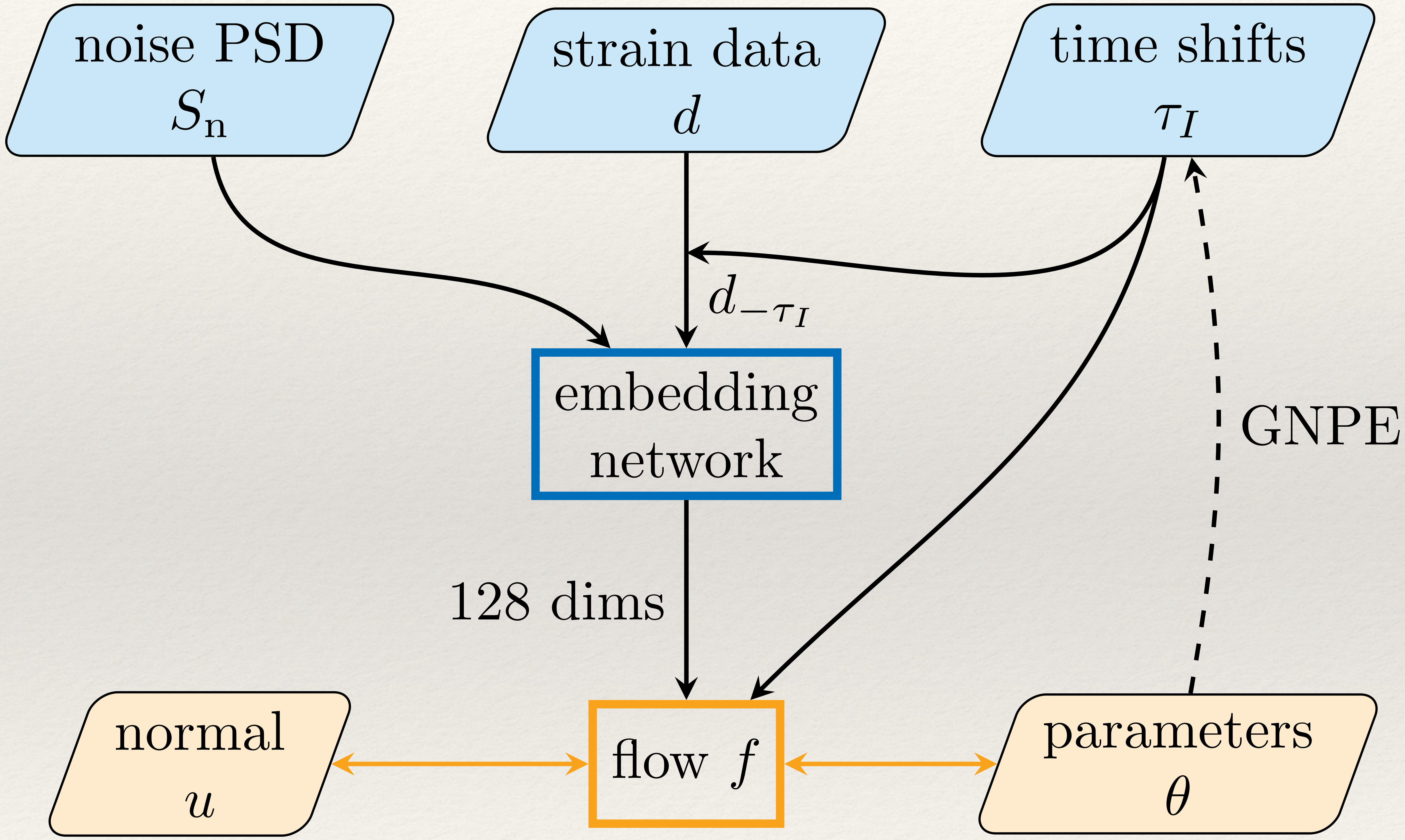
$$p(\theta, \hat{\theta}|d) \longrightarrow \begin{cases} p(\theta|d, \hat{\theta}) & \text{normalizing flow} \\ p(\hat{\theta}|\theta) & \text{fixed kernel} \end{cases}$$

- Gibbs sampling enables us to apply a $\hat{\theta}$-dependent transformation to *d*,
  - $p(\theta\,|\,d, \hat{\theta}) \equiv q(\theta\,|\,T_{\hat{\theta}}d, \hat{\theta})$
- To enforce a symmetry, e.g., time translation,
  - $p(t\,|\,d, \hat{t}) \equiv q(t - \hat{t}\,|\,T_{-\hat{t}}d)$

Generic method to incorporate symmetries:
- **Any** symmetry connecting data and parameters
- **Any** architecture
- **Minimal** changes needed

Group-Equivariant NPE

Dax, SRG+ (ICLR, 2022)



noise PSD S_{n}

strain data d

time shifts τ_I

$d_{-\tau_I}$

embedding network

GNPE

128 dims

normal u

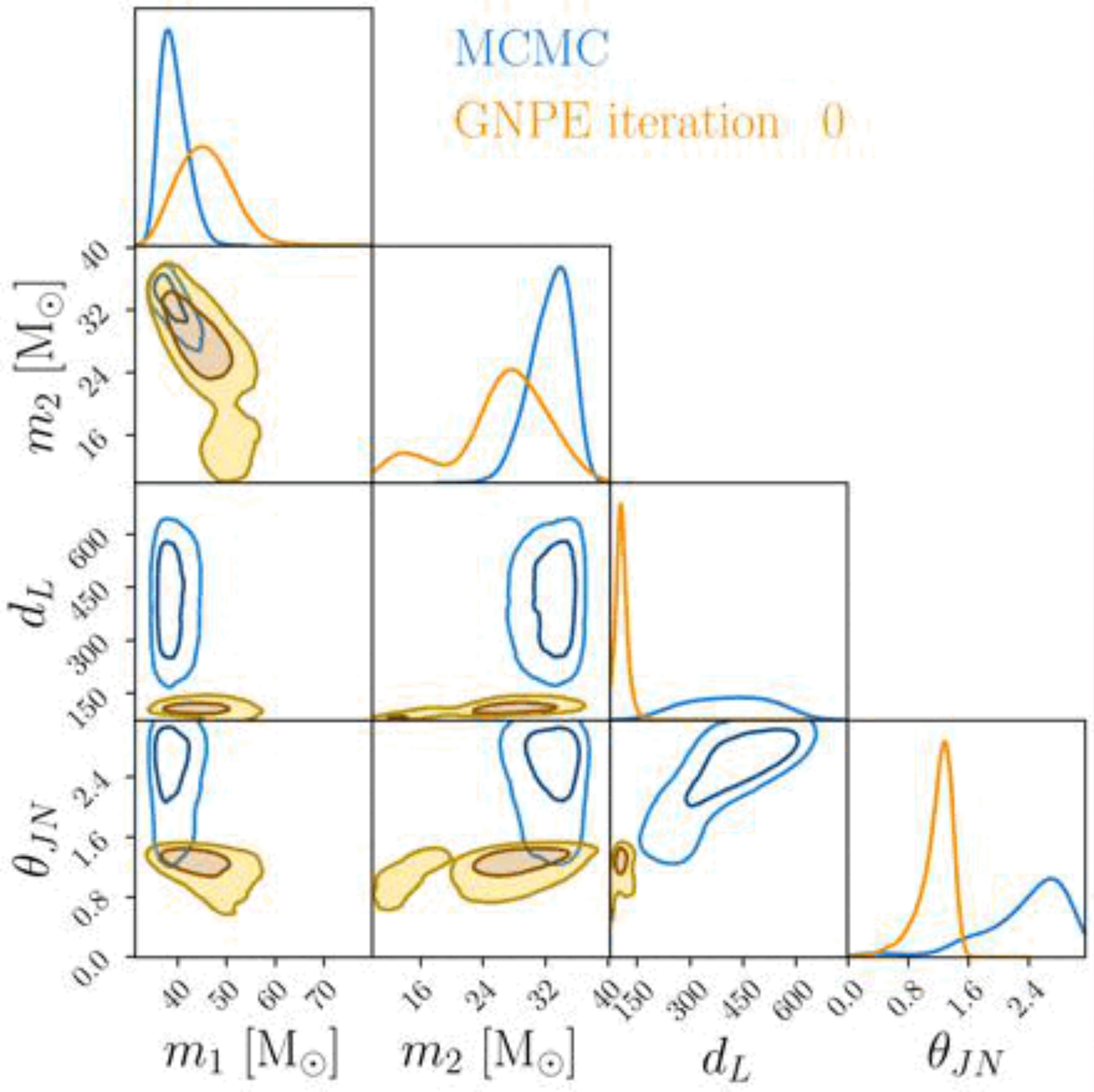
flow f

parameters θ

Real-time convergence with GNPE

Dax, SRG+ (ICLR, 2022)
Dax, SRG+ (PRL 2021)

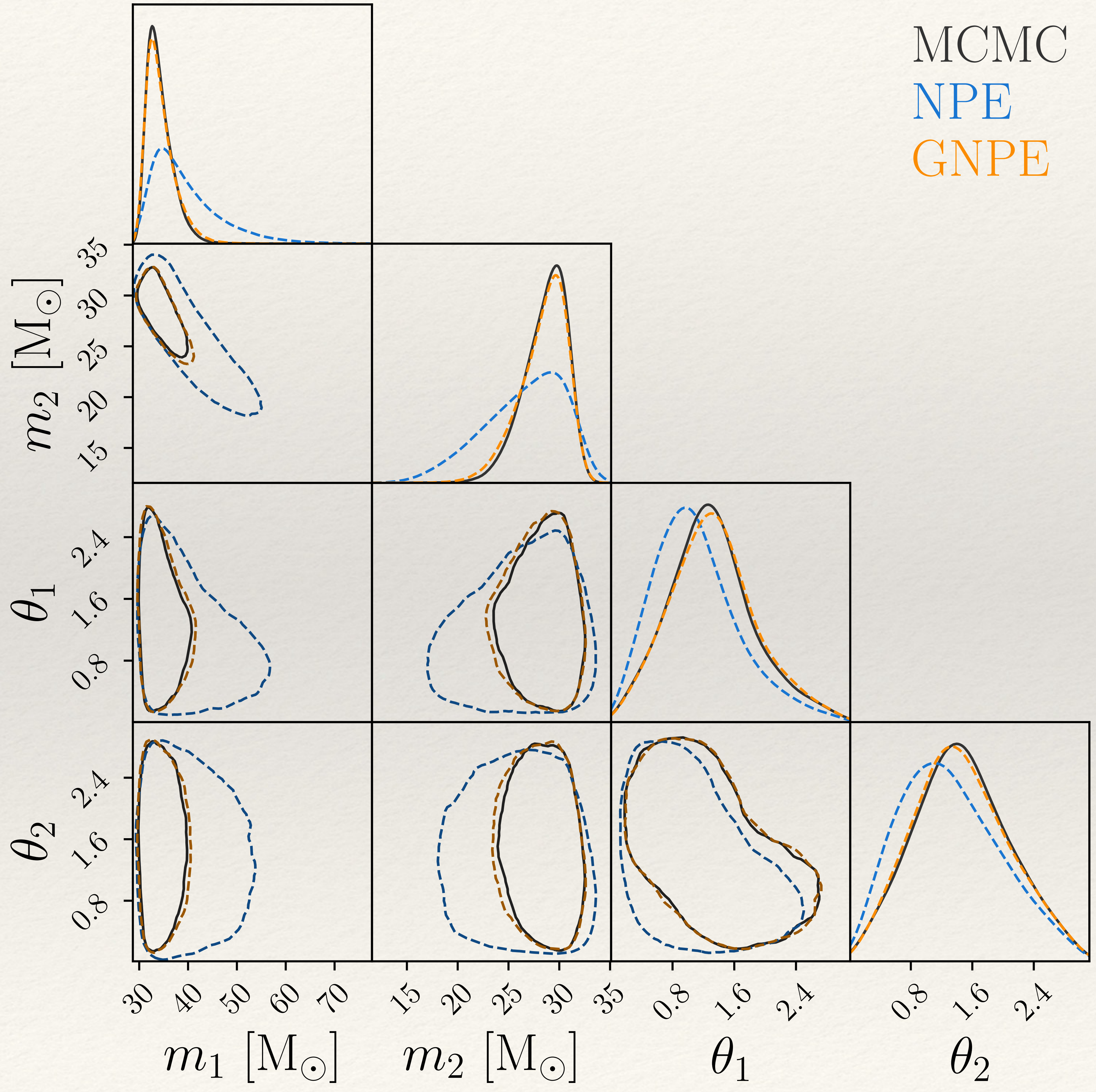
- Application to sky position + time of coalescence:
    - Aligns waveforms in each detector.
    - Trade-off between wide kernel (fast convergence) and narrow (data simplification).
    - 1 ms kernel $\Longrightarrow$ 30 iterations



Example

Dax, SRG+ (ICLR, 2022)

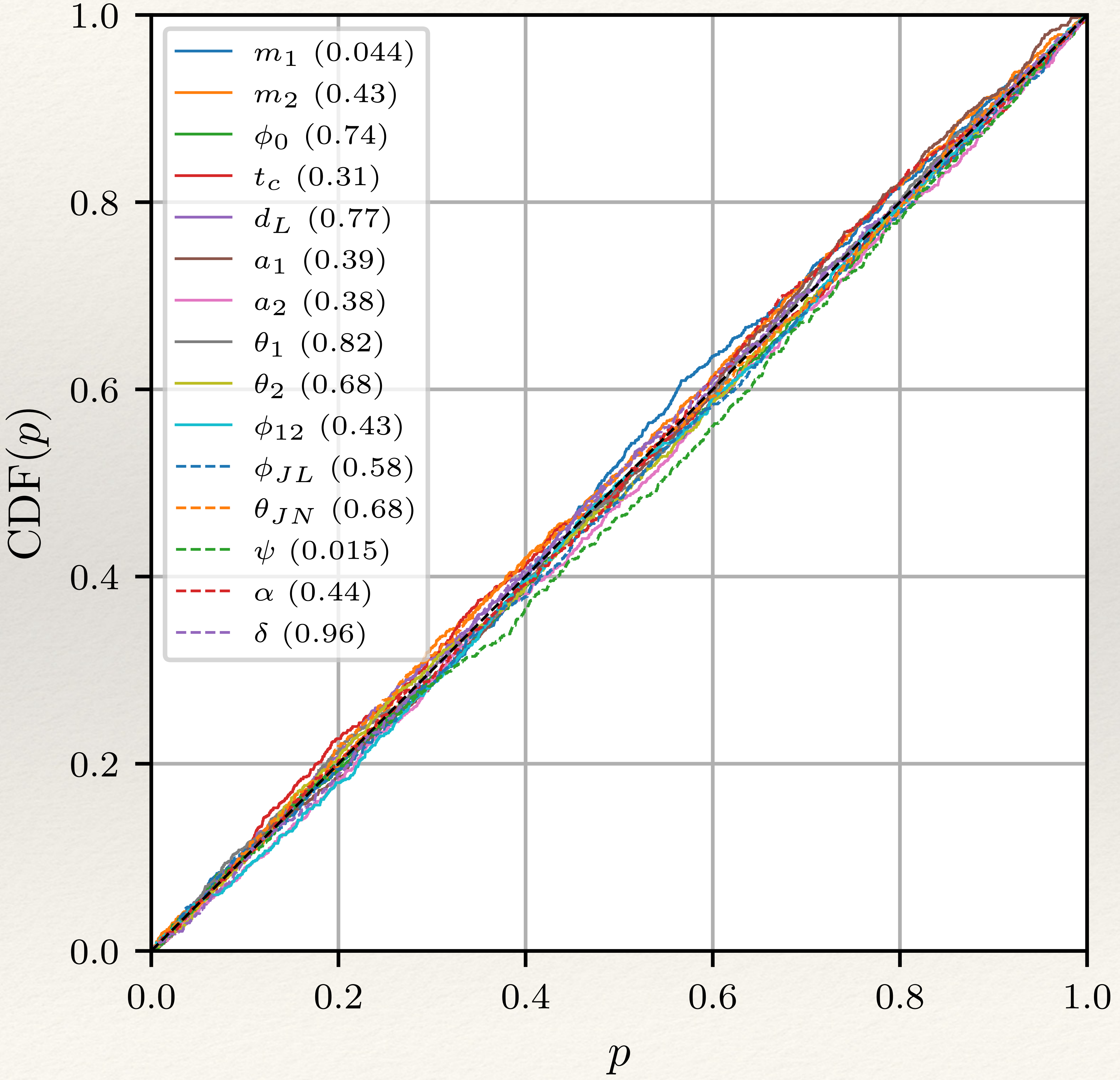
GW170814



Results: P-P plot

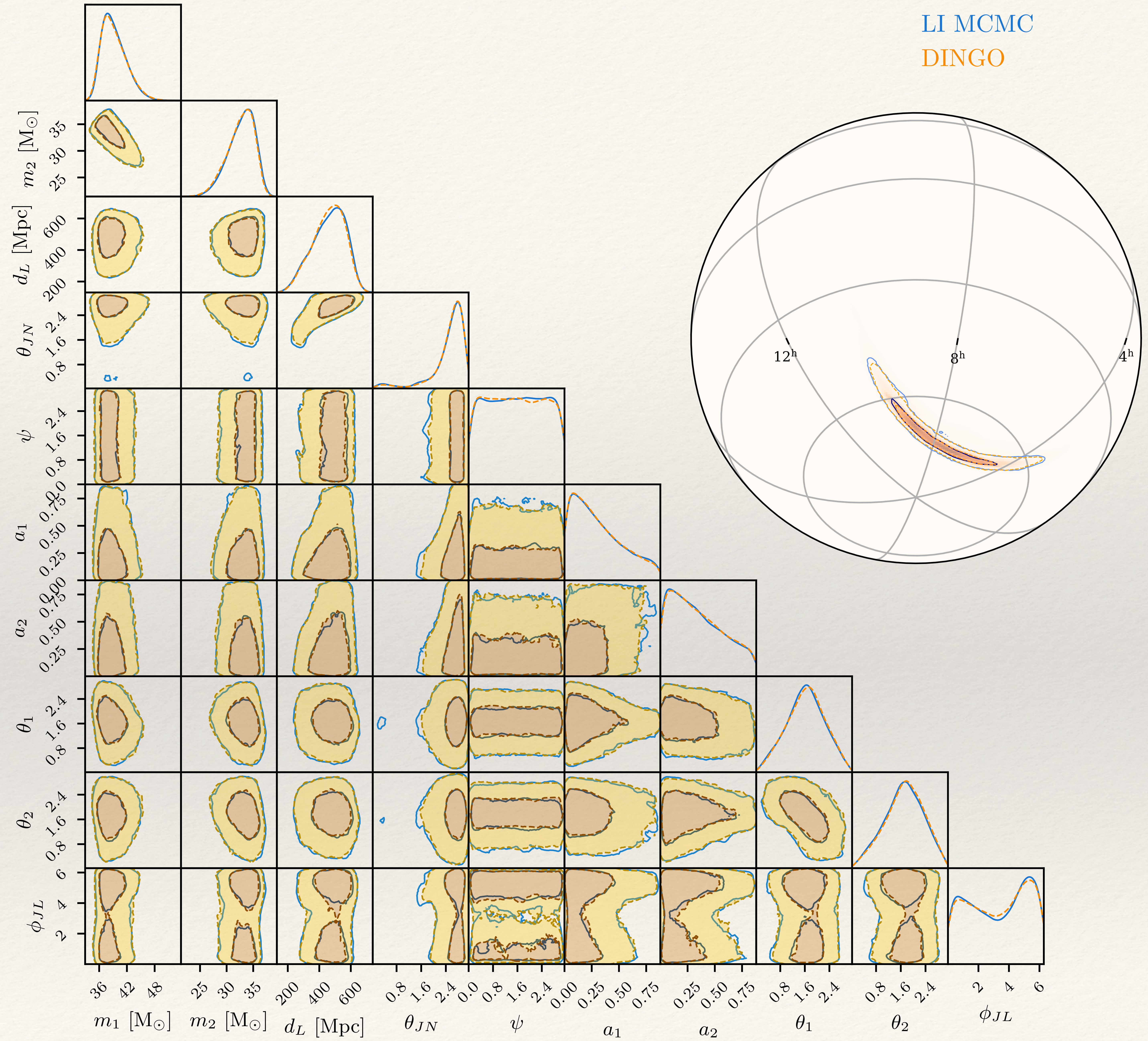
Dax, SRG+ (PRL 2021)

- Perform inference on 1000 simulated data sets
  - "within-distribution" test



GW150914

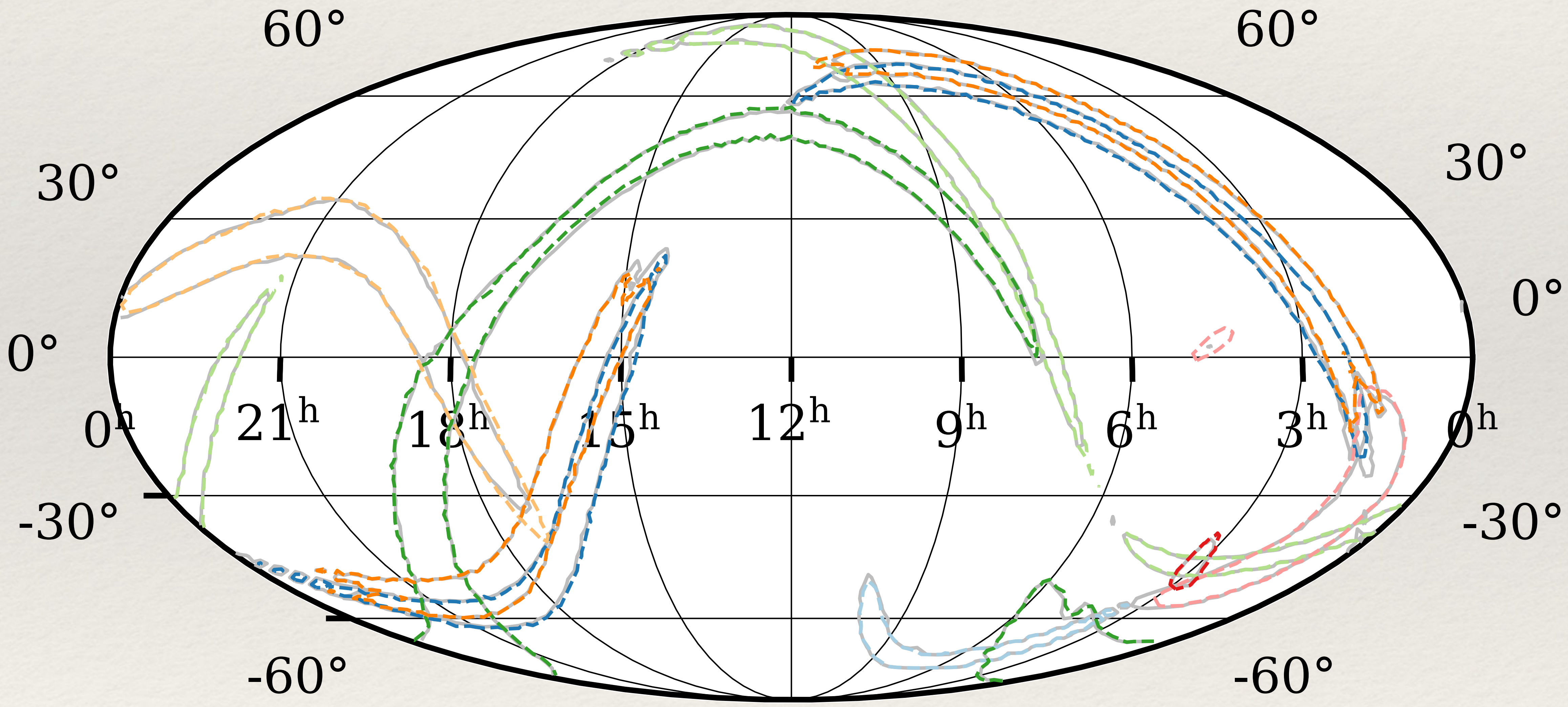
- Real data “out of distribution”
  - Noise not perfectly stationary Gaussian
  - Signal model not perfectly accurate
- Inference times:
  - **NPE: < minute**
    MCMC: ~ day

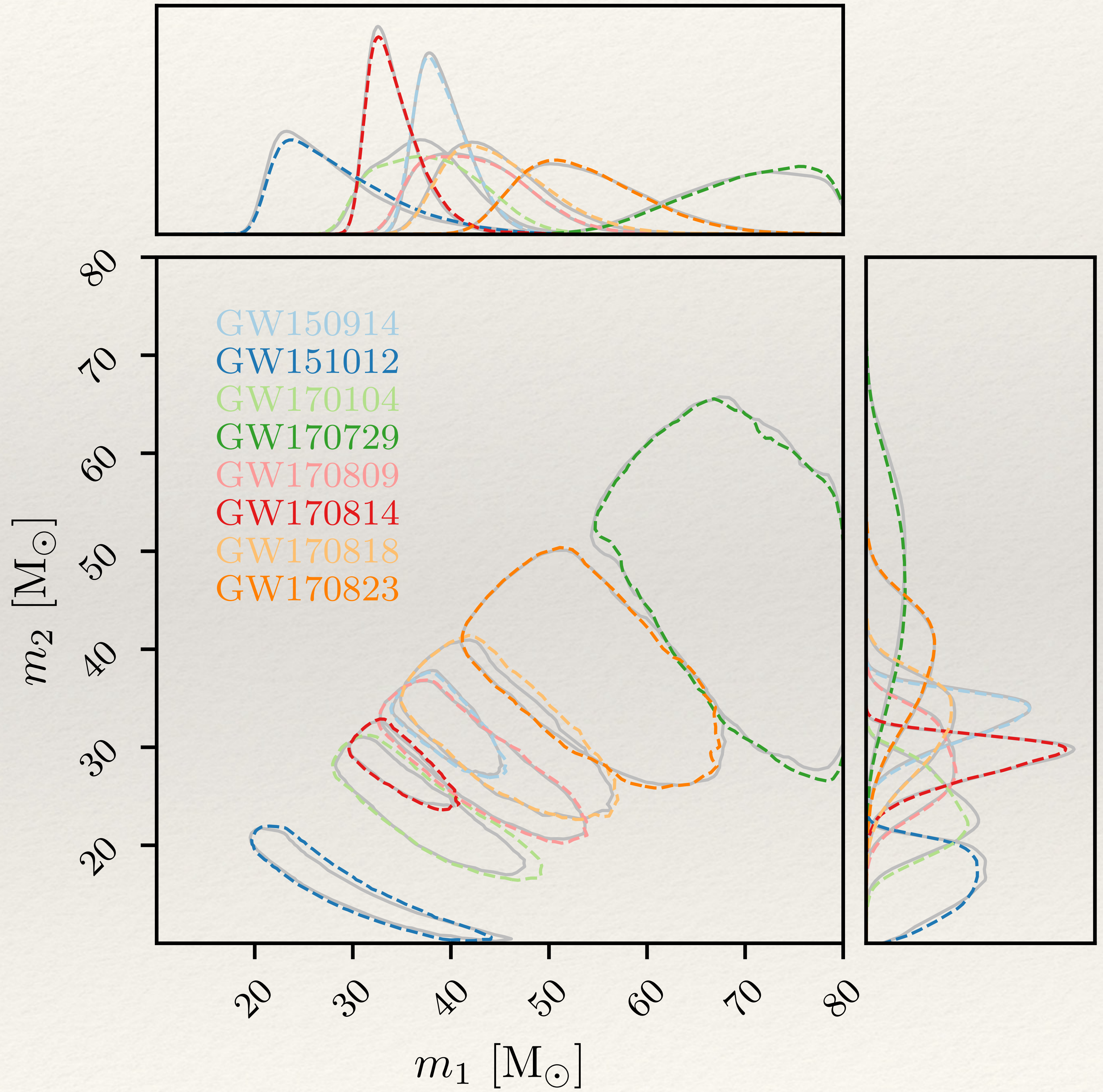


Amortized inference

Dax, SRG+ (PRL 2021)

First Gravitational-Wave Transient Catalog

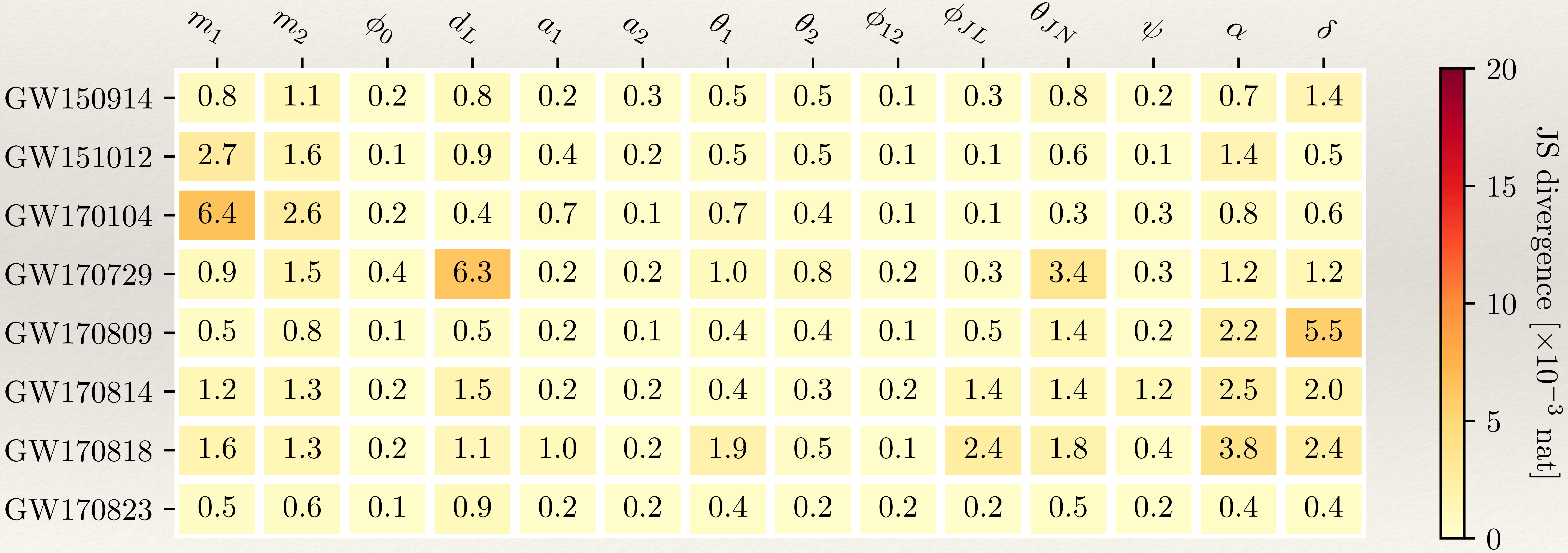




Quantitative comparisons

Dax, SRG+ (PRL 2021)

- 1D marginals: < 2 nat $\Longrightarrow$ “indistinguishable”

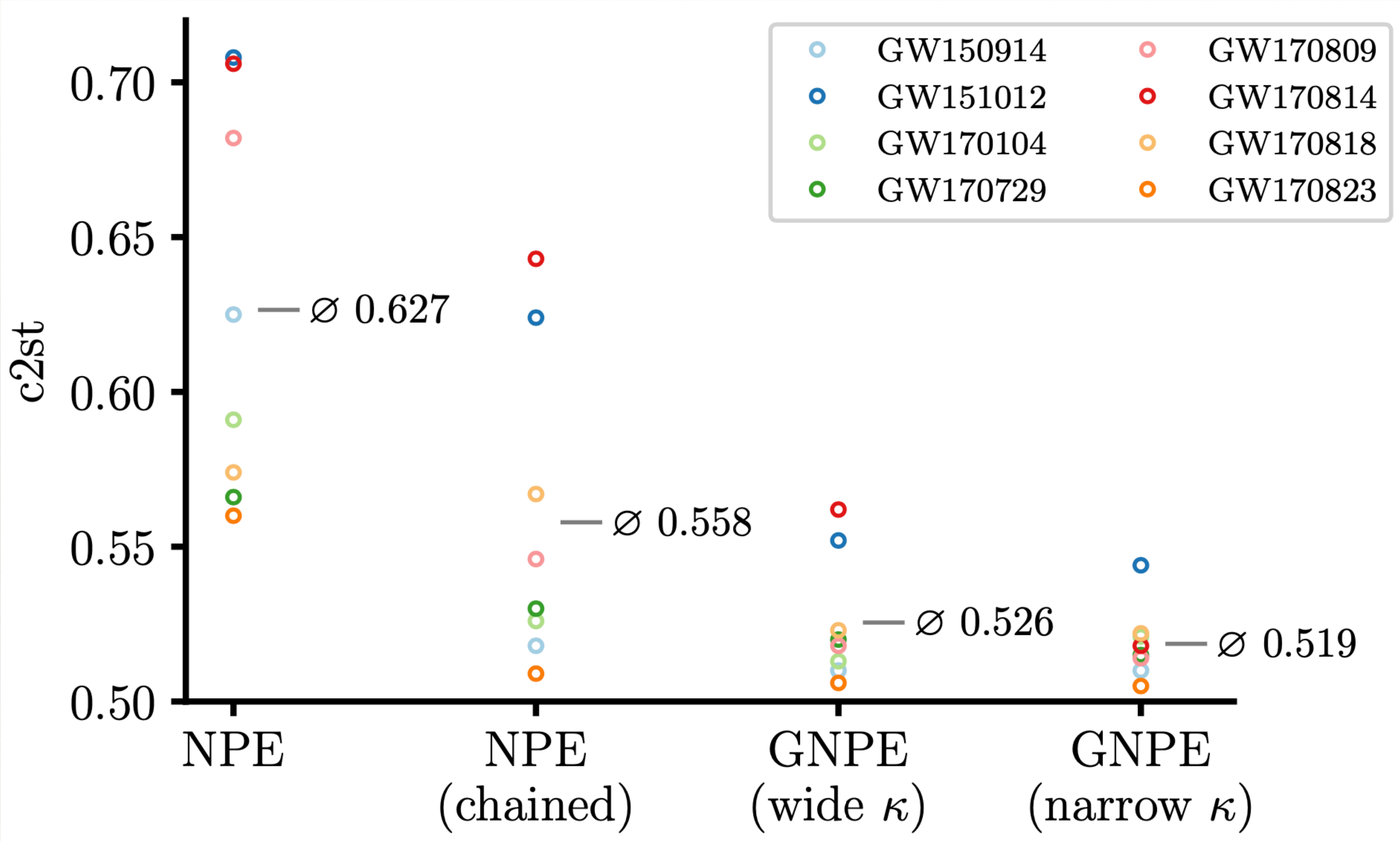


	m_1	m_2	ϕ_0	d_L	a_1	a_2	θ_1	θ_2	ϕ_{12}	ϕ_{JL}	θ_{JN}	ψ	α	δ
GW150914	0.8	1.1	0.2	0.8	0.2	0.3	0.5	0.5	0.1	0.3	0.8	0.2	0.7	1.4
GW151012	2.7	1.6	0.1	0.9	0.4	0.2	0.5	0.5	0.1	0.1	0.6	0.1	1.4	0.5
GW170104	6.4	2.6	0.2	0.4	0.7	0.1	0.7	0.4	0.1	0.1	0.3	0.3	0.8	0.6
GW170729	0.9	1.5	0.4	6.3	0.2	0.2	1.0	0.8	0.2	0.3	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

c2st score

Dax, SRG+ (ICLR, 2022)

- Compare posteriors in 15D
- Train a classifier to distinguish posteriors
  - 0.5 — best
    1.0 — worst



Questions I get asked at this point

- How can I trust your black box? Nobody understands why deep learning works.
- What if the real data don't match the training data? How do I know if the results are any good?
- If I have to run a classical sampler to check each result, what is the point of this?

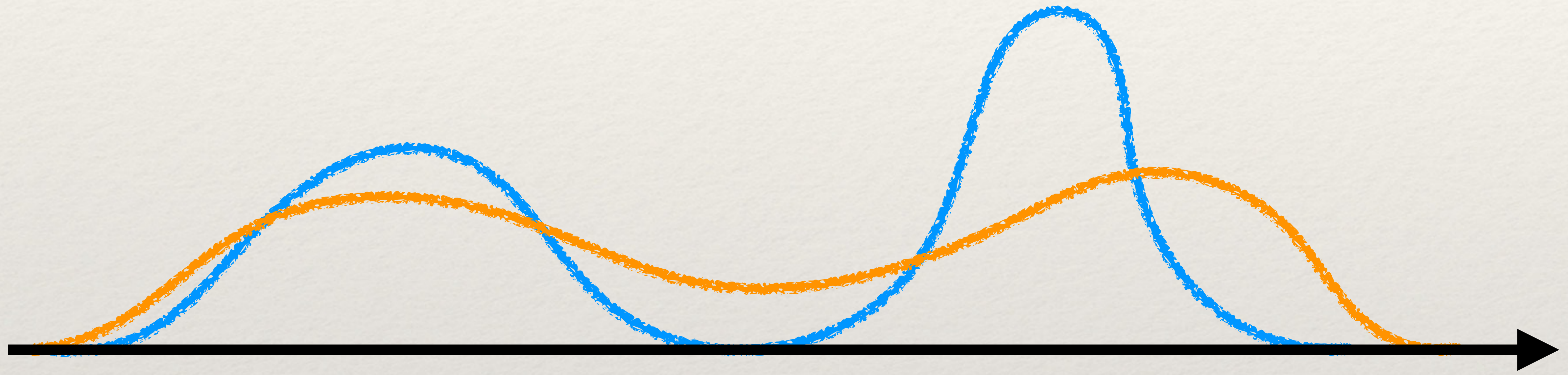
Outline

- Gravitational-waves: Why use Bayesian deep learning?
- Neural posterior estimation
- Using symmetries to simplify data
- Validating / improving results with importance sampling

PRELIMINARY

Importance sampling

- Define importance weights as post-processing step.



target (prior x likelihood)

$$w_i \propto \frac{p(\theta_i)p(d\,|\,\theta_i)}{q(\theta_i\,|\,d)}$$

proposal (NPE)

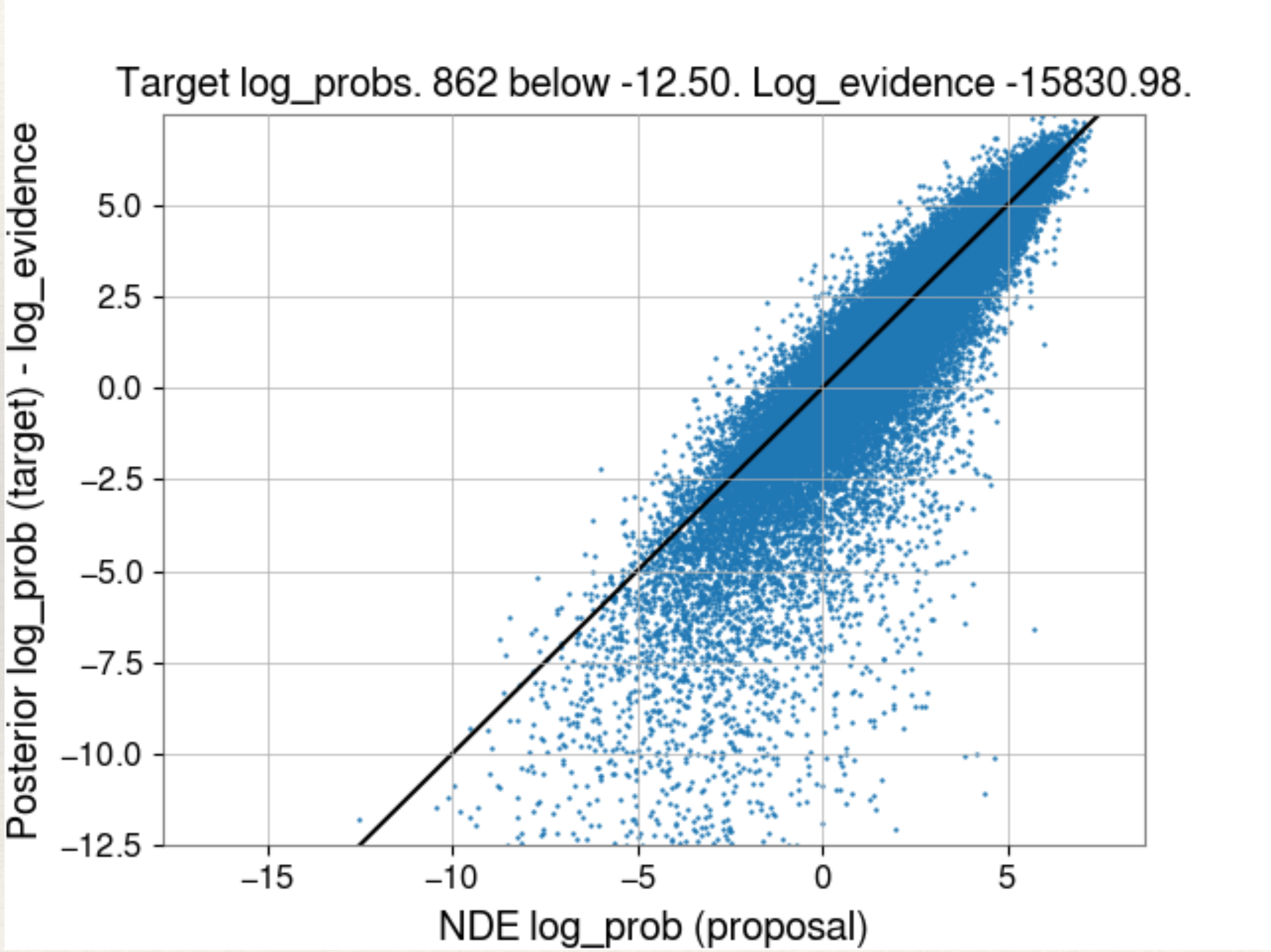
- **Key ingredients:**
  - Access to density, not just samples ✓
  - Excellent proposal in 15D ✓
  - GW likelihood available (with IS, no longer likelihood-free) ✓

PRELIMINARY

Importance sampling

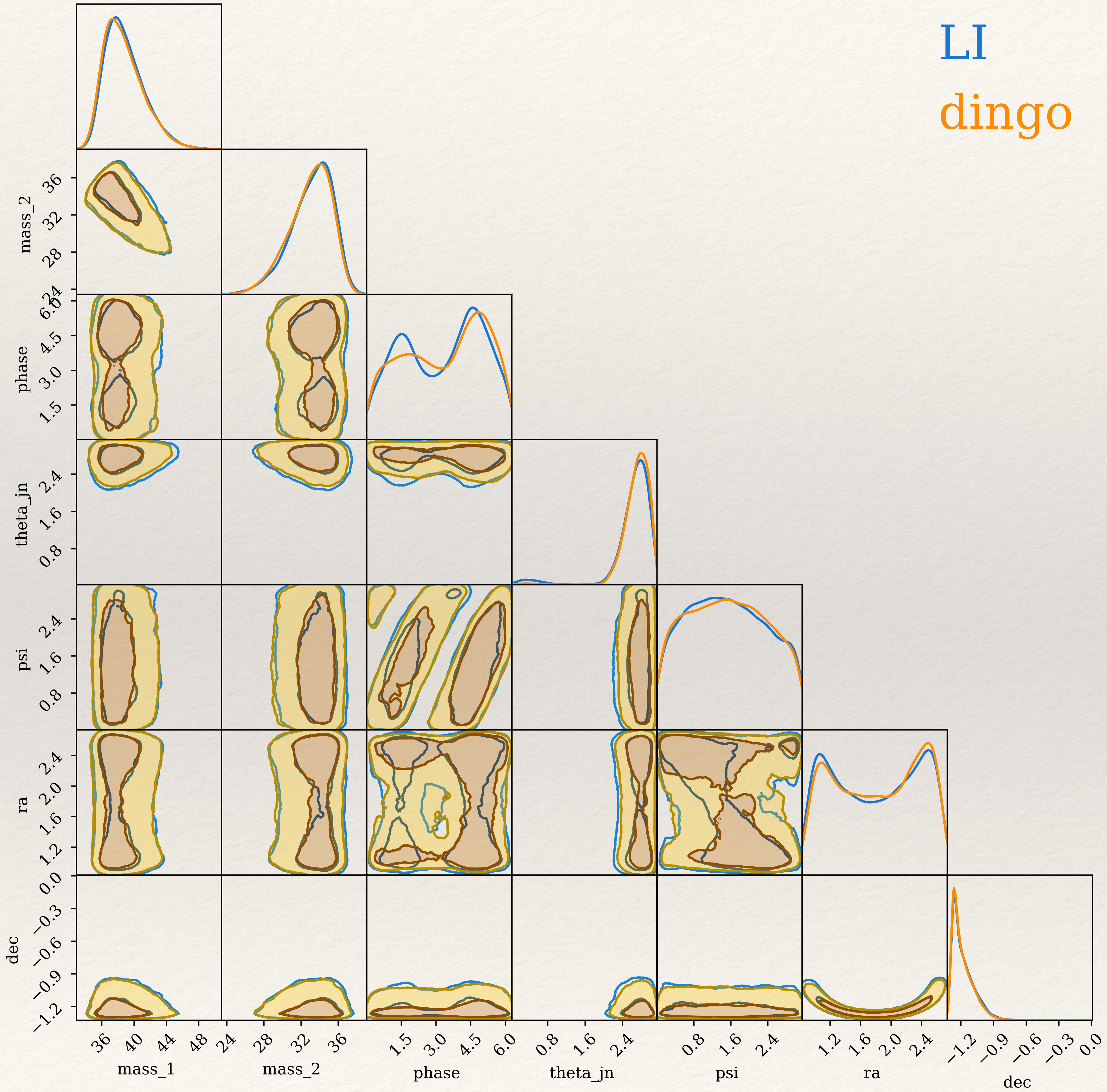
Advantages

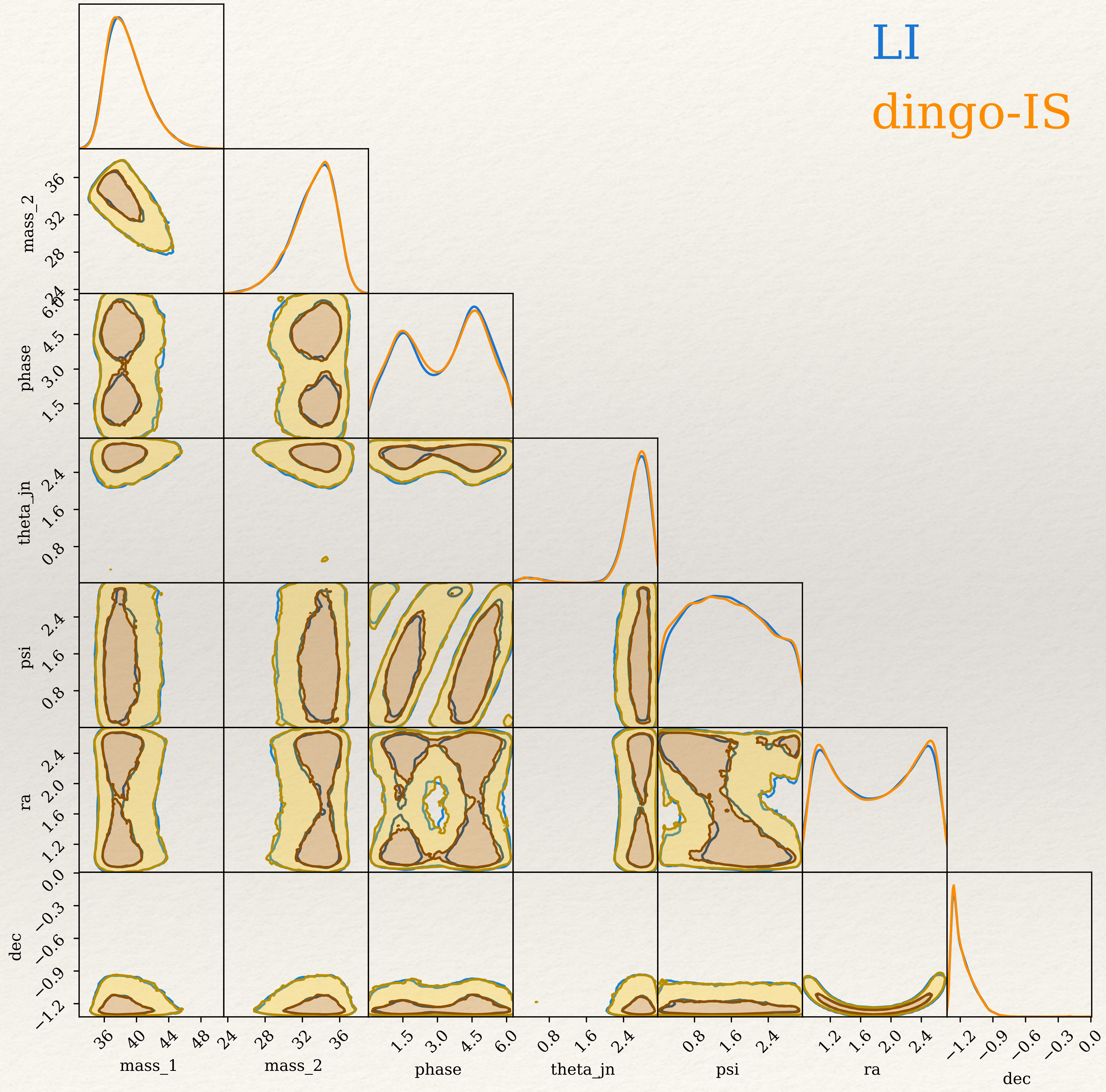
- **Correct any inaccuracies** in NPE samples.
- **Far fewer likelihood evaluations** than standard samplers (and parallelizable).
- **Effective sample size** $(\sum_i w_i)^2 / \sum_i w_i^2$ as **validation of results,** independent of other samplers.
- Provides estimate of the **evidence.**



PRELIMINARY

Example





ESS = 28k (14%)

Conclusions

- Neural posterior estimation produces **accurate inference results for binary black holes in seconds.**
  - Enables rapid alerts and a means to address large numbers of events.
- Taking advantage of known **symmetries / equivariances** can make the problem simpler.
- When a likelihood is available, **importance sampling** can be used to validate and improve upon SBI results.

Outlook

- SBI can go beyond likelihood-based methods by treating more realistic noise.
- Many other applications, including populations and cosmology; LISA inference.

Thank You!