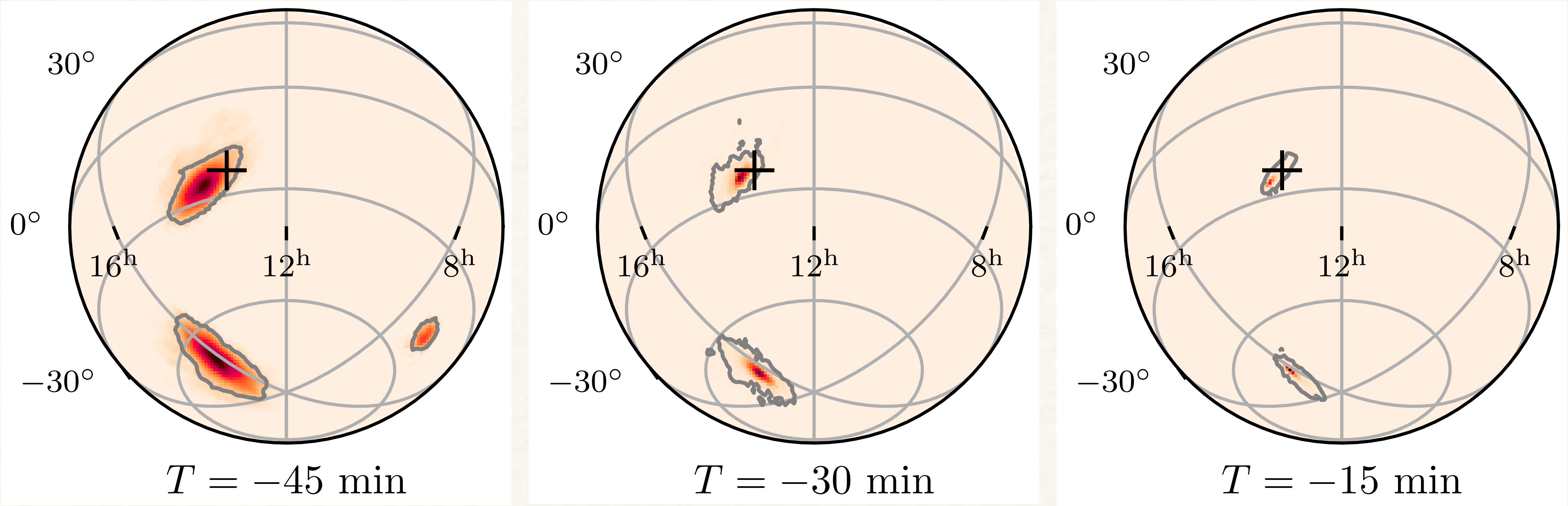
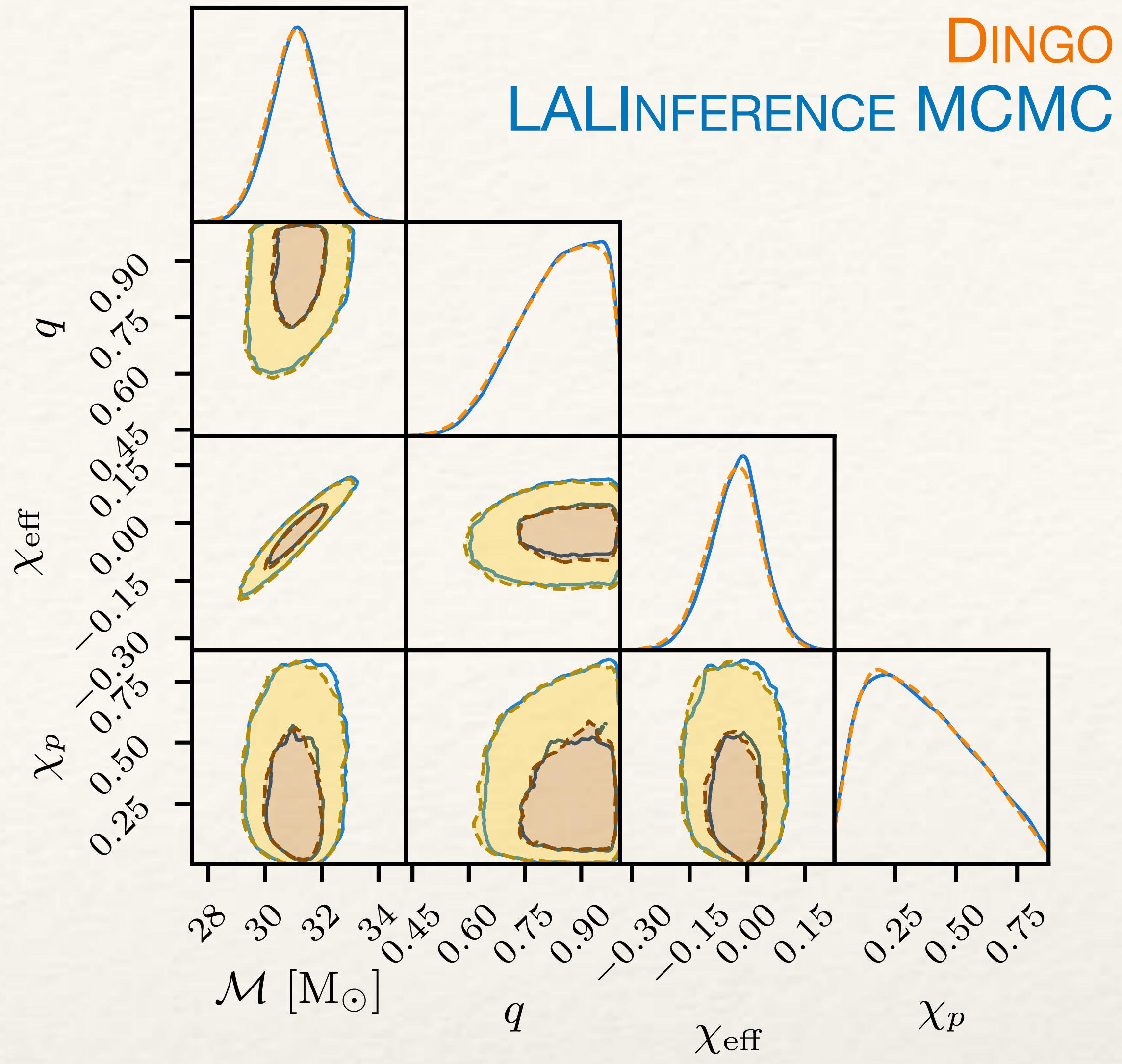




Scientific Machine Learning for Gravitational Wave Astronomy, ICERM
4 June 2025

Neural Posterior Estimation for Gravitational Waves

Stephen Green

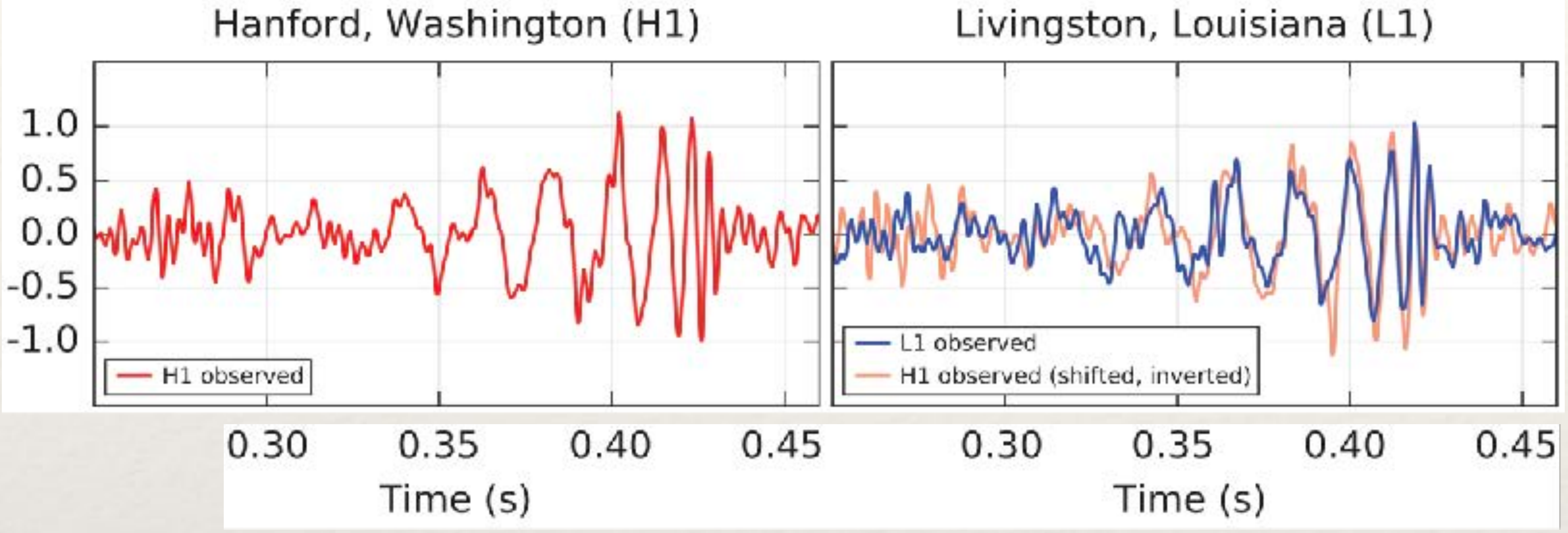
UK Research and Innovation
Future Leaders Fellowships

University of Nottingham
UK | CHINA | MALAYSIA

Outline

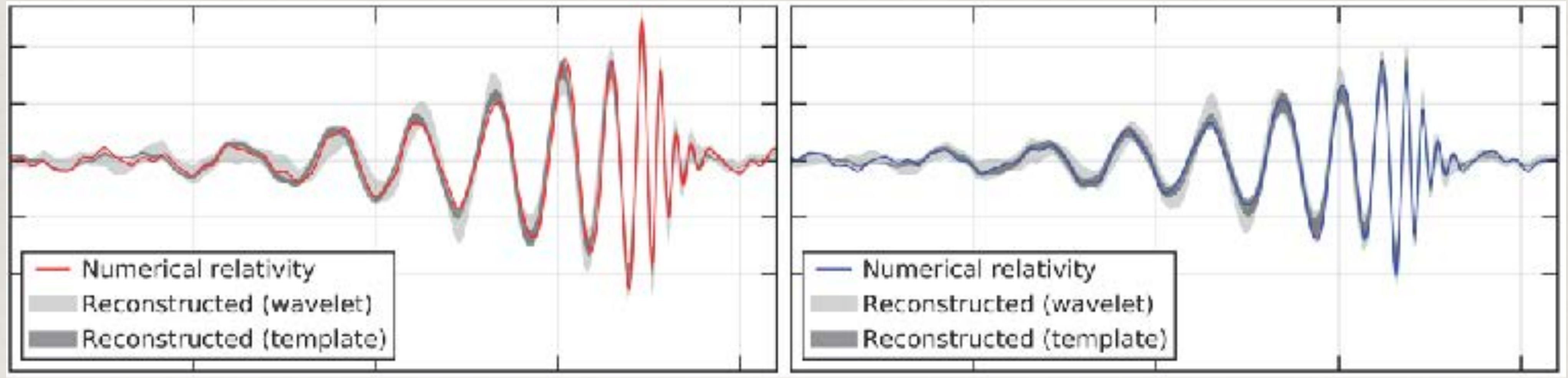
- Simulation-based inference for gravitational waves
- Three case studies
  - Eccentricity
  - Binary neutron stars
  - Flow matching
- DINGO tutorial

Parameter estimation for gravitational waves



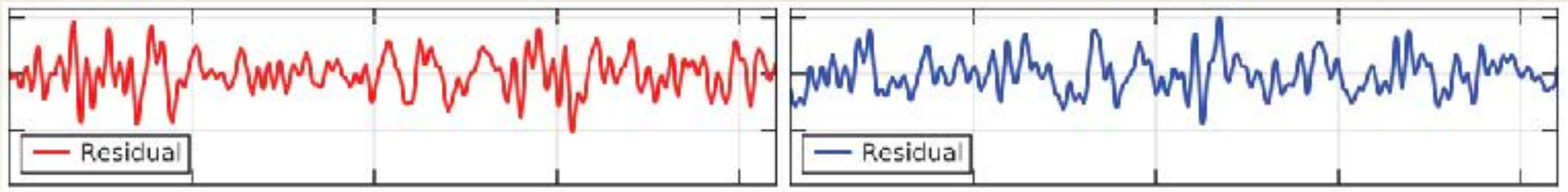
observed data d

$=$

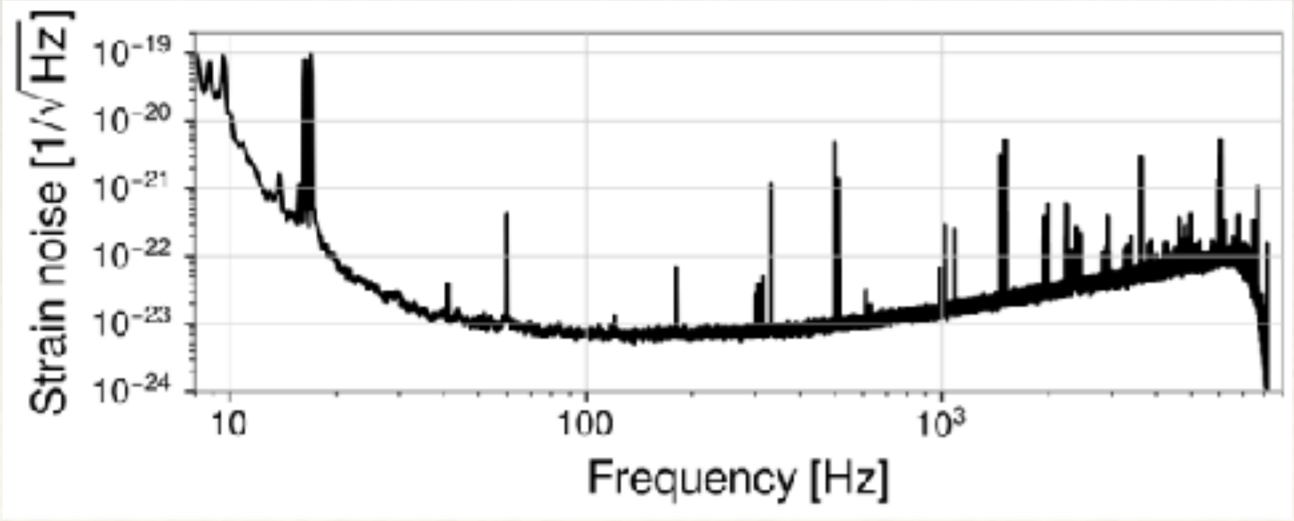


signal $h(\theta)$

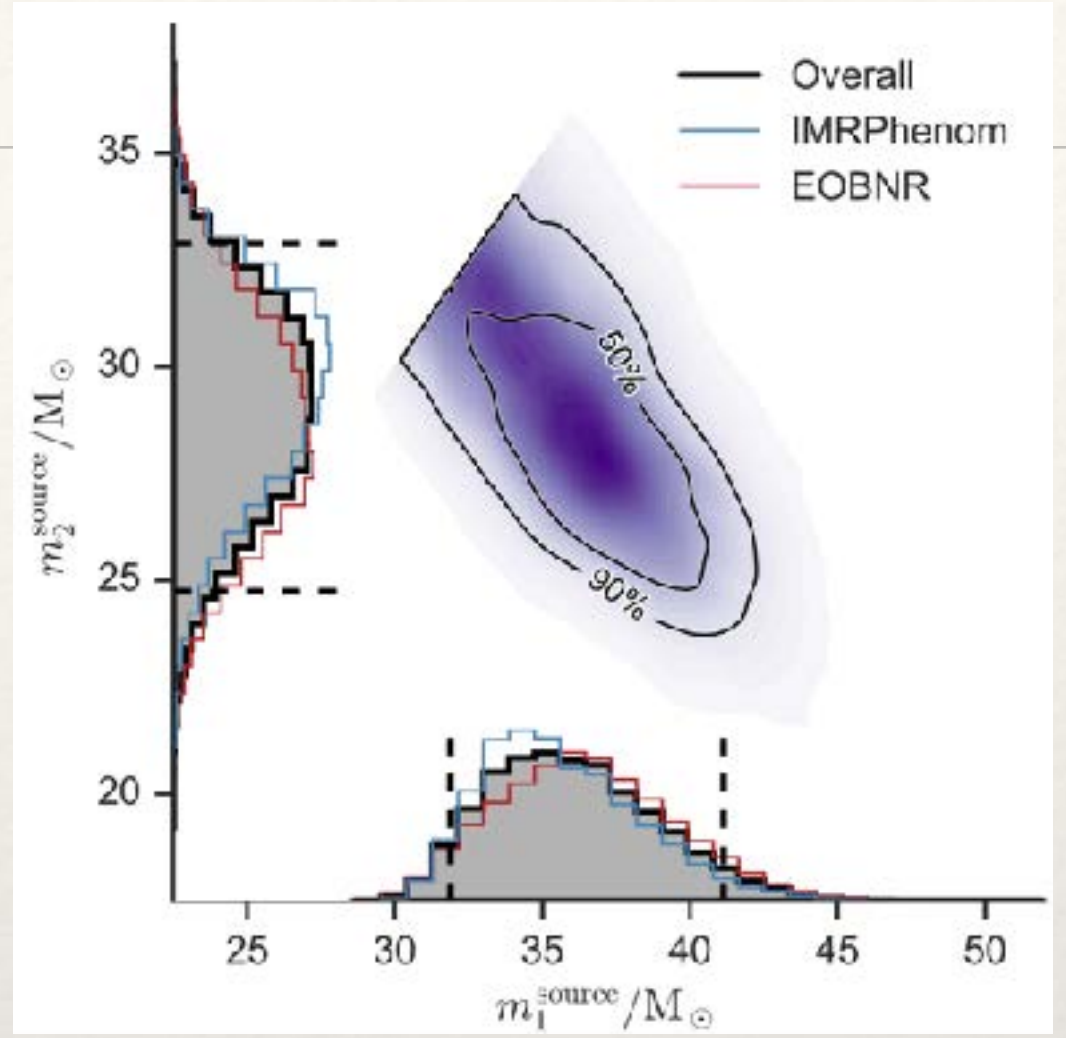
$+$



noise



Bayesian inference



- Goal is to draw samples from the **posterior distribution**

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

- **Gravitational waves**
  - Likelihood assumes **stationary Gaussian** detector noise $p(d \,|\, \theta) = \mathcal{N}(h_I(\theta), S_{\mathrm{n},I})$
  - Uninformative prior
- Typically use **stochastic sampler,** repeatedly evaluating right hand side
  - **Can be expensive,** depending on cost of $h(\theta)$, and must be repeated for each event

Motivation

Speed

Handle the large number of events expected in the future

Enable fast alerts for electromagnetic observers

Accuracy

Move beyond approximations such as stationary Gaussian noise

Flexibility

Analyze data in most natural representation, e.g., time domain

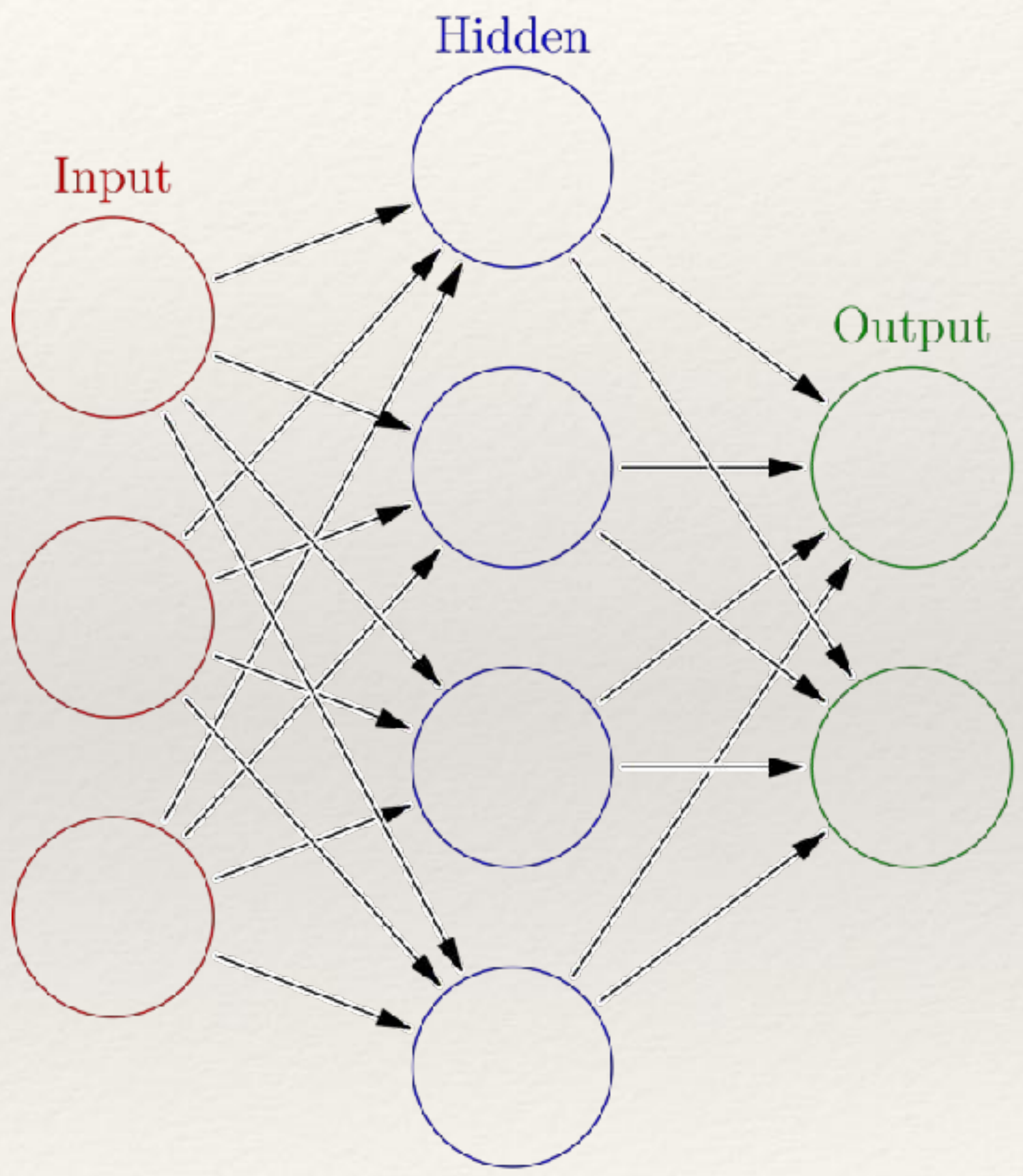
Incorporate other types of data, e.g., galaxy catalogs for populations

Marginalize unwanted parameters

Simulation-based inference

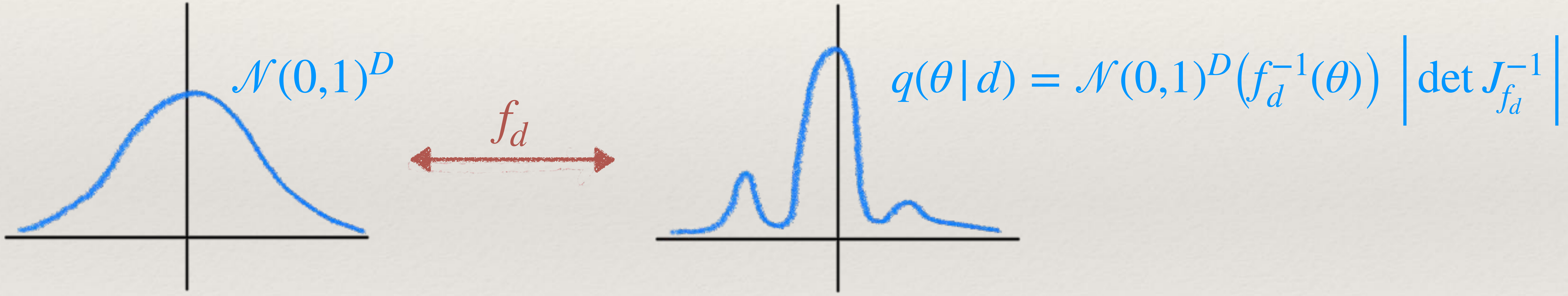
Two key facts

- Deep neural networks have tremendous capacity to **model complicated probability distributions**.
- Using **simulated data alone,** can train networks to learn Bayesian inference distributions (e.g., the posterior). No likelihood evaluations are required.



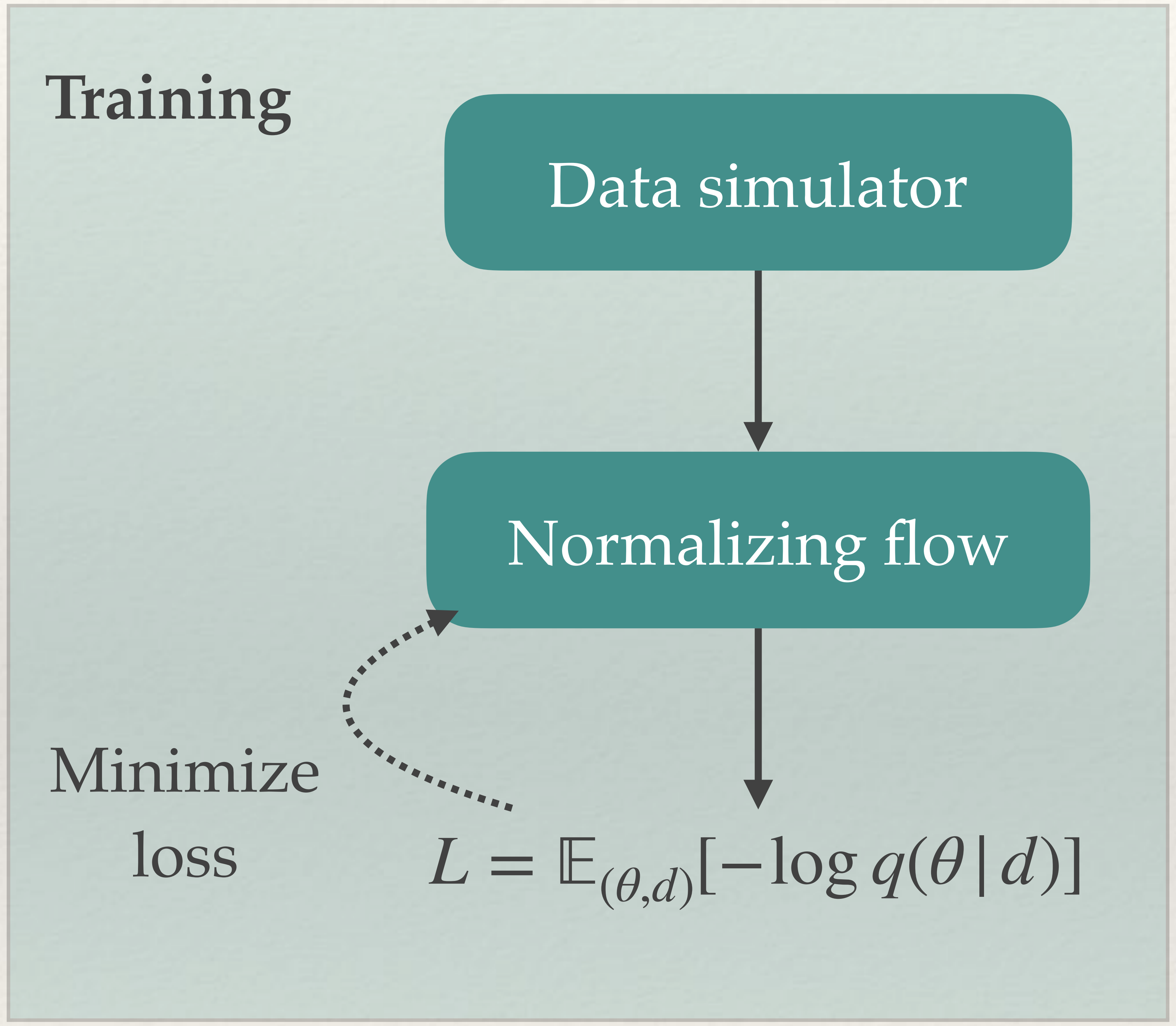
Distributions with neural networks

- For SBI, it is convenient to use a **normalizing flow.** Represents a complex distribution q using a **mapping** $f_d : u \mapsto \theta$ from simpler distribution:

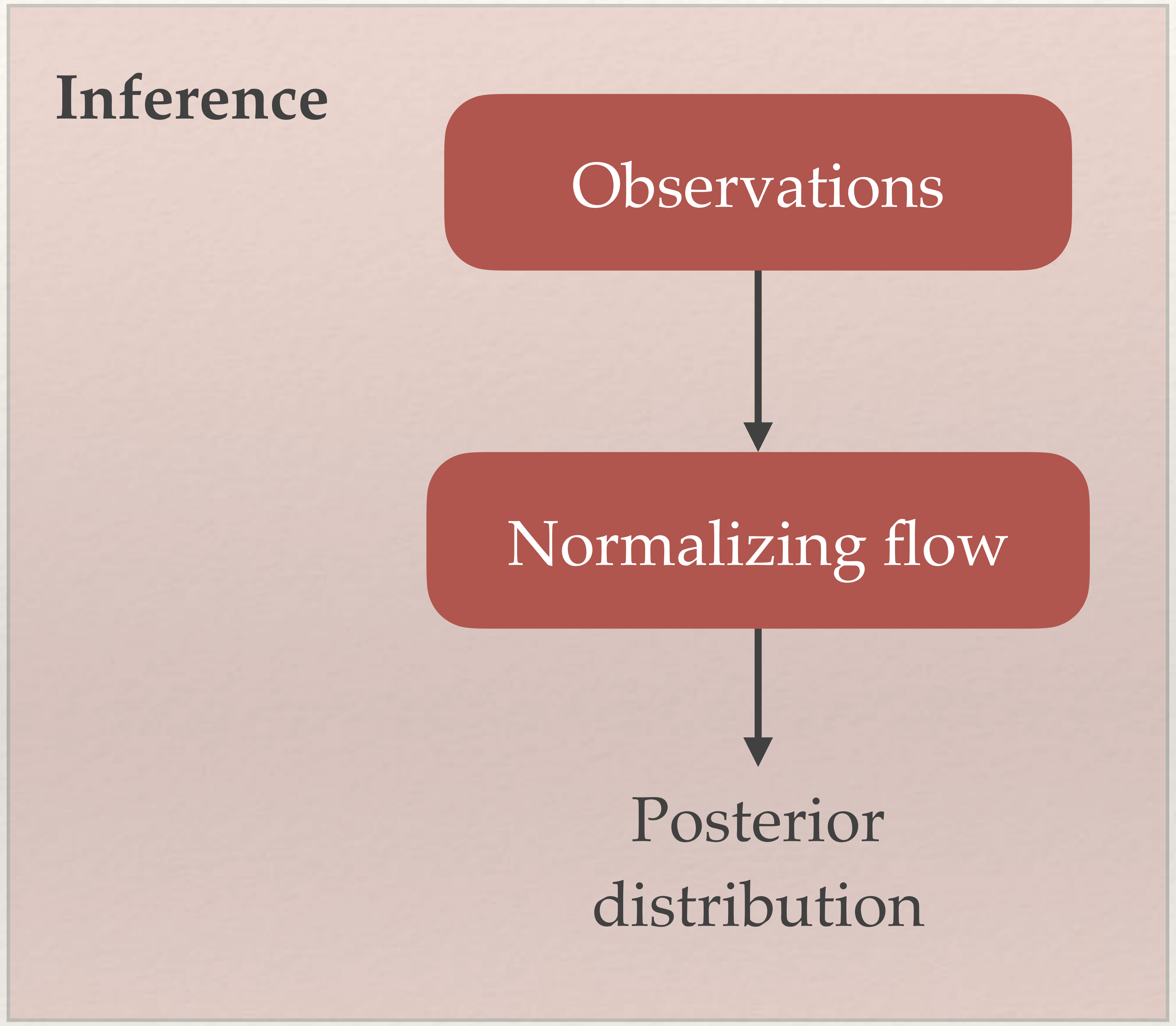


- f_d **is defined using neural networks.** It must be invertible with simple Jacobian determinant.

Neural posterior estimation



More general than likelihood-based inference methods.



Inference is fast since it uses only forward neural network passes.

Loss function

- To train the network, specify a target loss function.
- **Want** $q_\phi(\theta|d) \to p(\theta|d)$
  - Take **Kullbeck-Liebler (KL) divergence** D_{KL} between these distributions

$$D_{\mathrm{KL}}(p|q) = \int d\theta\, p(\theta|d) \log \frac{p(\theta|d)}{q_\phi(\theta|d)} \geq 0$$

$= 0$ for identical distributions

  - This still depends on d so **marginalize** over it

$$\mathbb{E}_{p(d)} D_{\mathrm{KL}}(p|q) = \int dd\, p(d) \int d\theta\, p(\theta|d) \log \frac{p(\theta|d)}{q_\phi(\theta|d)}$$

Loss function

$$\mathbb{E}_{p(d)} D_{\mathrm{KL}}(p\,|\,q) = \int dd\, p(d) \int d\theta\, p(\theta\,|\,d)\, \log \frac{p(\theta\,|\,d)}{q_\phi(\theta\,|\,d)}$$

- To evaluate, re-order the integrals using **Bayes' theorem**

$$\mathbb{E}_{p(d)} D_{\mathrm{KL}}(p\,|\,q) = \int d\theta\, p(\theta) \int dd\, p(d\,|\,\theta)\, \log \frac{p(\theta\,|\,d)}{q_\phi(\theta\,|\,d)}$$

- Finally

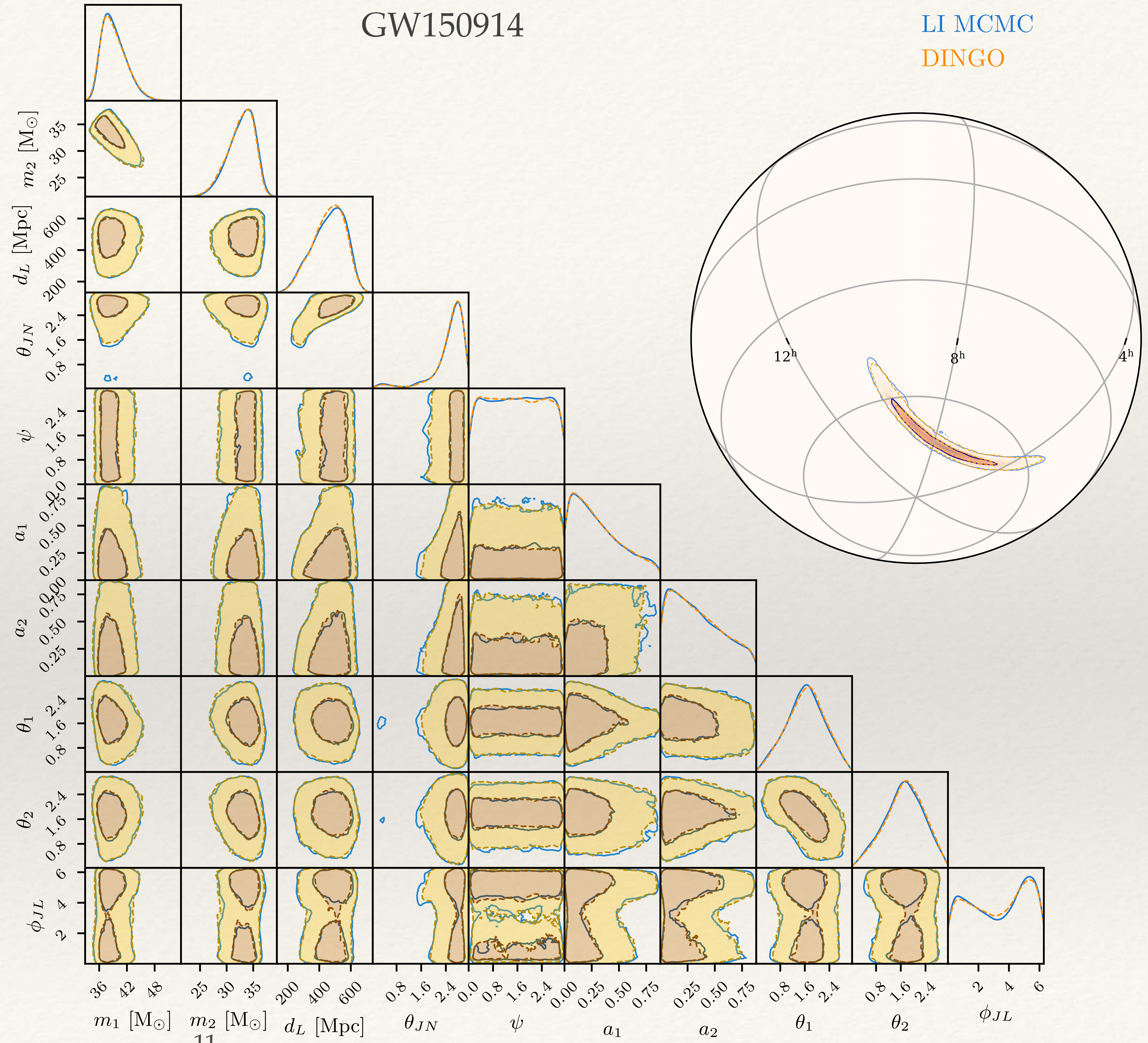
$$\boxed{L[\phi] = \frac{1}{N} \sum_{\substack{\theta^{(i)} \sim p(\theta) \\ d^{(i)} \sim p(d\,|\,\theta^{(i)})}} -\log q_\phi(\theta^{(i)}\,|\,d^{(i)})}$$

1. Sample $\theta^{(i)}$ from the prior
2. Simulate $d^{(i)}$ = signal + noise

- Well-trained networks give extremely good agreement with standard samplers.
- Inference in seconds to minutes.

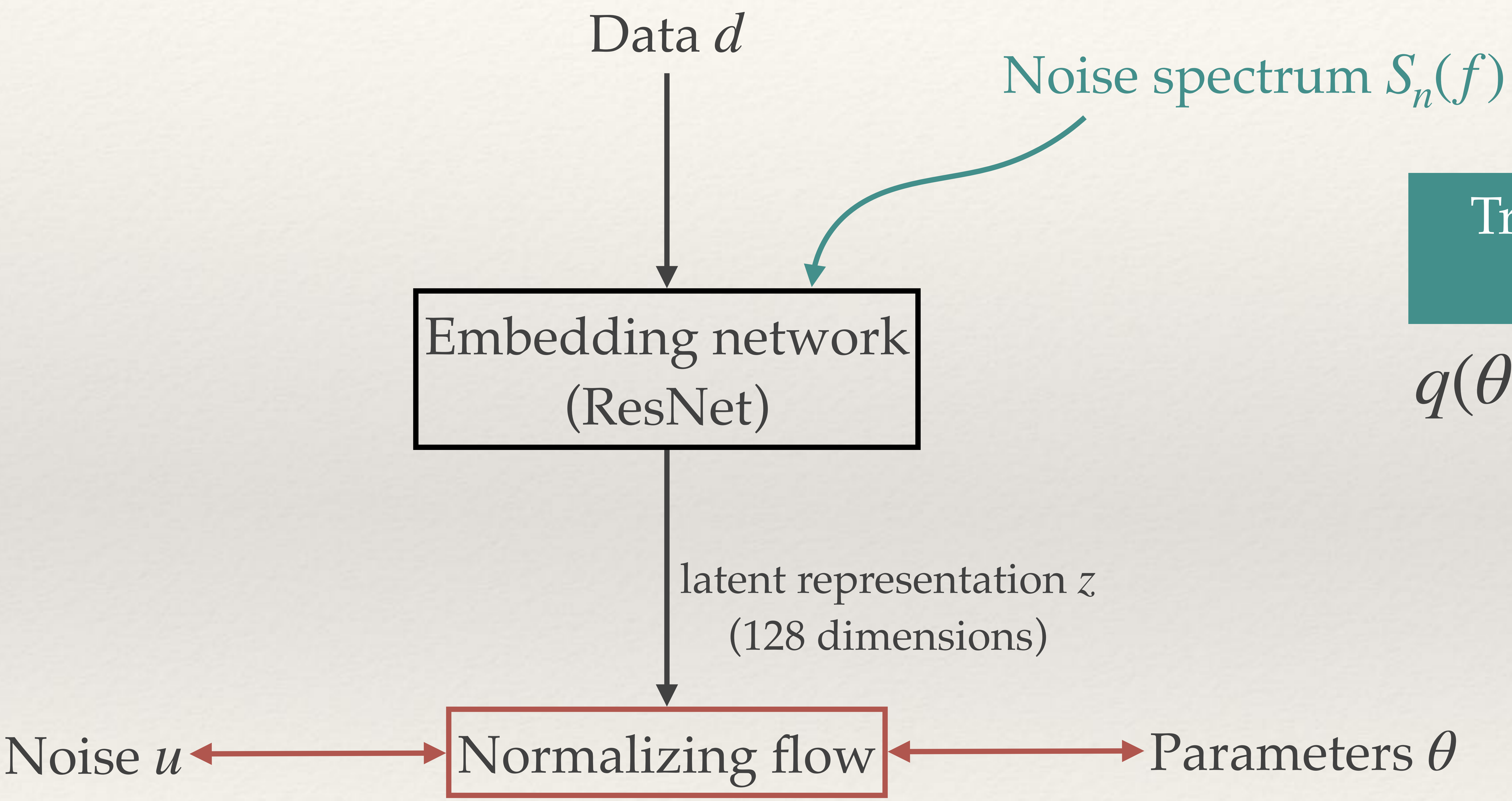
DINGO

https://github.com/dingo-gw/dingo



Dax, SRG+ (PRL 2021)

Architecture



Trivial to condition on extra information, e.g., PSDs

$$q(\theta \mid d) \longrightarrow q(\theta \mid d, S_n(f))$$

Other SBI approaches

We can use neural density estimators in many ways: highly flexible!

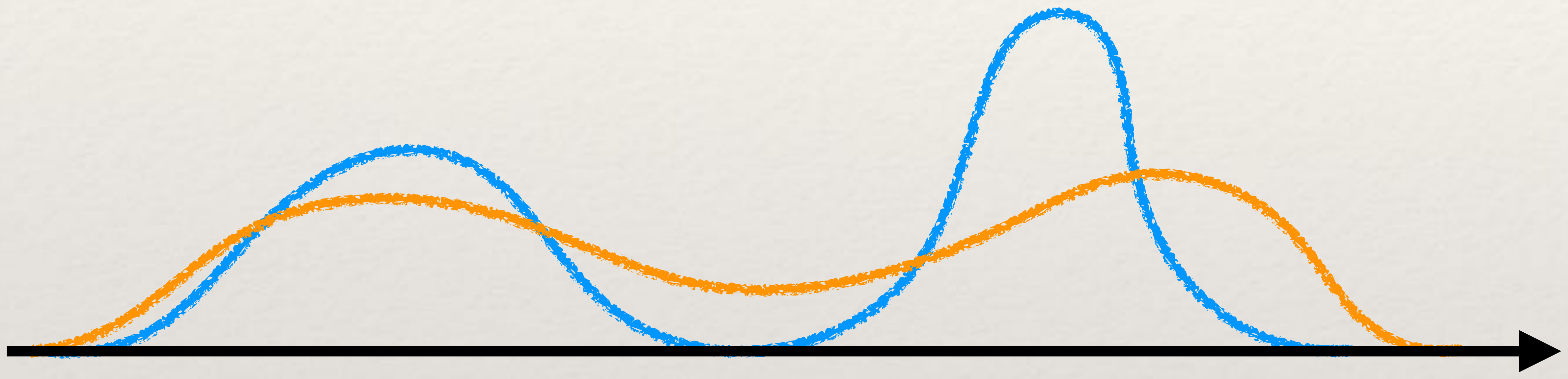
- **Neural likelihood estimation (NLE)**
  - Learn likelihood $p(d \mid \theta)$
  - Requires MCMC to obtain posterior in the end
- **Neural ratio estimation (NRE)**
  - Train a classifier to distinguish samples from joint $p(\theta, d)$ and $p(\theta)p(d)$.
  - This gives ratio $p(d \mid \theta)/p(d)$. Use MCMC to obtain posterior samples

Trivial to **marginalize parameters,** add **additional context,** different **data representations,** etc.

Dax, SRG+ (PRL 2023)

Neural importance sampling

- By combining with classical likelihood-based techniques, we can correct SBI inaccuracies using **importance sampling:**



target (prior x likelihood)

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

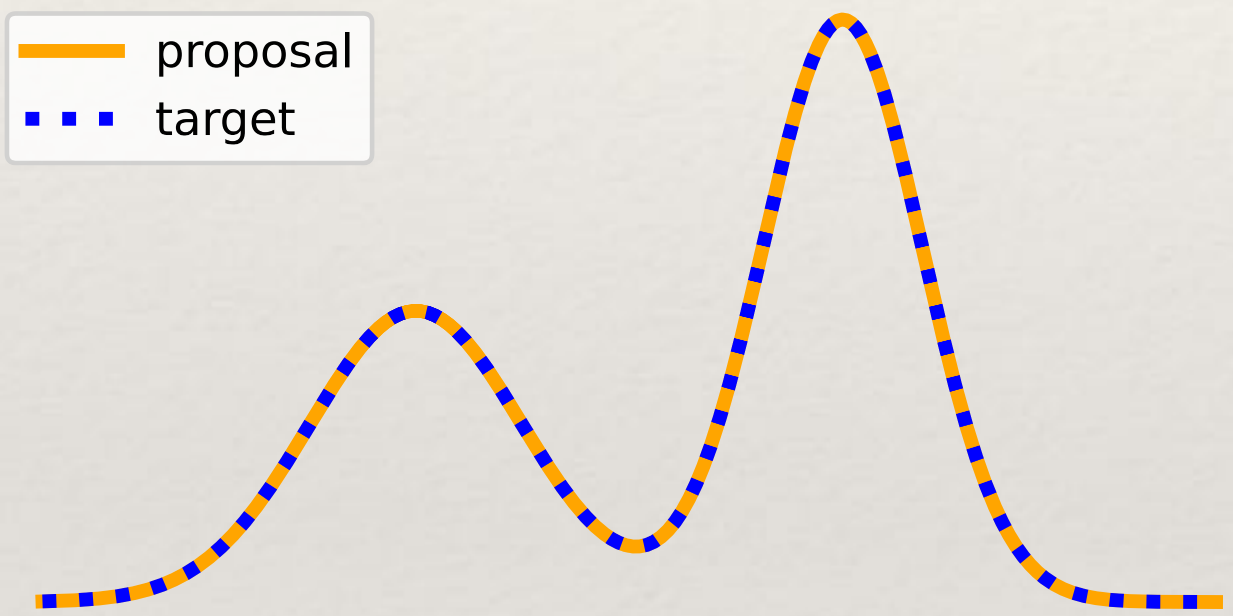
proposal (NPE)

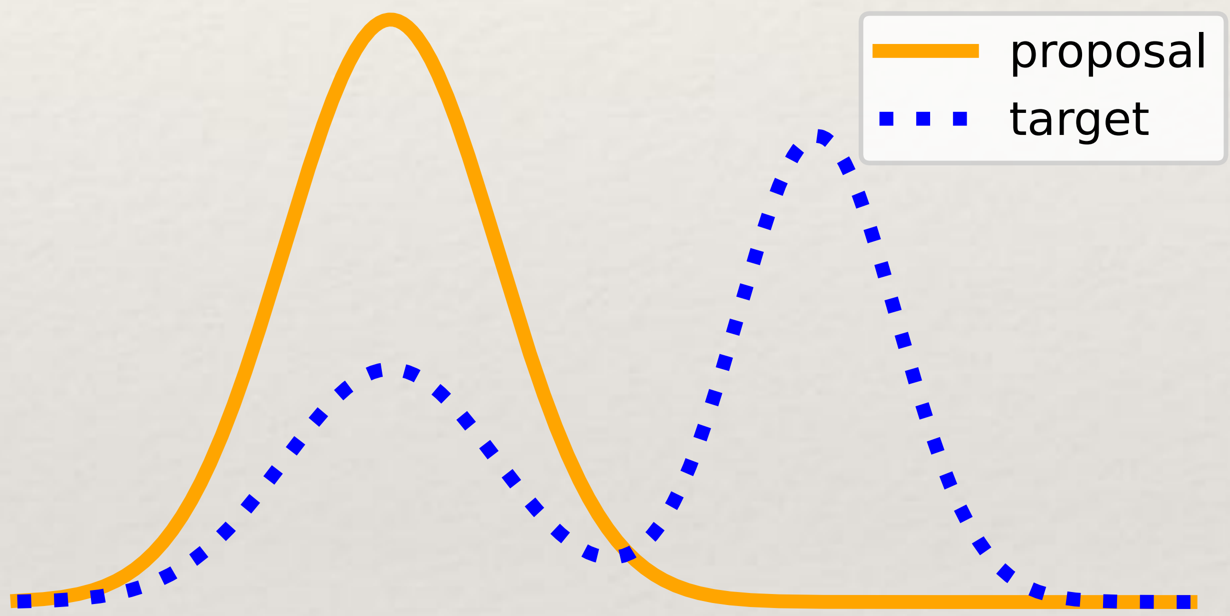
- The mean of the weights gives the **Bayesian evidence:** $p(d) = \dfrac{1}{n}\sum_{i=1}^{n} w_i$

- Variance gives the **sampling efficiency** $\epsilon = \dfrac{\left(\sum_i w_i\right)^2}{n\sum_i w_i^2}$

Neural importance sampling

- Performance depends on quality of the proposal:
  - **Perfect proposal**
    $w_i = \text{constant} \quad \Longrightarrow \quad \epsilon = 1$
  - **Poor proposal**
    $\text{large variance} \quad \Longrightarrow \quad \epsilon \approx 0$





- NPE proposals have the **mass-covering property,** making them well-suited to IS

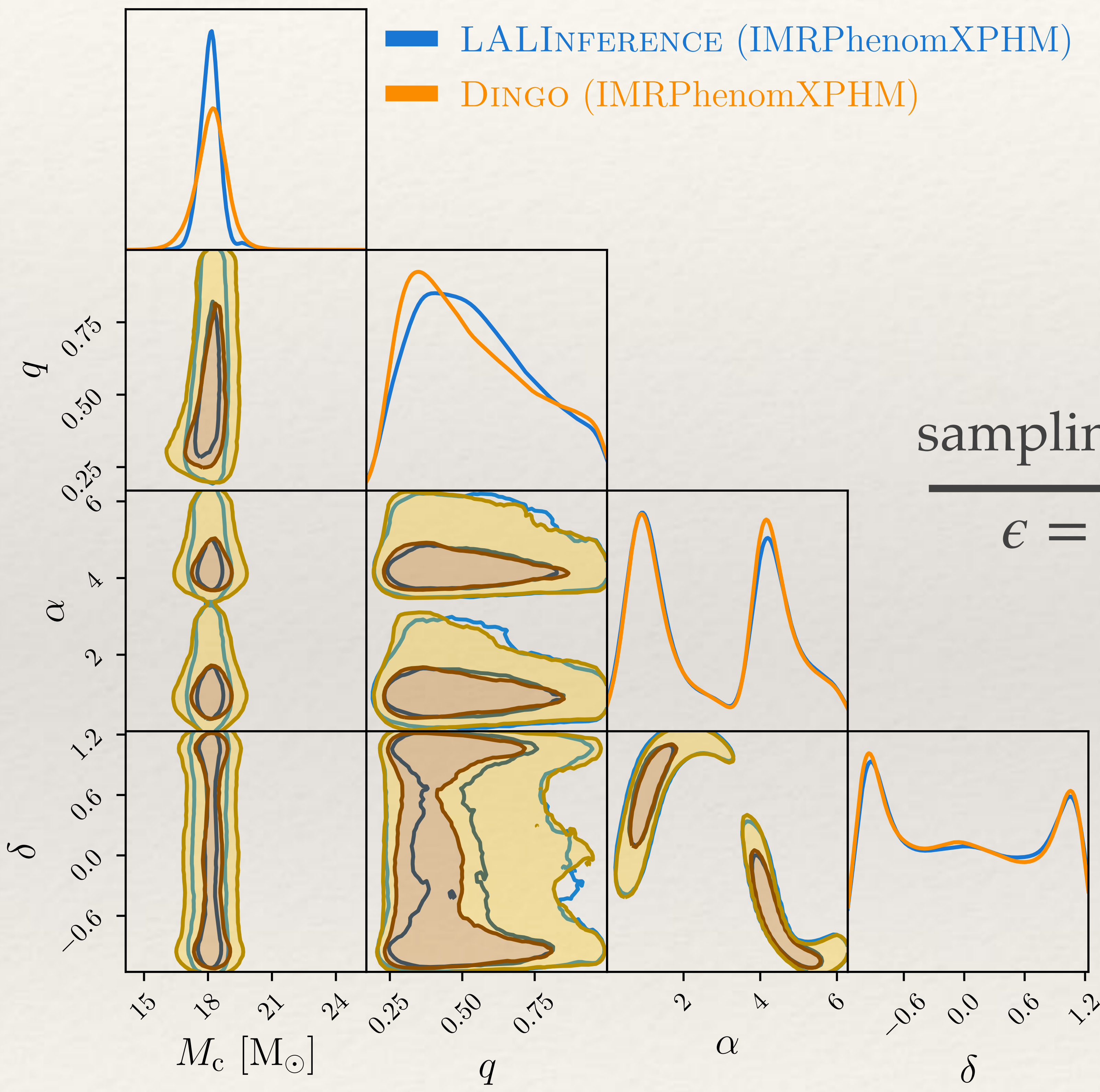
$$L[\phi] \sim D_{\mathrm{KL}}(p\,|\,q) = \int d\theta\, p(\theta\,|\,d)\, \log \frac{p(\theta\,|\,d)}{q_\phi(\theta\,|\,d)}$$

Penalty in loss if q_ϕ misses a mode!

Dax, SRG+ (PRL 2023)

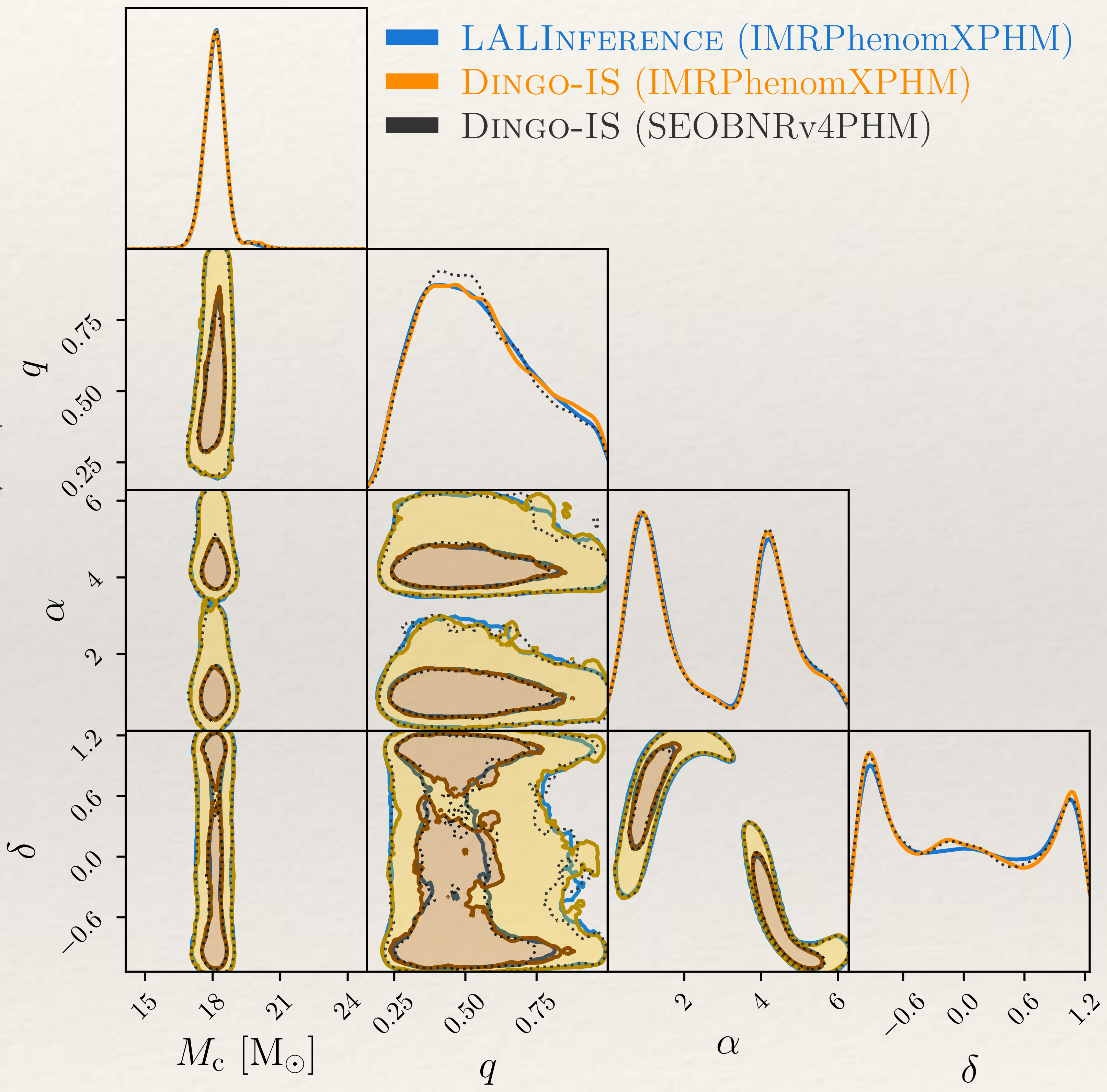
Neural importance sampling

GW151012



sampling efficiency

$\epsilon = 12.5\,\%$



Log evidence	
Dingo-IS:	-16412.88 $\boxed{\pm 0.01}$
Bilby:	-16412.73 ± 0.12

Dax, SRG+ (PRL 2023)

Validating results

- Good sampling efficiency for majority of events

Event	$\log p(d)$	ϵ
GW190408	-16178.332 ± 0.012	6.9%
_181802	-16178.172 ± 0.010	9.3%
GW190413	-15571.413 ± 0.006	22.5%
_052954	-15571.391 ± 0.005	26.3%
GW190413	-16399.331 ± 0.009	12.4%
_134308	-16399.139 ± 0.014	4.7%
GW190421	-15983.248 ± 0.008	15.3%
_213856	-15983.131 ± 0.010	9.4%
GW190503	-16582.865 ± 0.022	2.0%
_185404	-16583.352 ± 0.027	1.4%
GW190513	-15946.462 ± 0.043	0.6%
_205428	-15946.581 ± 0.017	3.4%
GW190514	-16556.466 ± 0.009	11.6%
_065416	-16556.314 ± 0.017	3.5%
GW190517	-16271.048 ± 0.027	1.3%
_055101	-16272.428 ± 0.034	0.9%
GW190519	-15991.171 ± 0.008	15.2%
_153544	-15991.287 ± 0.068	0.2%
GW190521	-16008.876 ± 0.008	13.4%
_074359	-16008.037 ± 0.015	4.2%
GW190527	-16119.012 ± 0.008	13.8%
_092055	-16118.781 ± 0.013	6.1%
GW190602	-16036.993 ± 0.006	25.0%
_175927	-16037.529 ± 0.006	23.5%
GW190701	-16521.381 ± 0.040	0.6%
_203306	-16521.609 ± 0.010	10.1%
GW190719	-15850.492 ± 0.008	13.4%
_215514	-15850.339 ± 0.011	8.0%
GW190727	-15992.017 ± 0.009	10.3%
_060333	-15992.428 ± 0.005	30.8%
GW190731	-16376.777 ± 0.005	32.6%
_140936	-16376.763 ± 0.005	31.0%
GW190803	-16132.409 ± 0.006	21.4%
_022701	-16132.408 ± 0.005	27.8%
GW190805	-16073.261 ± 0.006	20.0%
_211137	-16073.656 ± 0.007	16.6%
GW190828	-16137.220 ± 0.009	12.2%
_063405	-16136.799 ± 0.010	9.1%
GW190909	-16061.634 ± 0.011	7.4%
_114149	-16061.275 ± 0.016	3.8%
GW190915	-16083.960 ± 0.015	20.8%
_235702	-16083.937 ± 0.027	4.8%
GW190926	-16015.813 ± 0.019	2.8%
_050336	-16015.861 ± 0.009	12.1%
GW190929	-16146.666 ± 0.018	3.2%
_012149	-16146.591 ± 0.021	2.4%
GW191109	-17925.064 ± 0.025	1.7%
_010717	-17922.762 ± 0.041	0.6%
GW191127	-16759.328 ± 0.019	2.7%
_050227	-16758.102 ± 0.029	1.2%
‡GW191204	-15984.455 ± 0.015	4.2%
_110529	-15983.618 ± 0.063	0.3%
GW191215	-16001.286 ± 0.013	5.8%
_223052	-16000.846 ± 0.052	0.4%
GW191222	-15871.521 ± 0.007	16.5%
_033537	-15871.450 ± 0.005	25.8%
GW191230	-15913.798 ± 0.009	12.2%
_180458	-15913.918 ± 0.010	8.8%
GW200128	-16305.128 ± 0.013	6.1%
_022011	-16304.510 ± 0.007	18.3%
‡GW200129	-16226.851 ± 0.109	0.1%
_065458	-16231.203 ± 0.051	0.4%
GW200208	-16136.381 ± 0.007	16.6%
_130117	-16136.531 ± 0.009	11.2%
GW200208	-16775.200 ± 0.011	7.4%
_222617	-16774.582 ± 0.021	2.2%
GW200209	-16383.847 ± 0.009	12.5%
_085452	-16384.157 ± 0.025	1.6%
GW200216	-16215.703 ± 0.017	3.4%
_220804	-16215.540 ± 0.018	3.1%
GW200219	-16133.457 ± 0.011	9.6%
_094415	-16133.157 ± 0.017	4.0%
GW200220	-16303.782 ± 0.007	17.3%
_061928	-16303.087 ± 0.026	1.5%
GW200220	-16136.600 ± 0.008	13.2%
_124850	-16136.519 ± 0.037	0.7%
GW200224	-16138.613 ± 0.006	22.5%
_222234	-16139.101 ± 0.006	21.4%
‡GW200308	-16173.938 ± 0.013	6.0%
_173609	-16173.692 ± 0.025	1.7%
GW200311	-16117.505 ± 0.011	7.4%
_115853	-16117.583 ± 0.009	11.9%
‡GW200322	-16313.568 ± 0.307	0.0%
_091133	-16313.110 ± 0.105	0.1%

[Dax+ 2022]

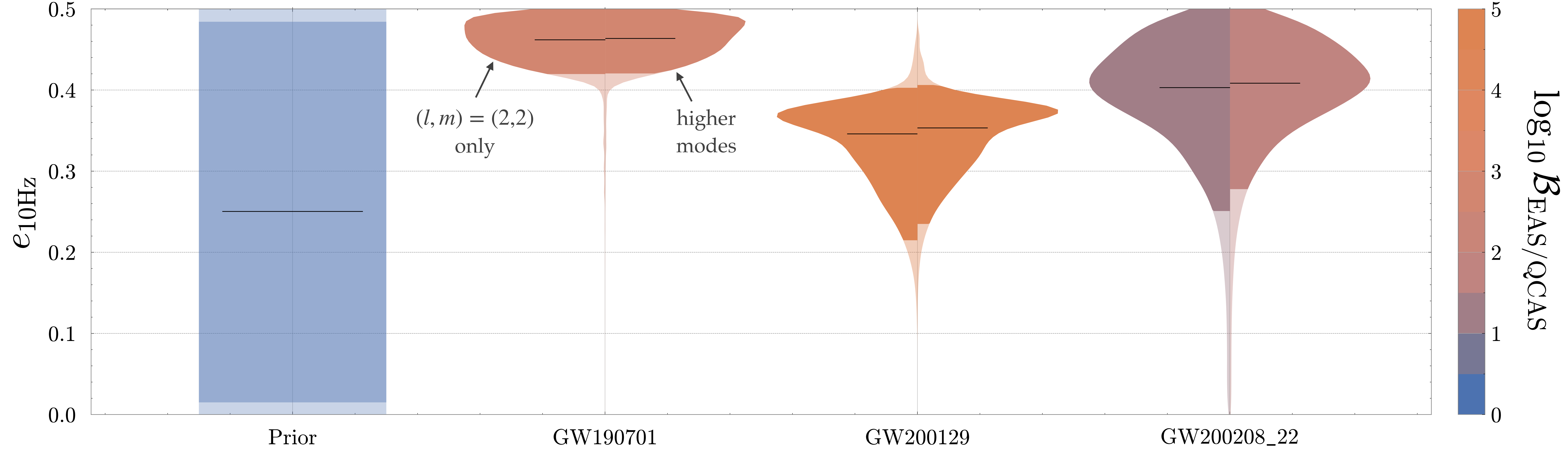
Application: Eccentric binaries

- **Use DINGO for a large study of eccentricity binaries using an expensive waveform model.**
- "Effective one body" SEOBNRv4EHM model (Ramos-Buades+, 2022).
  - Two new parameters: **eccentricity** and **relativistic anomaly**
  - Aligned spin, but includes effect of higher-order multipoles
- Costs:
  - Nested sampling: O(week) per event w/ 320 cores.
  - DINGO: O(minute) for initial samples, and O(hour) for importance sampling.

Application: Eccentric binaries

- **Three events** with evidence for eccentricity ($\log_{10}$ Bayes ≥ 1 compared to aligned spin)

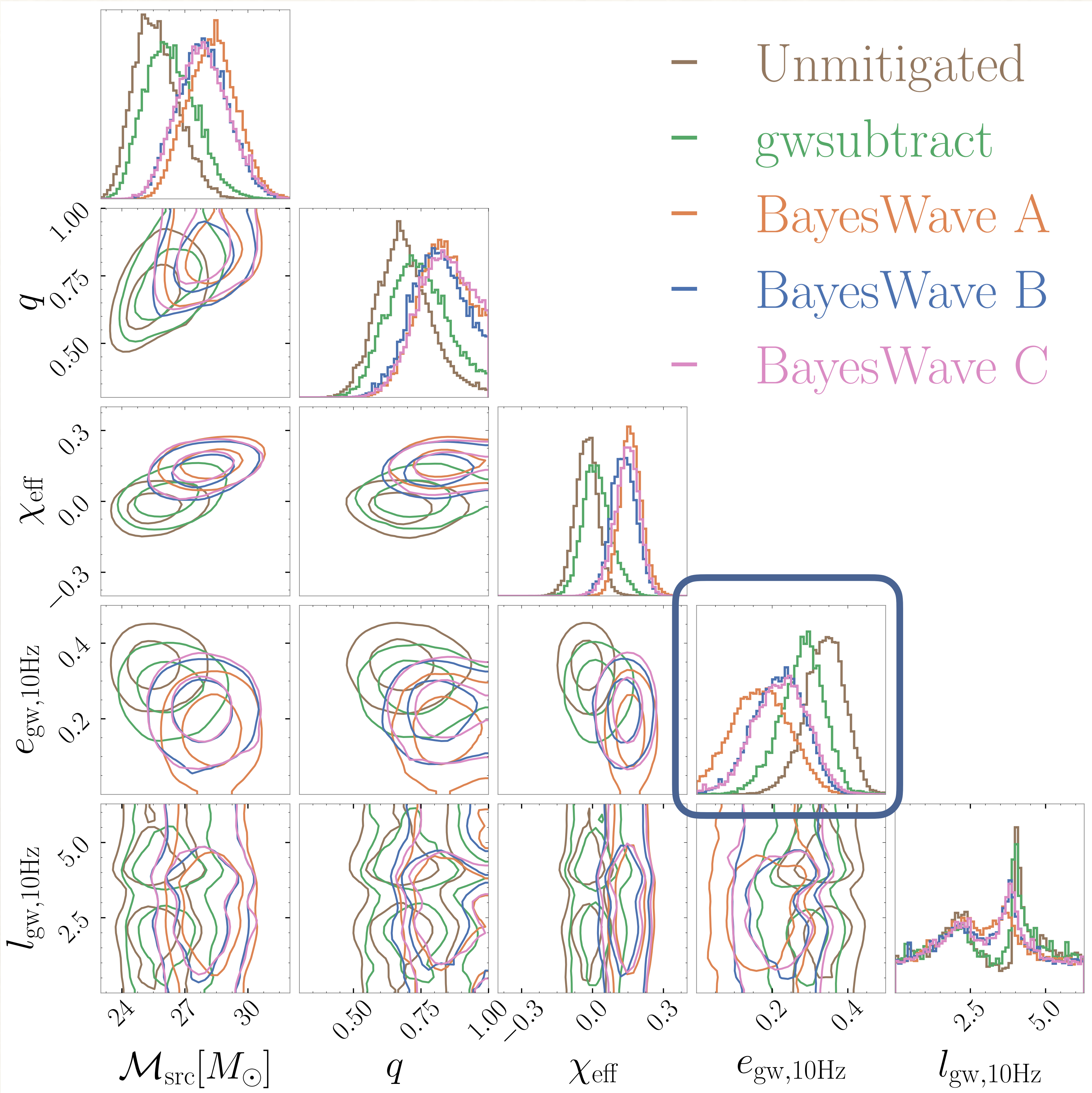
$\log_{10} \mathcal{B}_{\mathrm{HM}}$	3.0	4.7	1.8
$\log_{10} \mathcal{B}_{22}$	2.7	4.6	1.3



Application: Eccentric binaries

- Caveat for GW200129: Result highly dependent on glitch subtraction algorithm

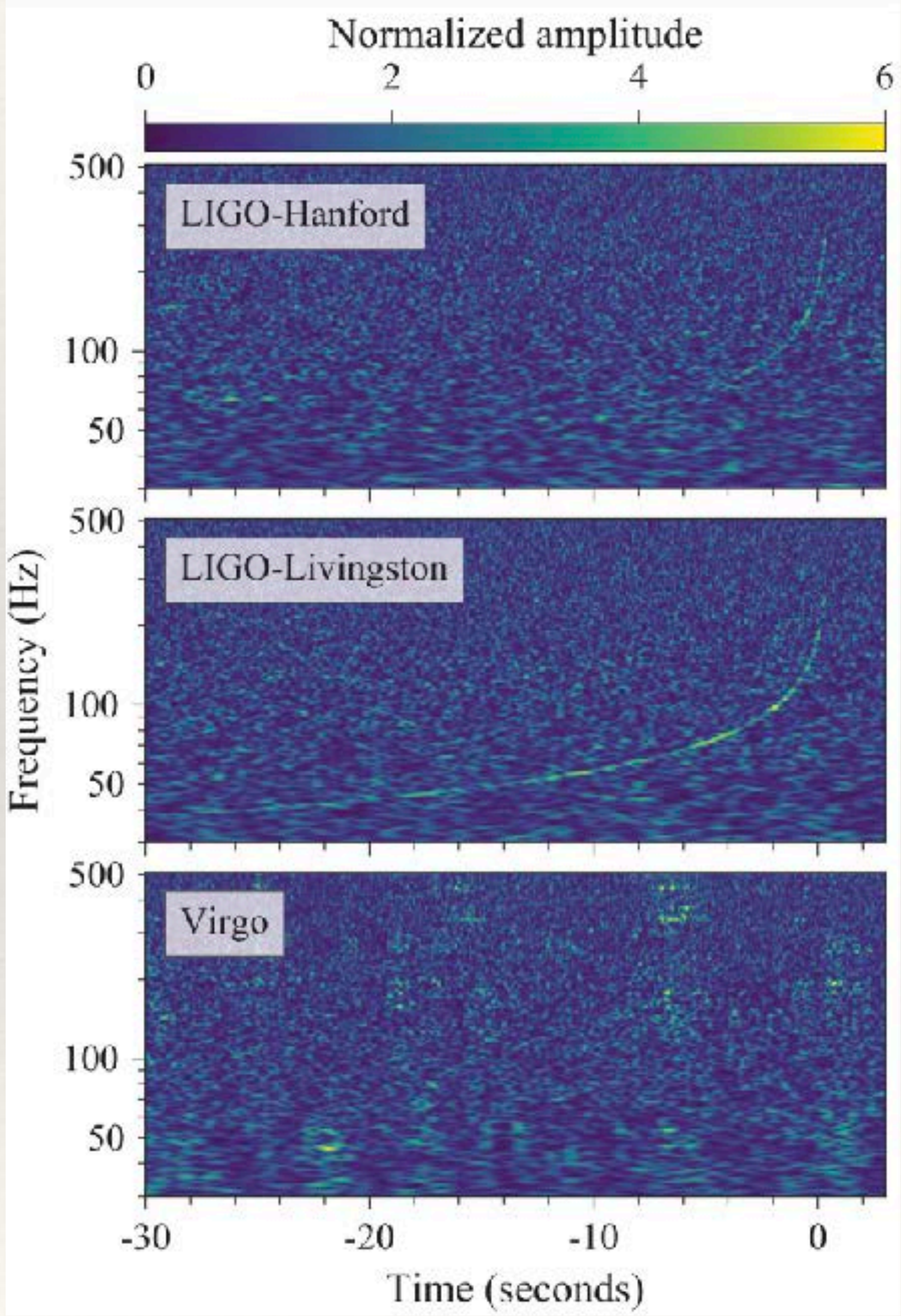
Glitch Subtraction	e Prior	SEOBNRv4 $\log_{10}\mathcal{B}$	SEOBNRv4HM $\log_{10}\mathcal{B}$	SEOBNRv4PHM $\log_{10}\mathcal{B}$	NRSur7dq4 $\log_{10}\mathcal{B}$	$e_{10\mathrm{Hz}}$	$e_{\mathrm{gw,\,10Hz}}$
			GW200129				
gwsubtract	Uniform	4.57	4.75	4.92	4.0	$0.34^{+0.11}_{-0.06}$	$0.27^{+0.10}_{-0.12}$
BayesWave A	Uniform	1.7	1.84	2.20	1.53	$0.24^{+0.10}_{-0.10}$	$0.17^{+0.14}_{-0.13}$
BayesWave B	Uniform	2.92	3.08	3.43	2.35	$0.28^{+0.09}_{-0.11}$	$0.22^{+0.12}_{-0.13}$
BayesWave C	Uniform	2.85	2.93	2.63	1.43	$0.27^{+0.09}_{-0.10}$	$0.22^{+0.13}_{-0.14}$



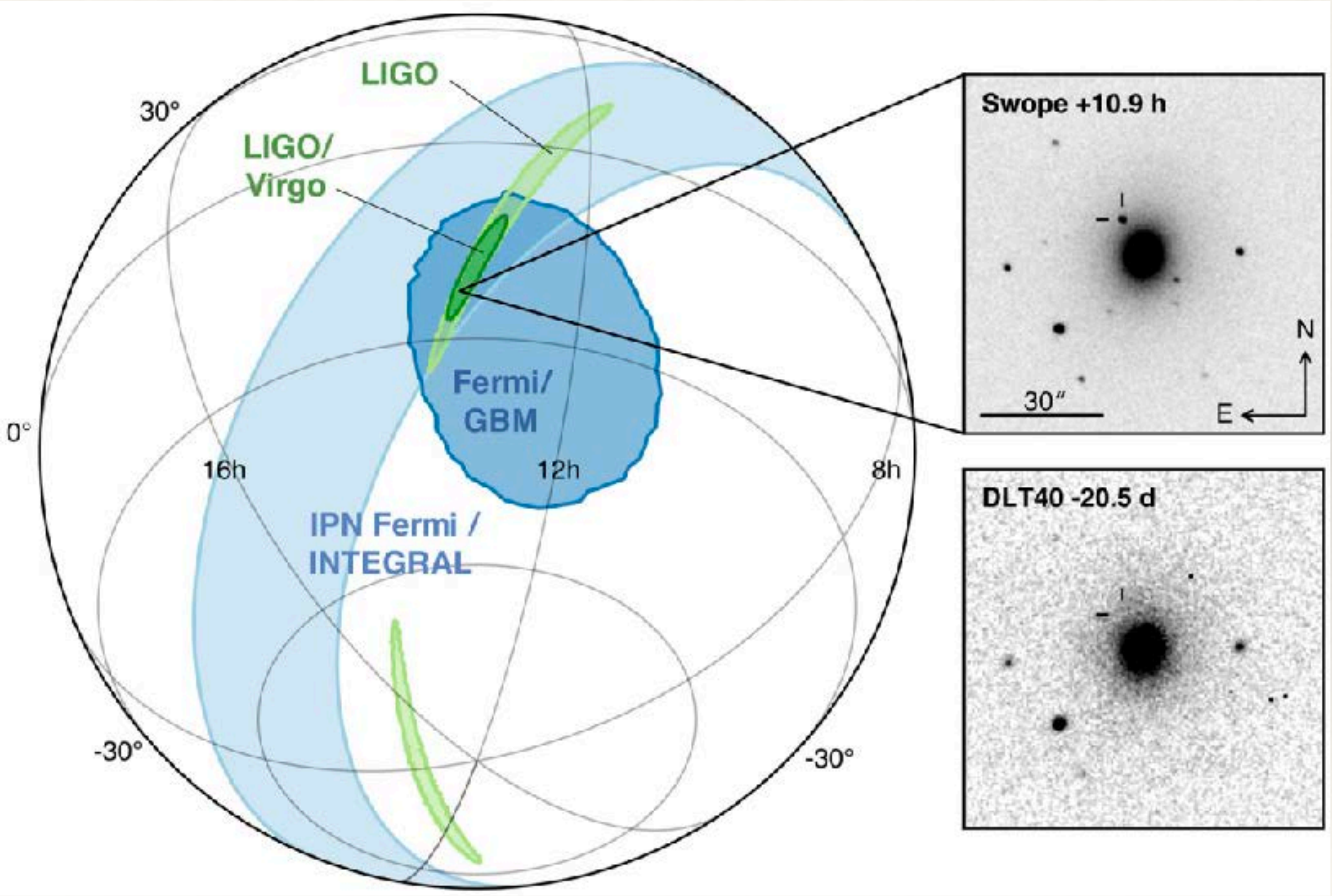
- **However, all cases show eccentricity, even in comparison to precession!**

Binary neutron stars

GW170817: Rapid sky localization enables multimessenger astrophysics



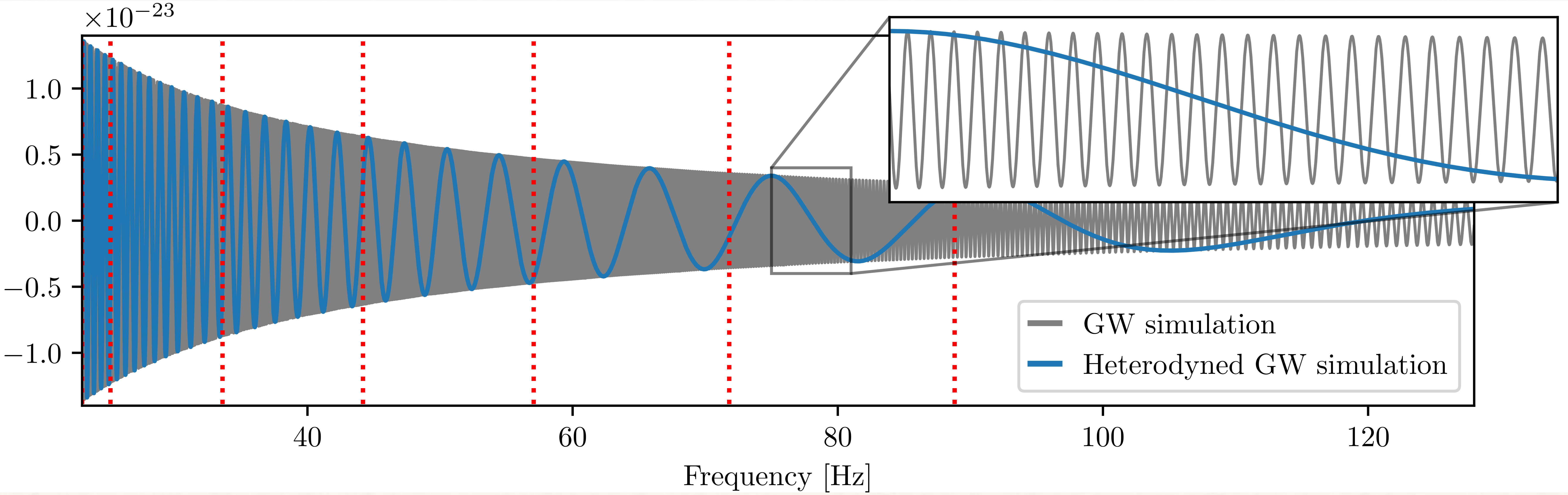
PRL 119, 161101 (2017)



ApJ Lett 848:L12 (2017)

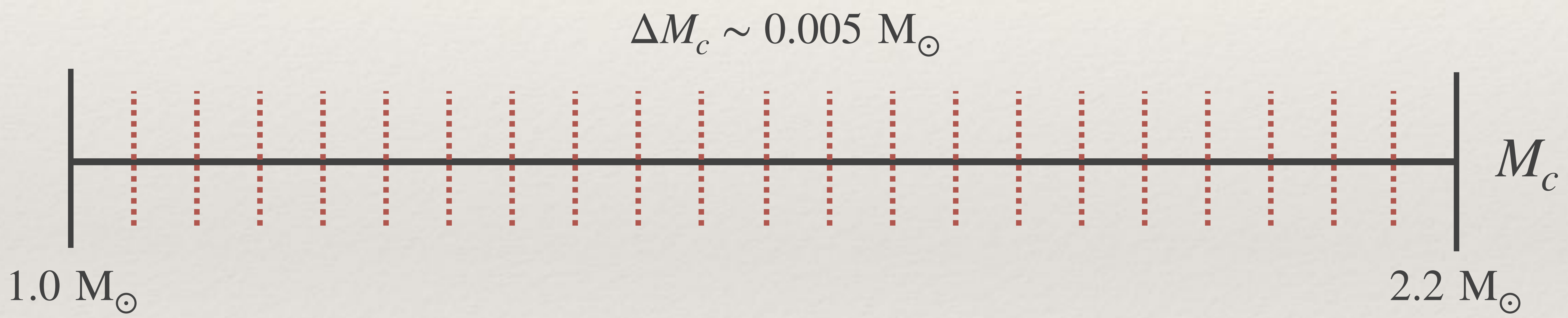
Binary neutron stars

- **Challenge:** BNS signals are longer and more complex than BBH
- **Solution:**
  1. Heterodyning (Cornish 2010) — factor out overall phase $\propto (M_c f)^{-5/3}$
  2. Multibanding (Vinciguerra+ 2017) — use reduced resolution at higher f



Binary neutron stars

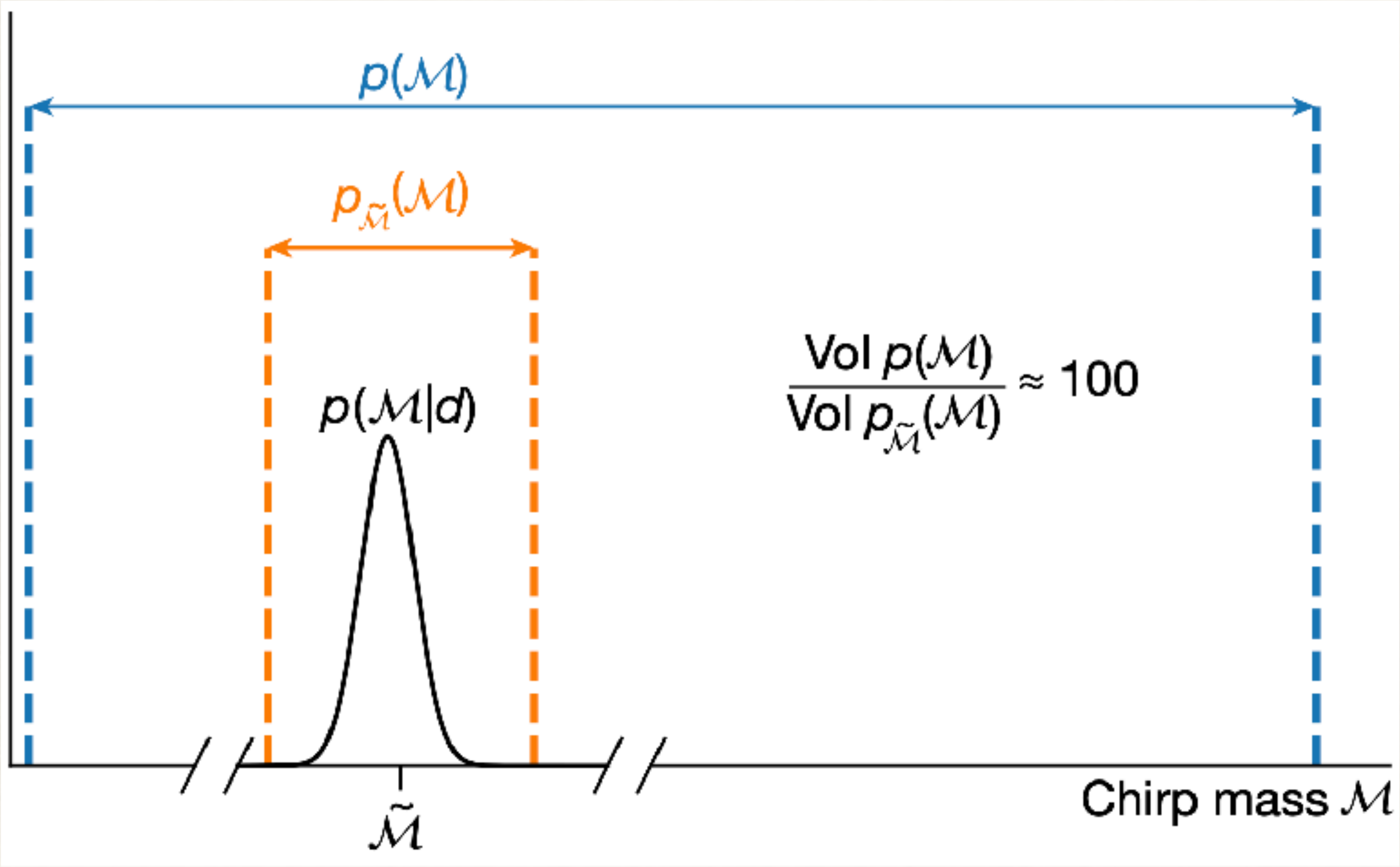
- But heterodyning depends on a **specific chirp mass** $\tilde{M}_c$. To achieve significant simplification this must be close to the true chirp mass.
- **One solution:** Divide up the prior, and train a network only over a narrow chirp mass range,

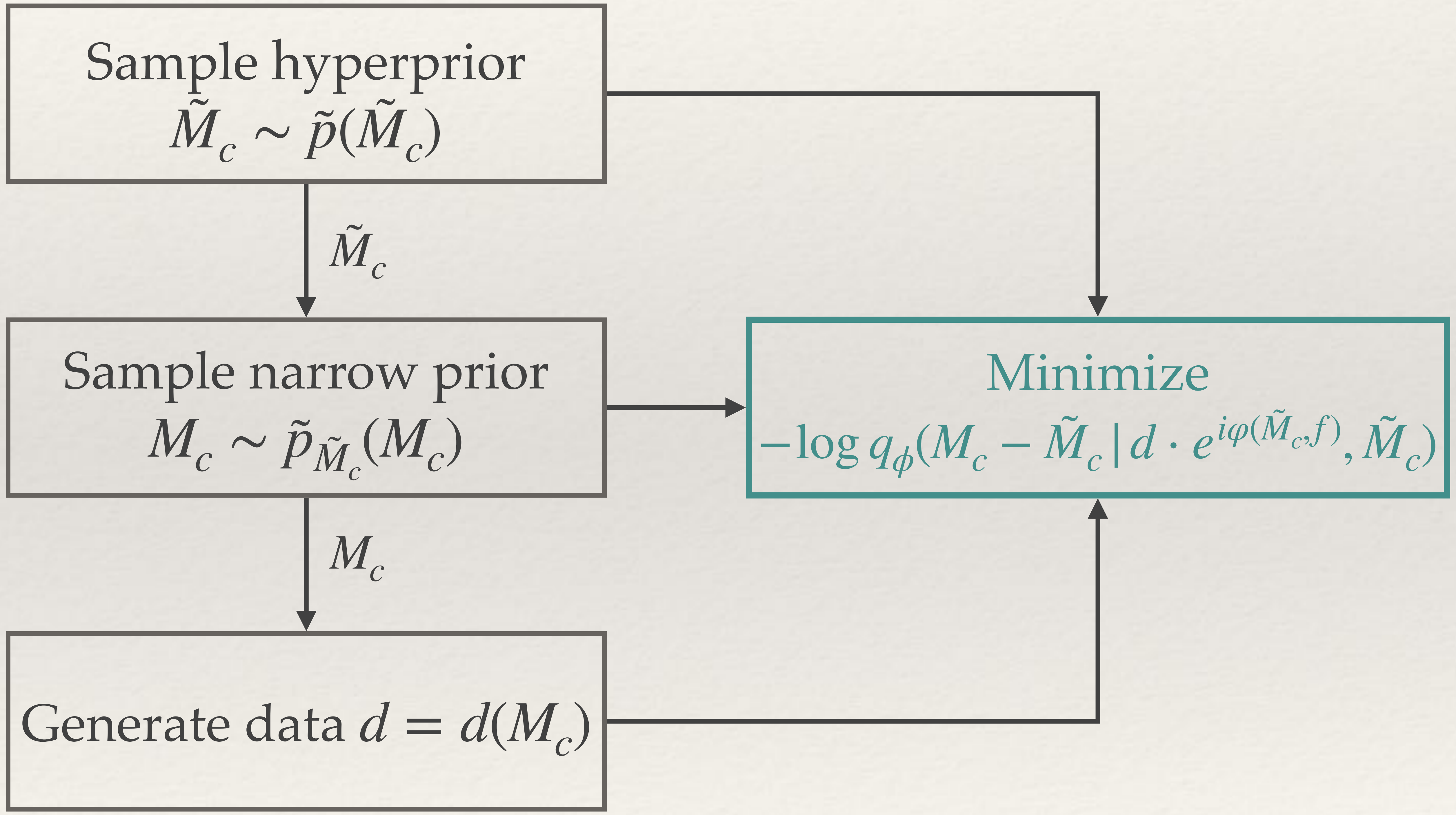


- **Impractical:** Would require too many networks.
- Instead use **prior conditioning.**

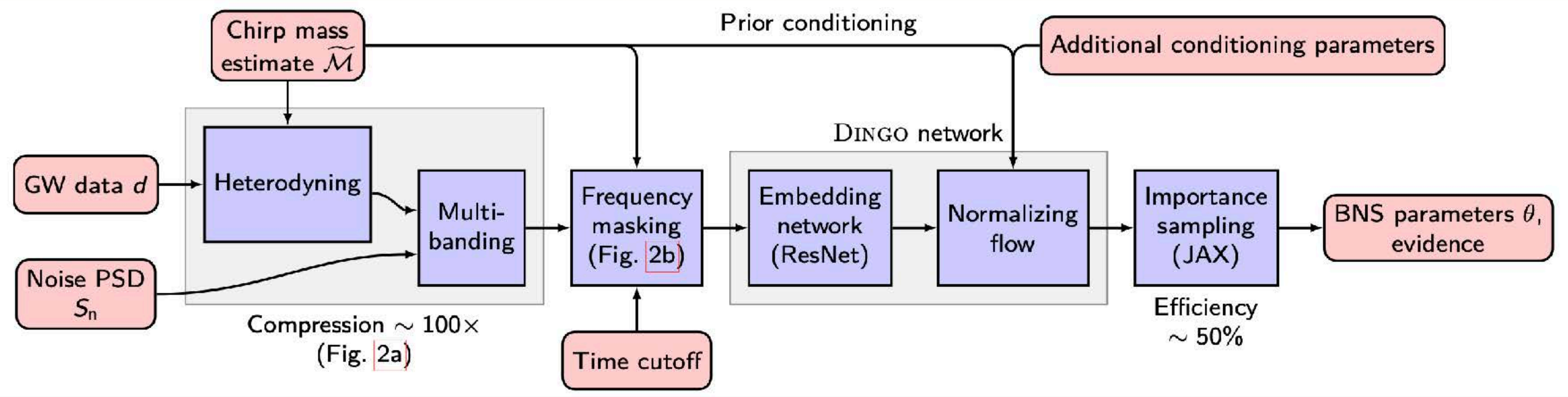
Prior conditioning

- Train network **conditioned** on narrow prior $p_{\tilde{M}_c}(M_c)$.

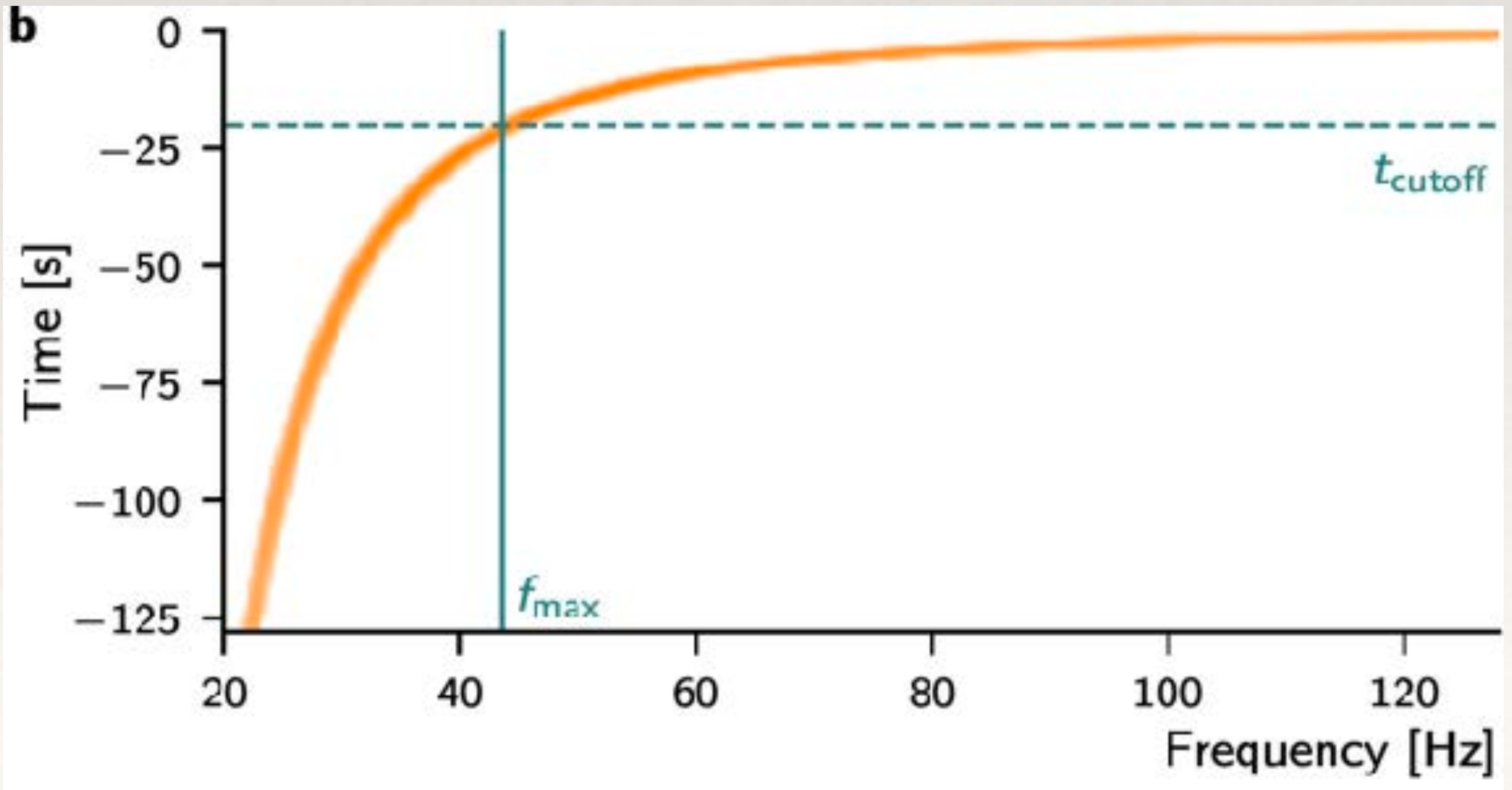




Binary neutron star inference



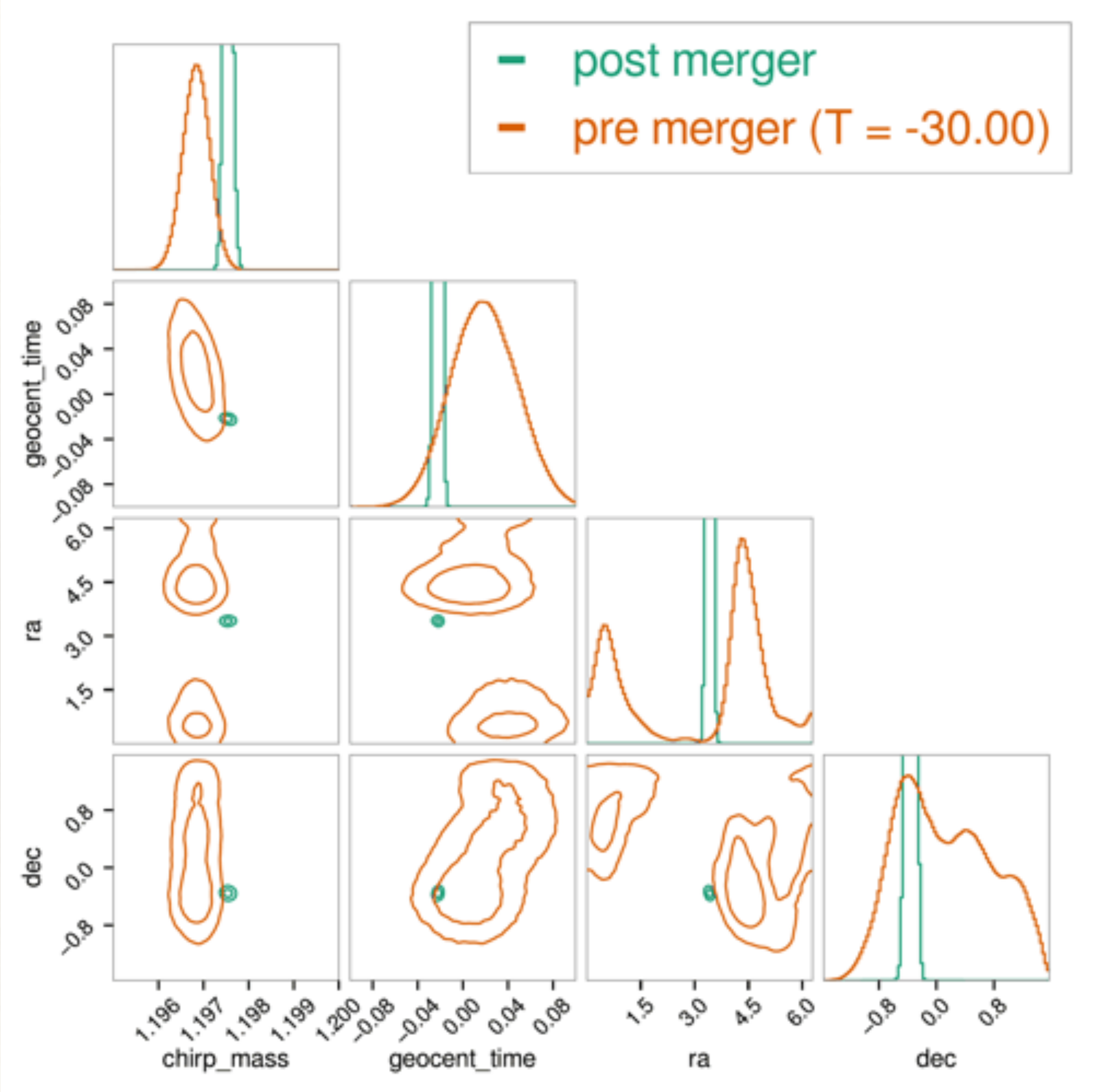
Enable pre-merger inference



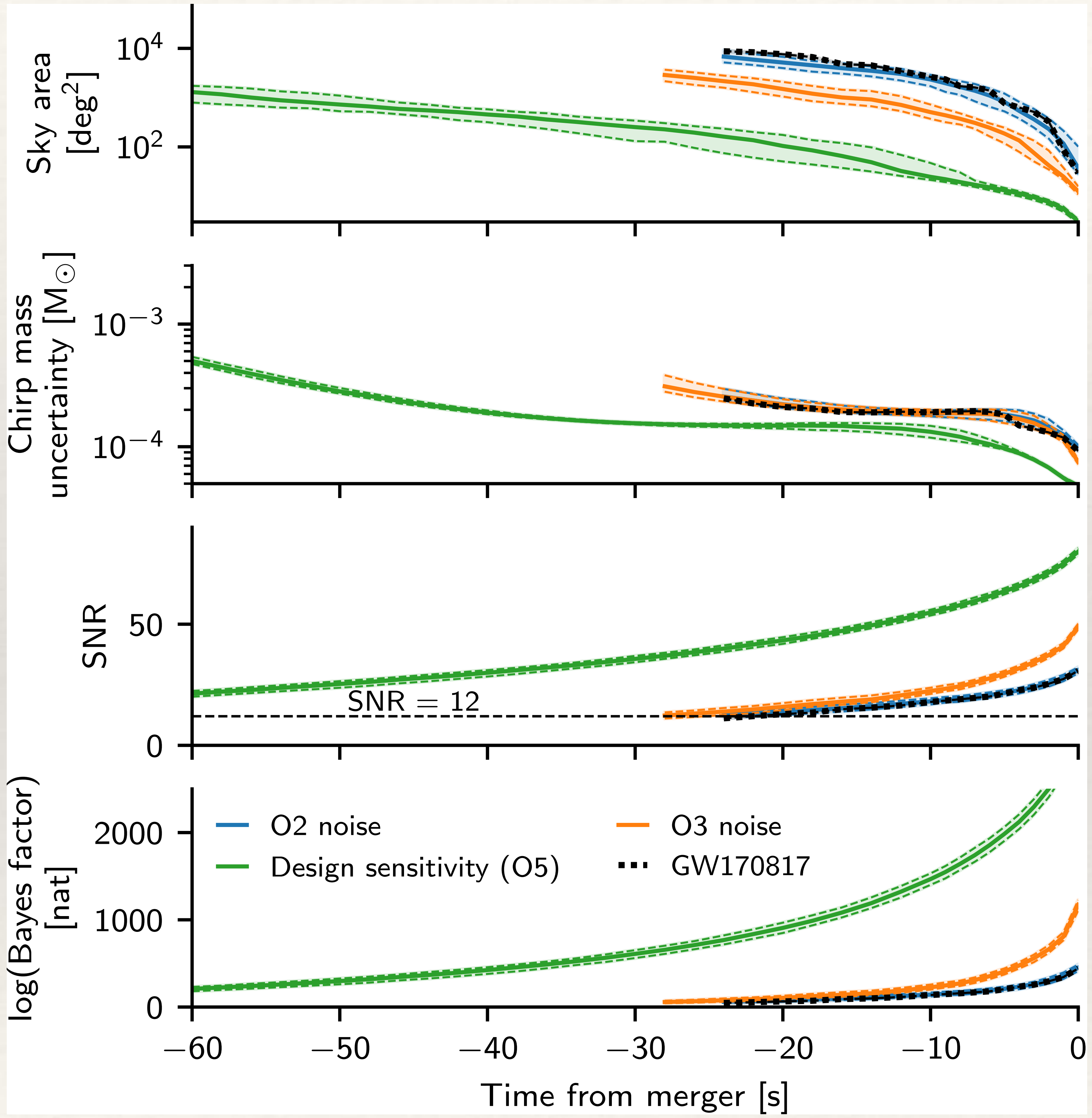
Binary neutron star inference

- **Pre-merger inference:**

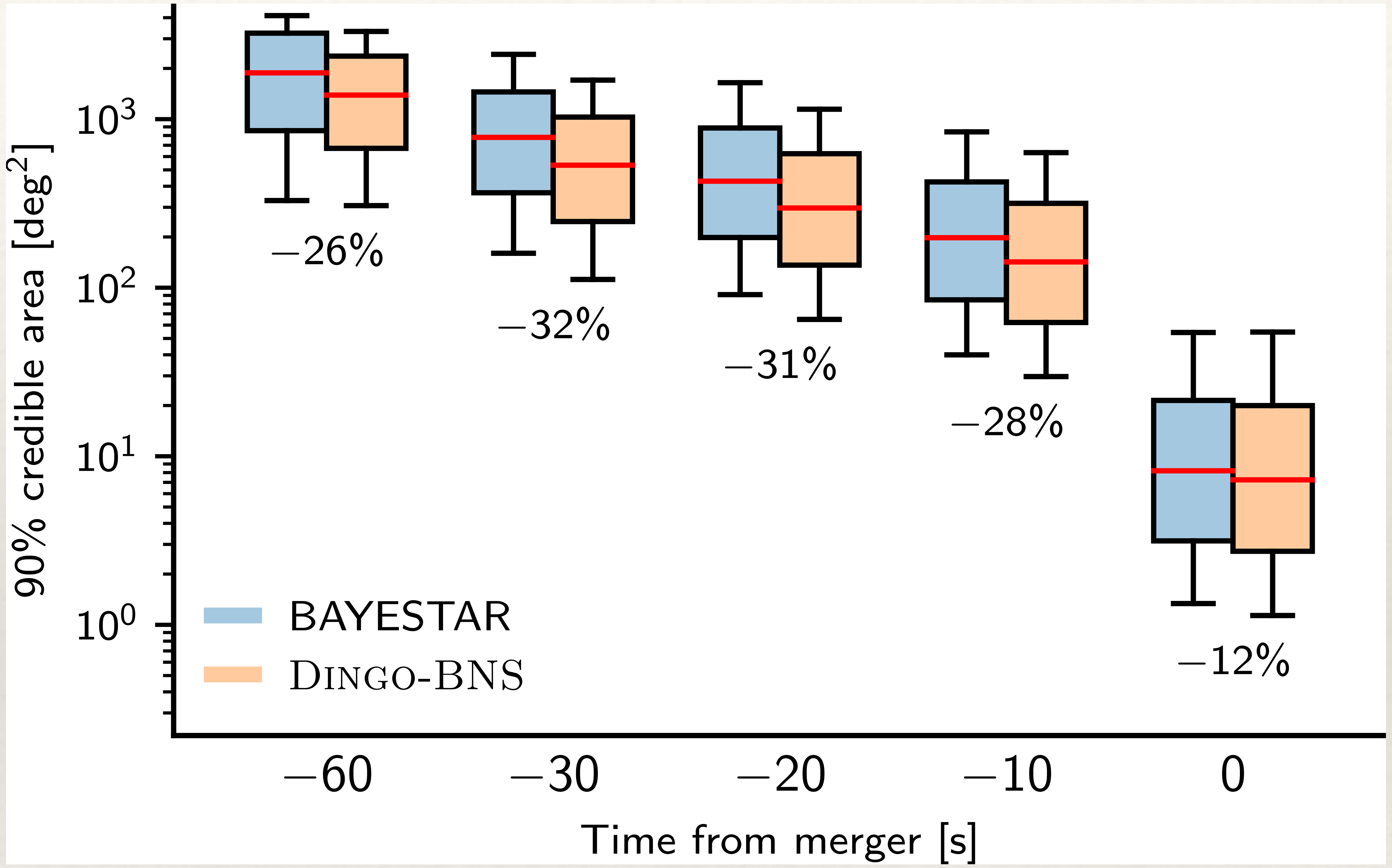
  Full Bayesian analysis takes ~ 1 second!



Pre-merger BNS



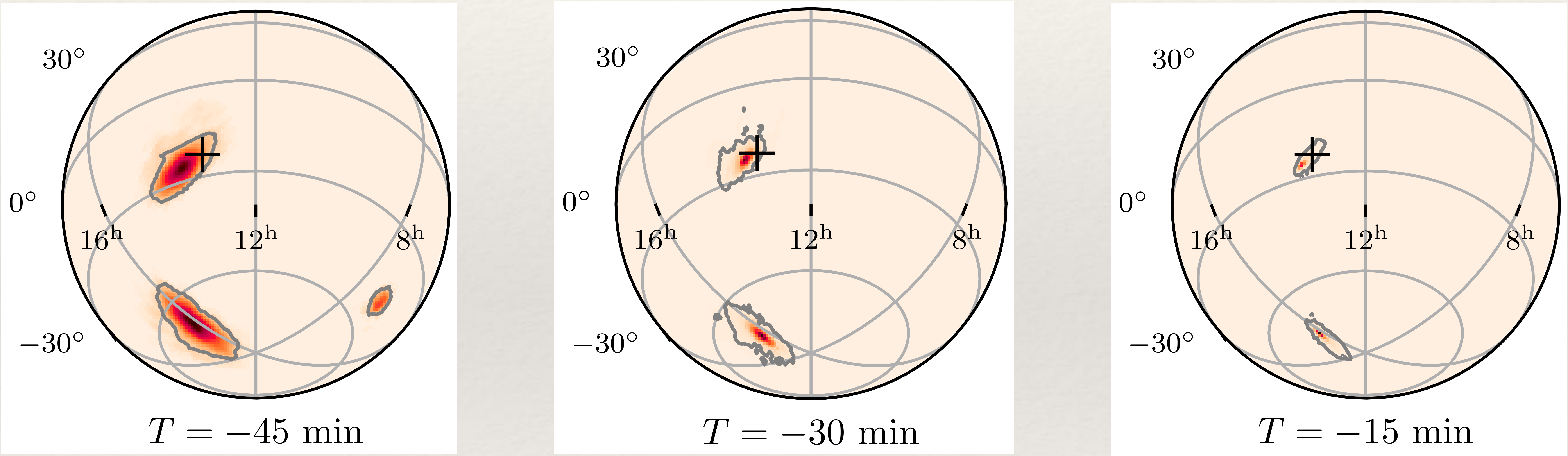
Evolution of pre-merger estimates



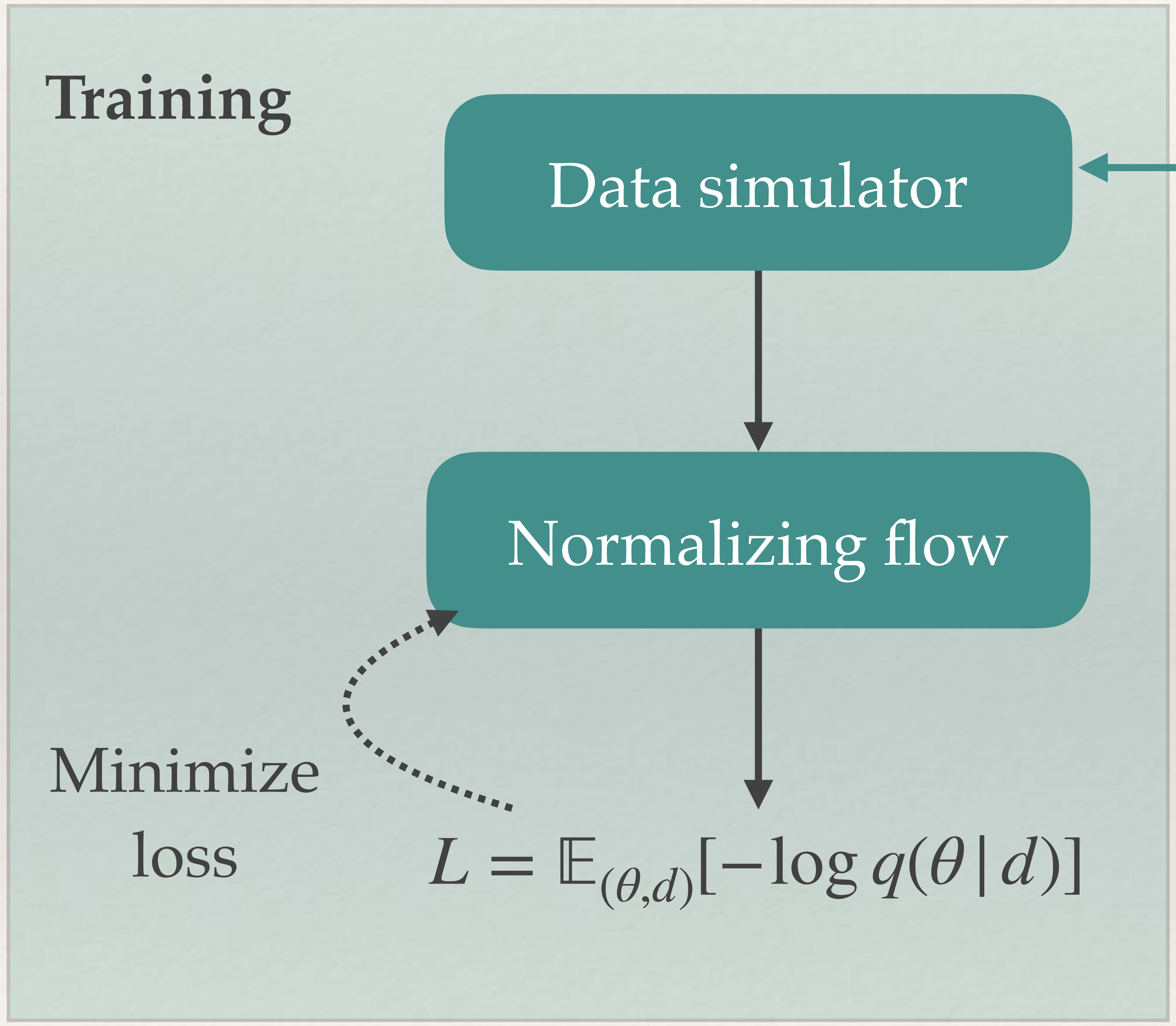
Improvements over BAYESTAR

Binary neutron stars

- With XG detectors, obtain sky position many minutes before merger



SBI is extremely general



Augment data simulator as desired, e.g.,

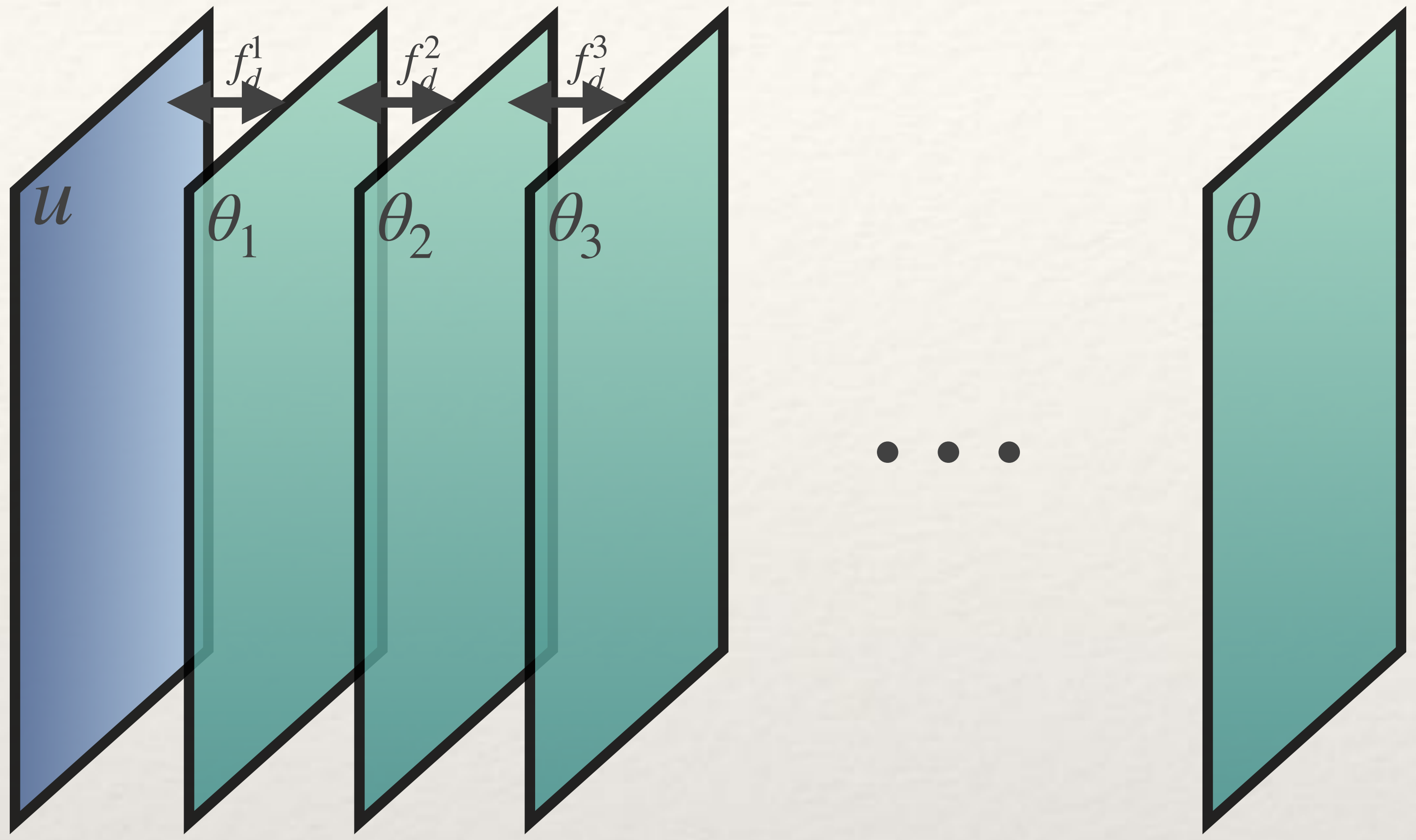
- Include varying noise spectrum to treat non-stationarity from event to event
- Inject signals into real noise
- Marginalize over unwanted parameters
- Use any data representation
- Include additional data channels

The network will learn how to incorporate this information.

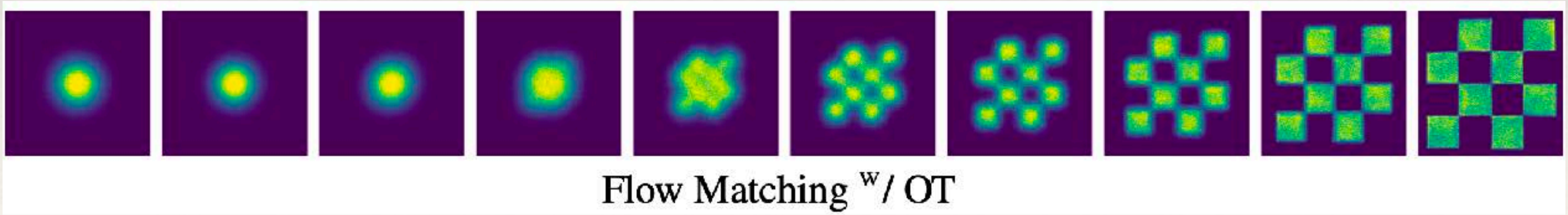
Challenge is making it work in practice!

Improved density estimators

- **Discrete** normalizing flow



- $\longrightarrow$ **continuous normalizing flow** / **flow matching** [Lipman+ (ICLR 2023)]



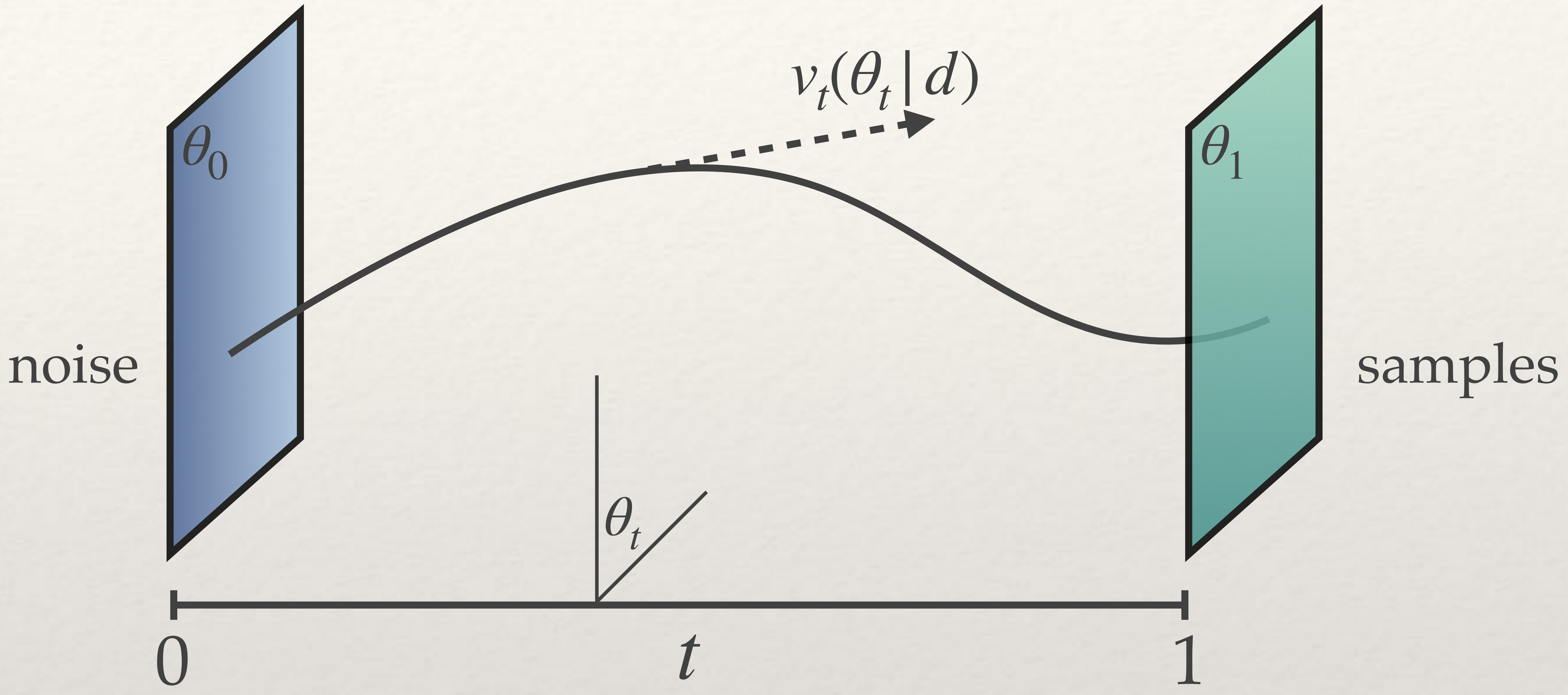
$t = 0$ $\longrightarrow t$ $t = 1$

Continuous normalizing flows

- **Sample trajectories defined by vector field**

$$\frac{d\theta_t}{dt} = v_t(\theta_t \,|\, d)$$

neural network



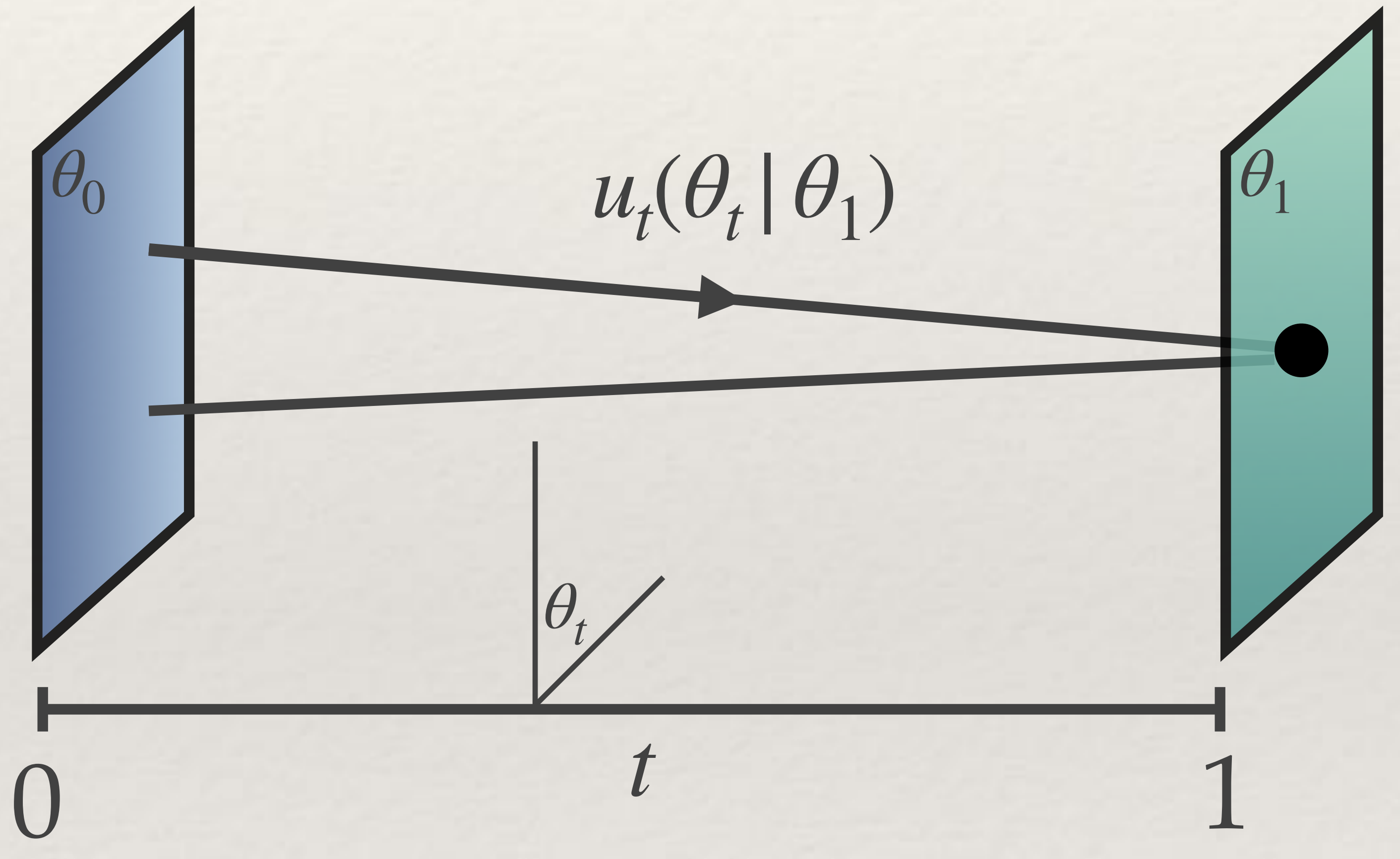
- Density satisfies **transport equation** $\partial_t q_t = \nabla_{\theta_t} \cdot (q_t v_t)$ [cf. diffusion models - stochastic differential equations]

- However, expensive to sample and evaluate density, since many network evaluations required. Makes training with loss $L = \mathbb{E}_{(\theta,d)}[-\log q(\theta \,|\, d)]$ impractical.

Flow matching

- Lipman+ (ICLR 2023): Directly regress vector field using **sample-conditional flow matching**
  - If we knew a target trajectory (u_t, p_t) that gives rise to an approximation to posterior $p(\theta_1 \mid d)$, **flow match** with $L_{\mathrm{FM}} = \mathbb{E}_{(d,\theta_t,t)} \|v_t(\theta_t \mid d) - u_t(\theta_t \mid d)\|^2$
  - Instead, consider a single point θ_1 and a simple conditional trajectory $(u_t(\theta_t \mid \theta_1), p_t(\theta_t \mid \theta_1))$

    E.g., optimal transport

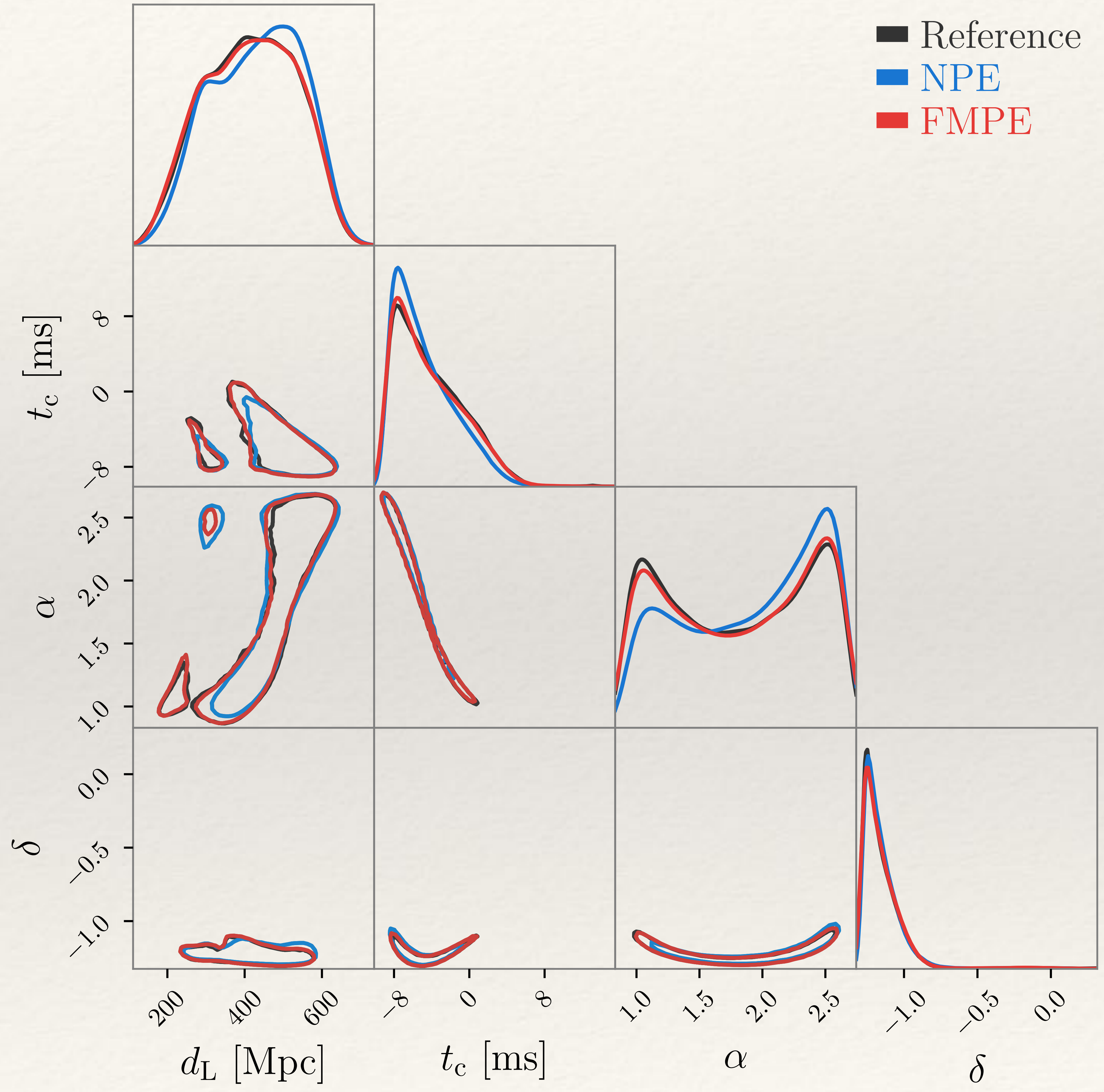


  - Remarkably, **matching to the sample-conditional path** yields a vector field that gives the marginal path

$$L_{\mathrm{SCFM}} = \mathbb{E}_{(d,\theta_1,t,\theta_t)} \|v_t(\theta_t \mid d) - u_t(\theta_t \mid \theta_1)\|^2$$

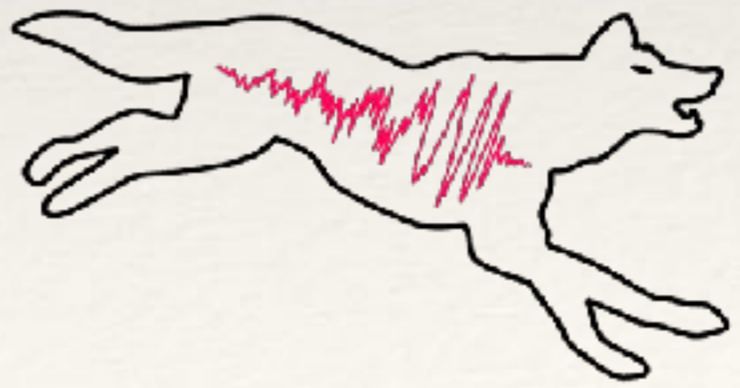
Flow matching posterior estimation (FMPE)

- We applied flow matching to posterior estimation.
- Proved that under reasonable assumptions, **mass covering property holds.**
- **Outperforms NPE with discrete flows**
  - **Faster training, better scaling to large networks**



Conclusions

- Simulation-based inference can deliver **fast and accurate** PE for gravitational waves.
  - Enabled new **eccentricity** studies, finding evidence in three events.
  - For **binary neutron stars,** prior conditioning enables O(second) pre- and post-merger inference.
  - New architectures promise to bring ever-improving performance.
- Going forwards, efforts focused on (1) extending to new sources and observatories, (2) training on realistic noise, and (3) building flexible networks.



Next: Tutorial