*Stats meets ML — September 10, 2024*

# Simulation-Based Machine Learning for Gravitational-Wave Analysis

**Maximilian Dax**

MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

*Stats meets ML — September 10, 2024*

# Simulation-Based Machine Learning for Gravitational-Wave Analysis

**Maximilian Dax**

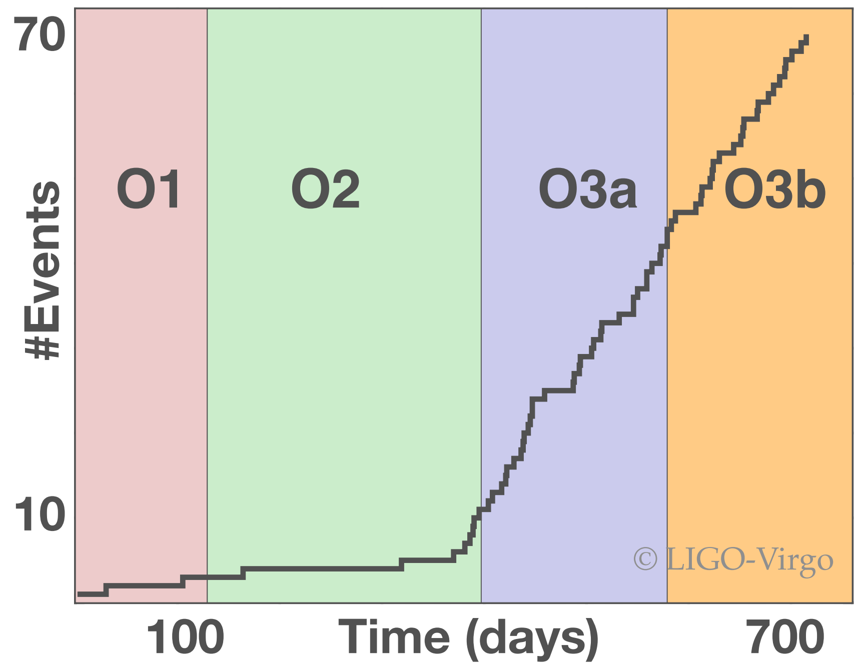
MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

- Inverse problems & simulation-based inference (SBI)
- SBI for gravitational wave inference
- Summary

# Motivation

## Why GWs need ML

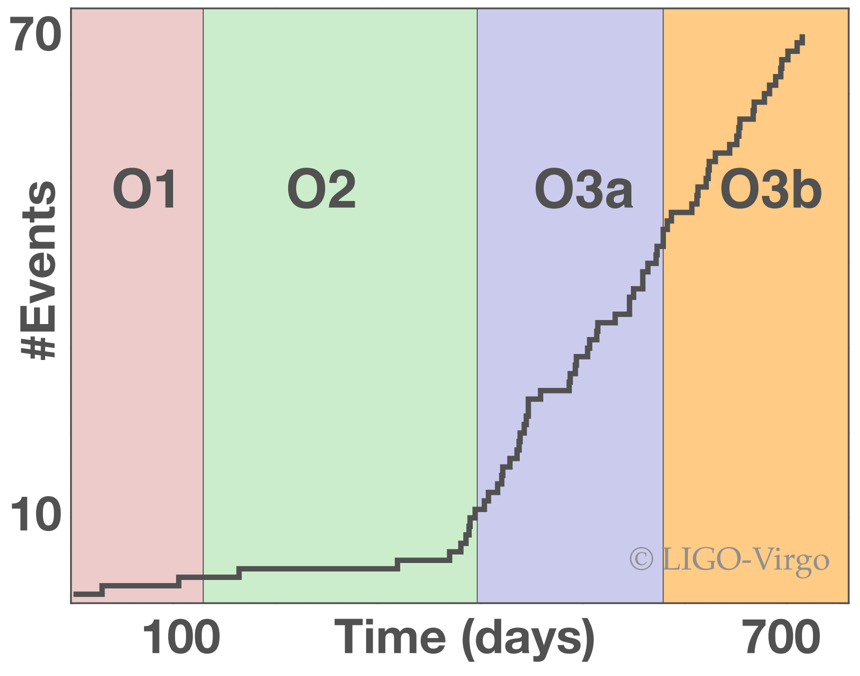
Increasing event rate



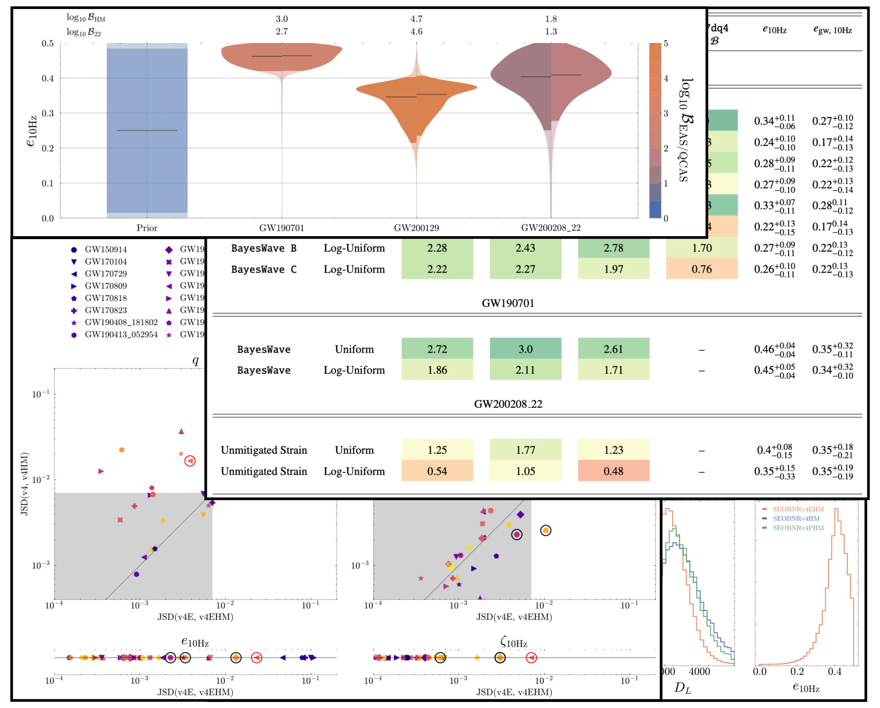
# Motivation

## Why GWs need ML

Increasing event rate



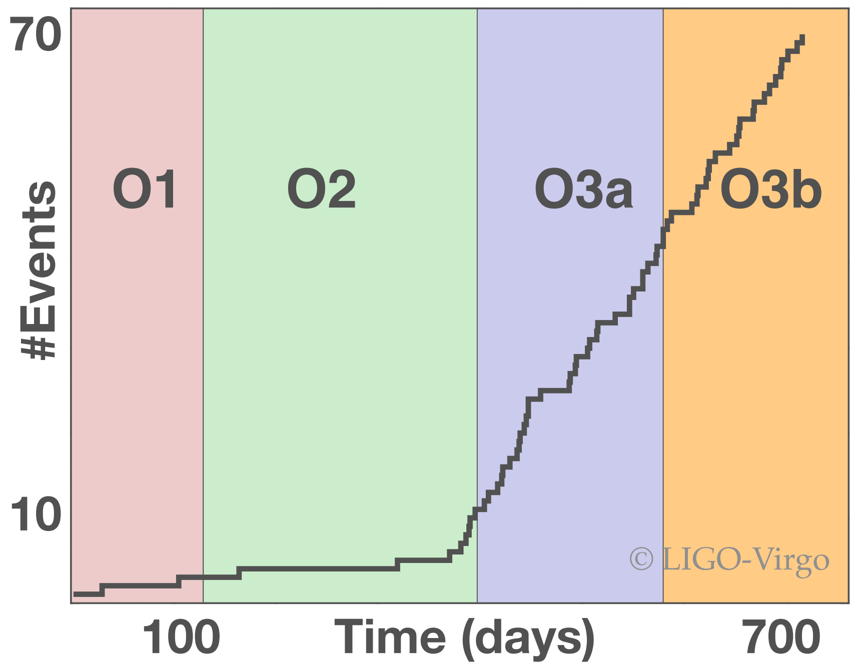

Large-scale analyses



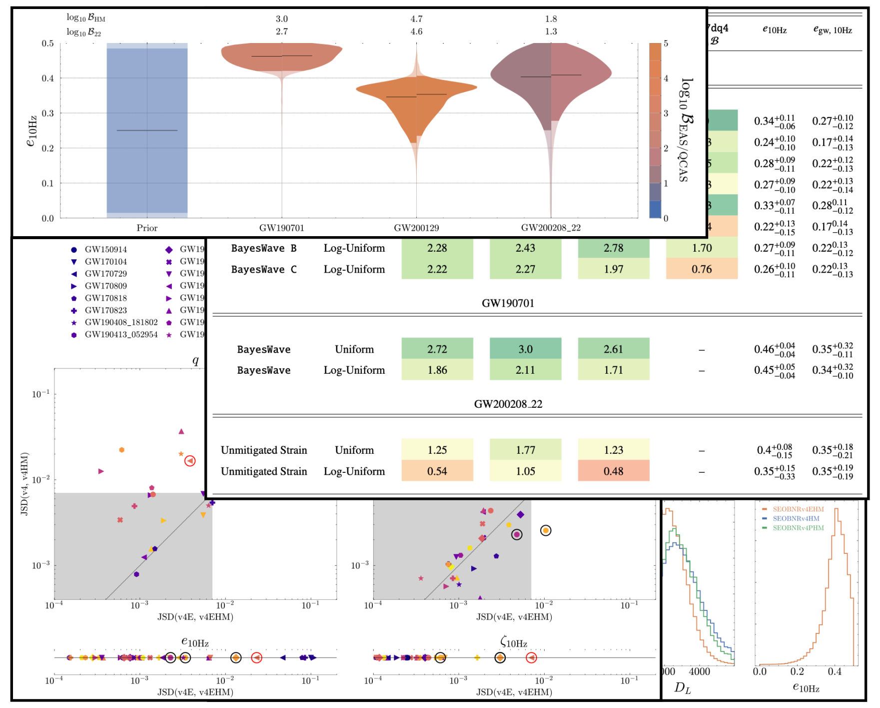
# Motivation

## Why GWs need ML

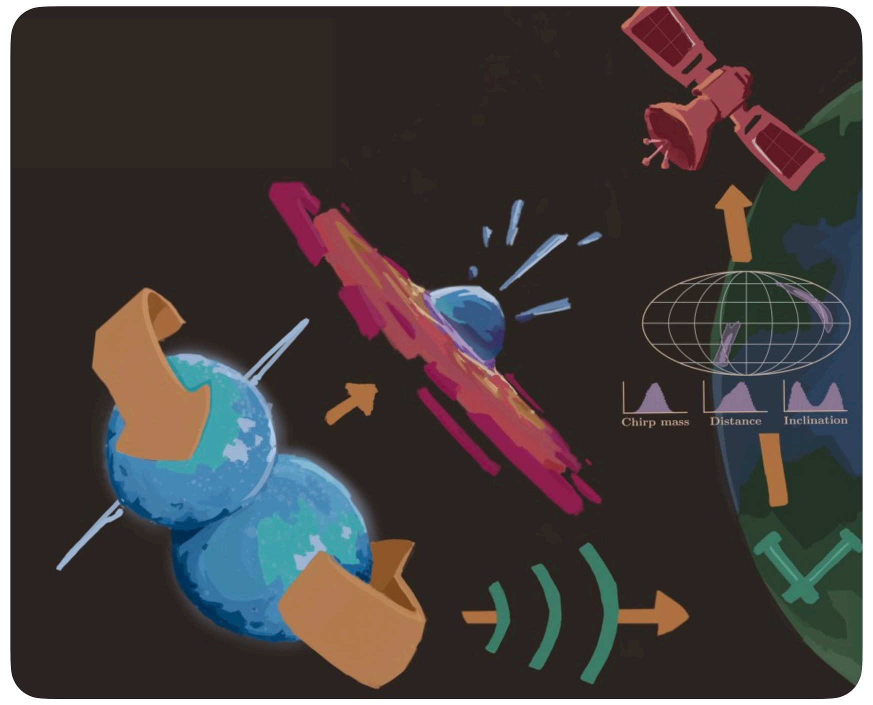
Increasing event rate




Large-scale analyses



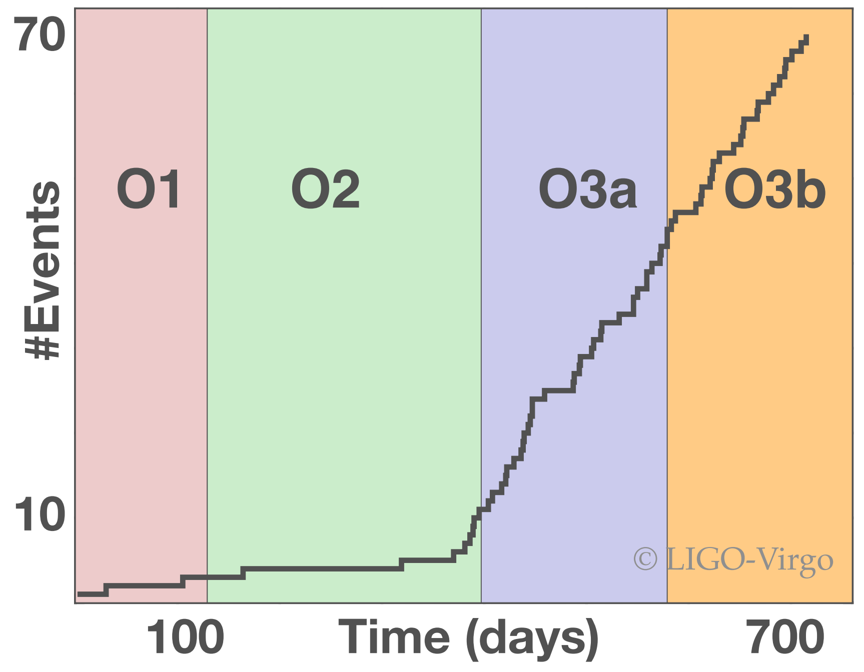
Follow-up searches



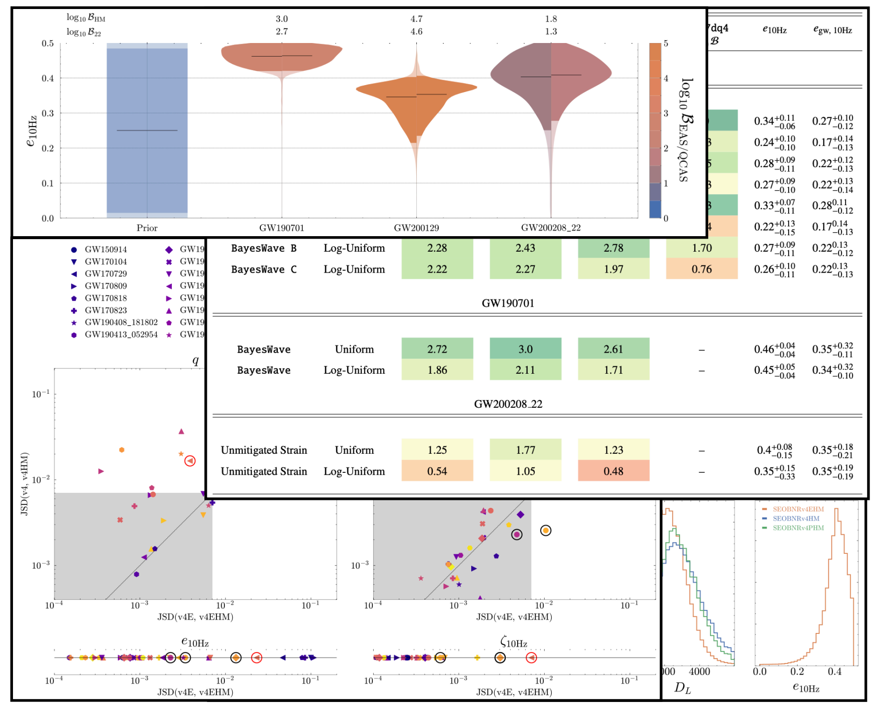
# Motivation

## Why GWs need ML

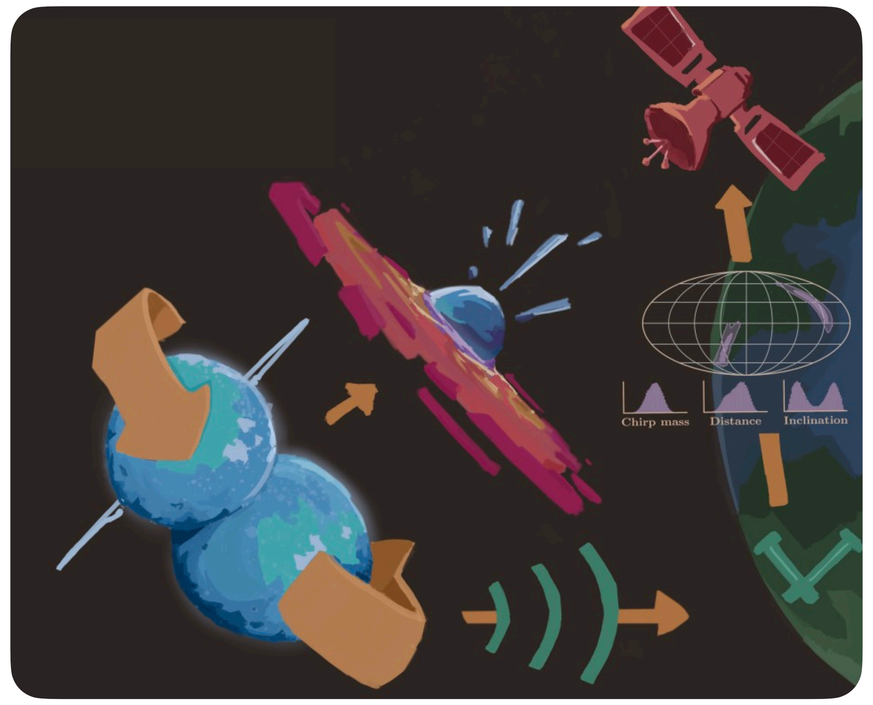
Increasing event rate




Large-scale analyses



Follow-up searches



## Why ML needs GWs

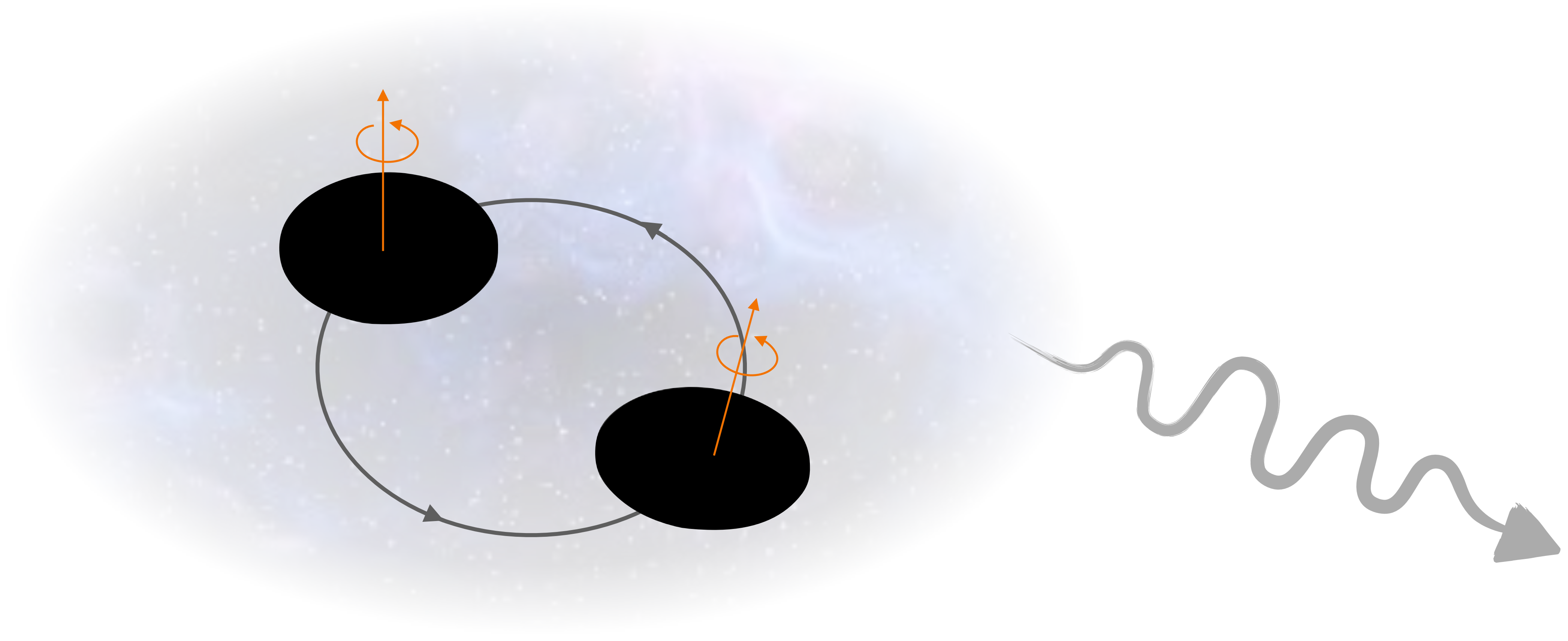
Strict requirements for **accuracy**, **reliability** and **interpretability**

**Complexity** of GW data

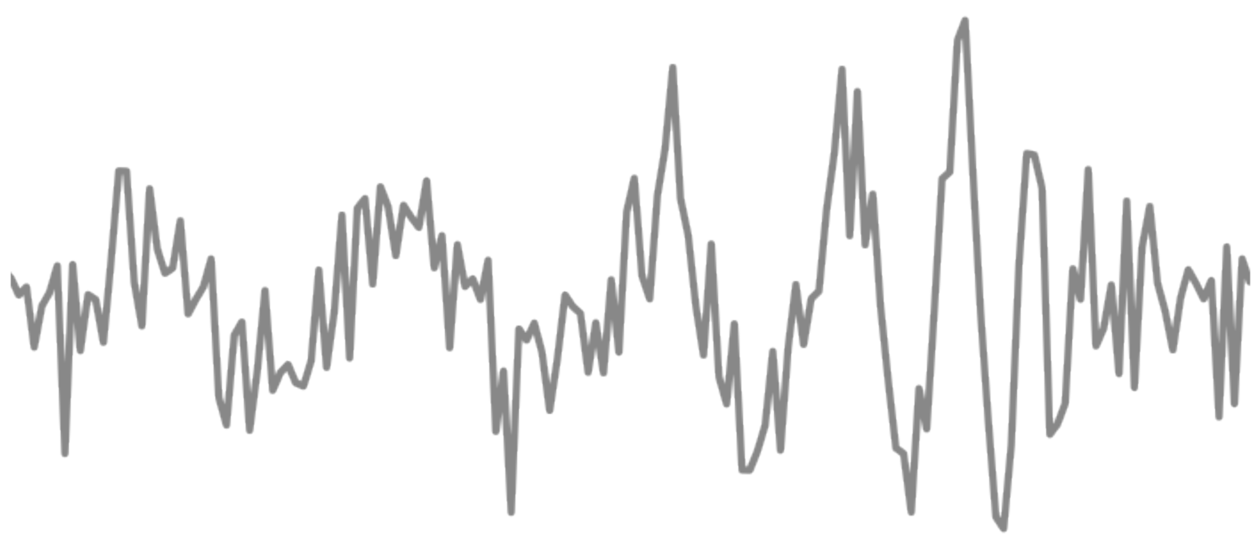
⇒ GW data analysis pushes existing ML past its limits

# Inverse Problems & Simulation-Based Inference

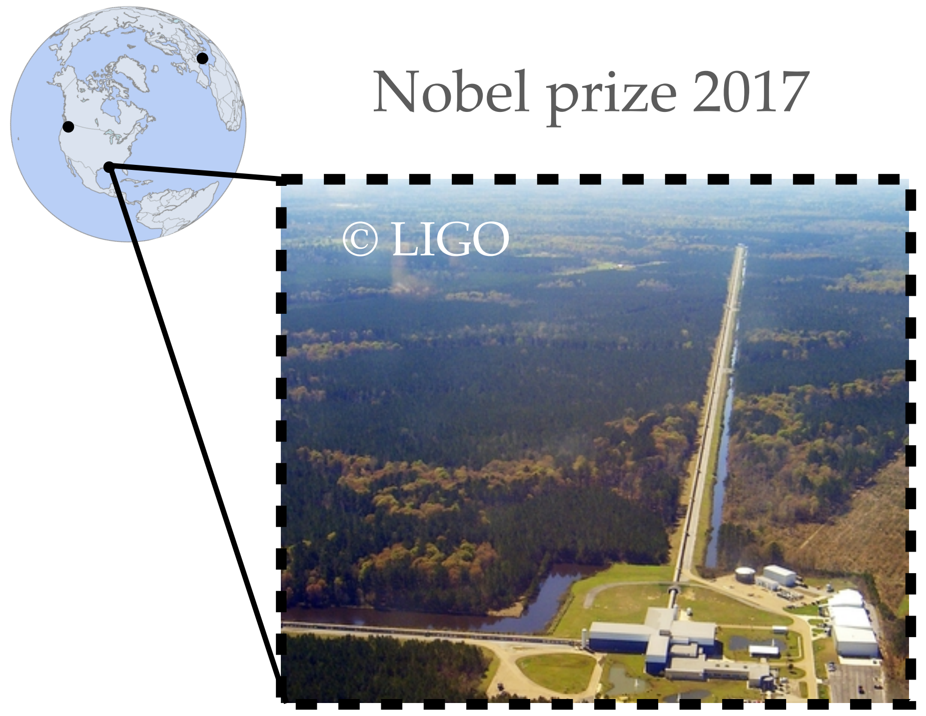
# Gravitational wave analysis: comparing data to models



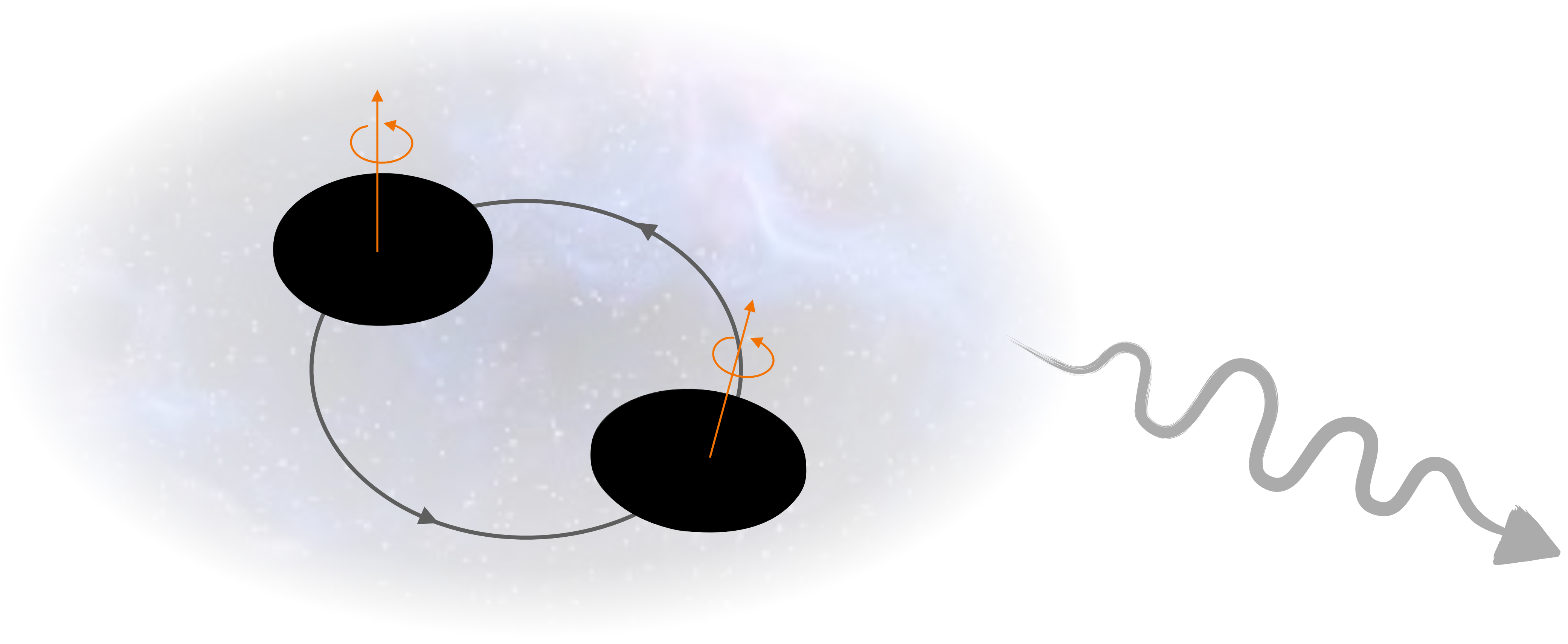
GW Measurement



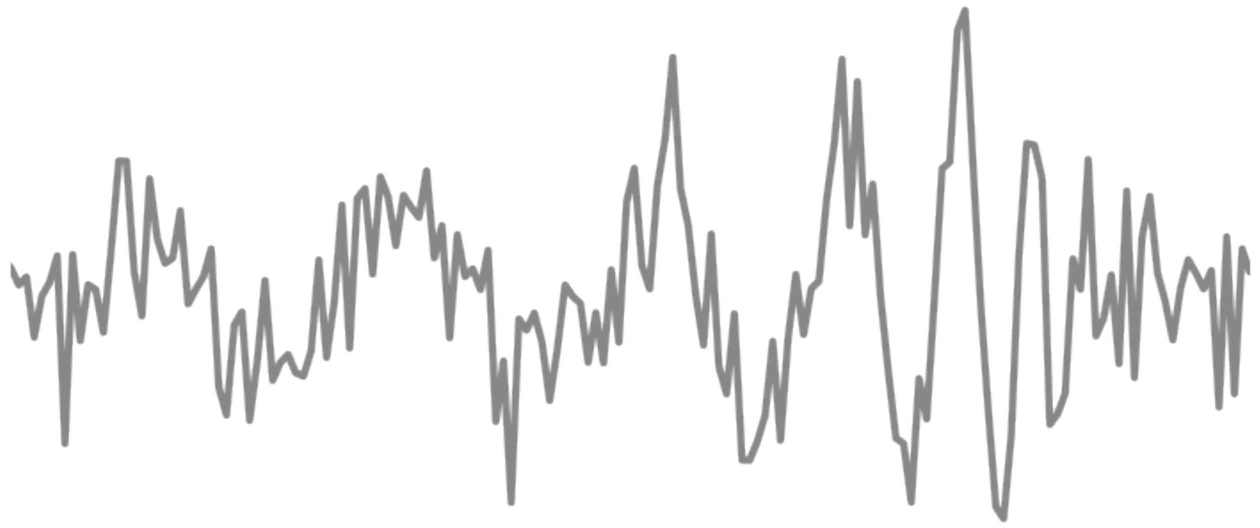
Nobel prize 2017



# Gravitational wave analysis: comparing data to models



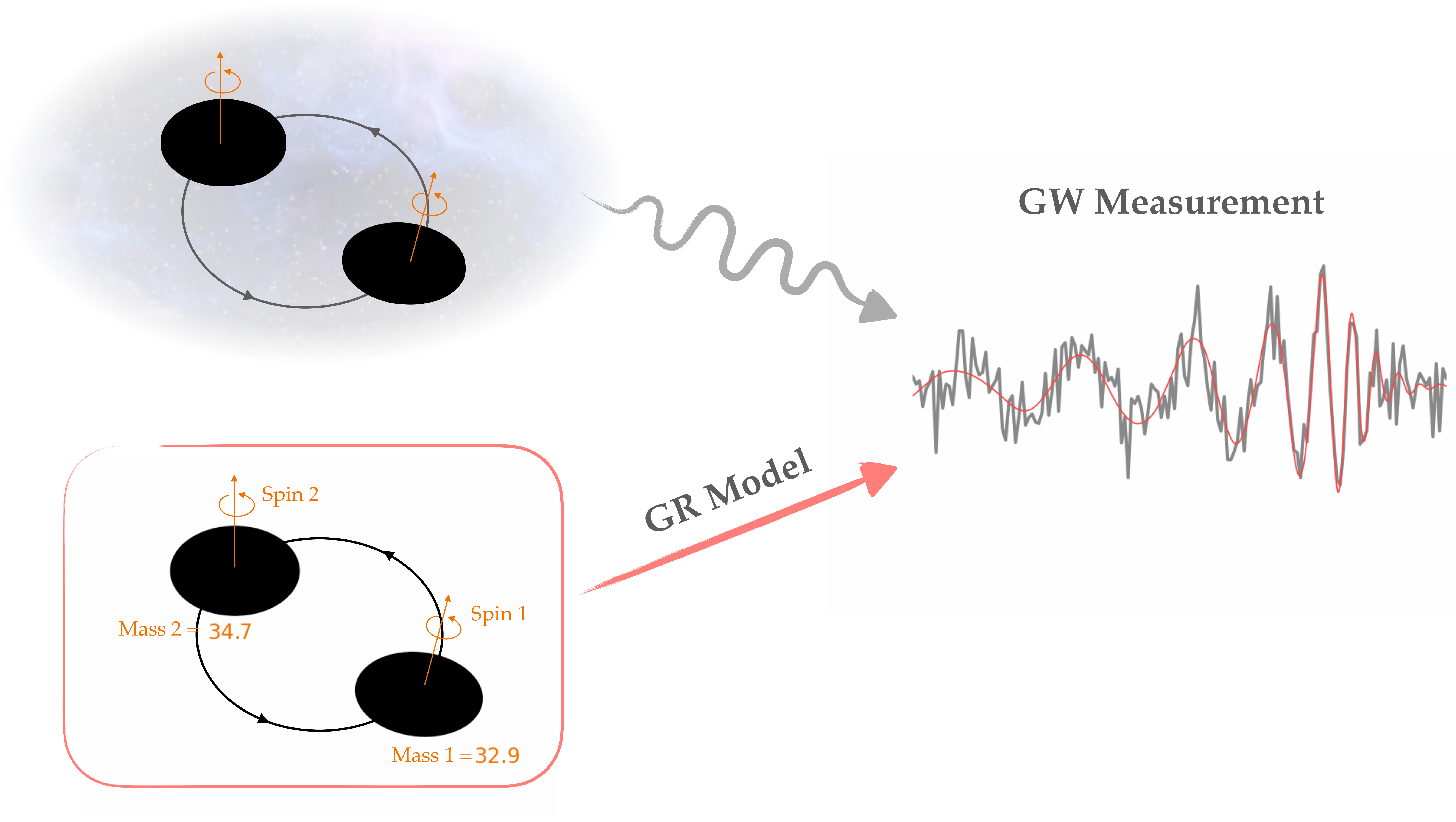
**GW Measurement**



**General relativity (GR)**

- Black hole mergers emit gravitational waves (GWs)
- GW shape depends on the black hole properties
  15 parameters: masses, spins, …

# Gravitational wave analysis: comparing data to models




**General relativity (GR)**

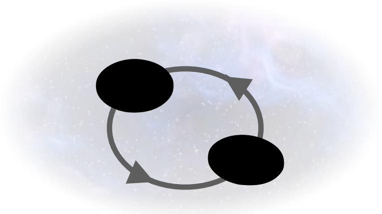
- Black hole mergers emit gravitational waves (GWs)
- GW shape depends on the black hole properties
  15 parameters: masses, spins, …

**GW analysis**

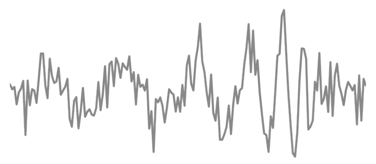
**Decode GW information to characterize the black holes**

# Inverse problems in science

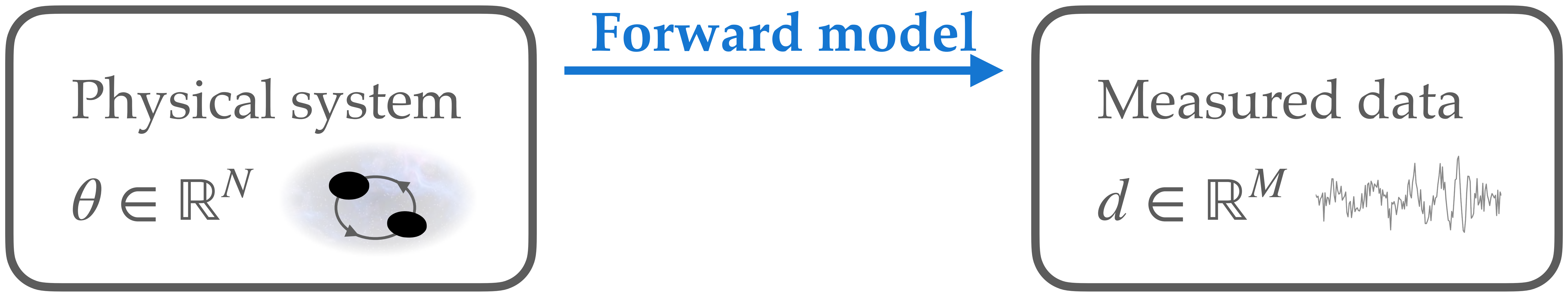
Physical system

$\theta \in \mathbb{R}^N$ 

Measured data

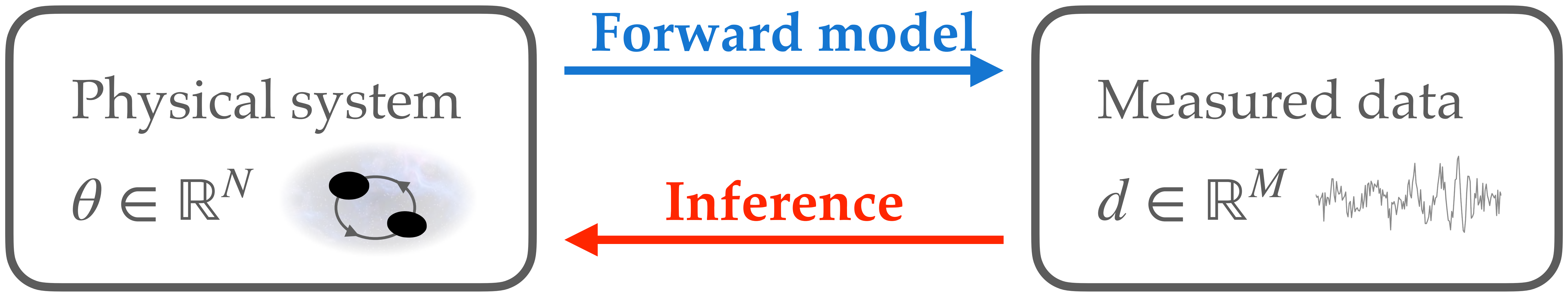
$d \in \mathbb{R}^M$ 

# Inverse problems in science




- **Forward direction $\theta \to d$** is defined by a simulator, $d \sim p(d\,|\,\theta)$

# Inverse problems in science




- **Forward direction** $\theta \rightarrow d$ is defined by a simulator, $d \sim p(d\,|\,\theta)$

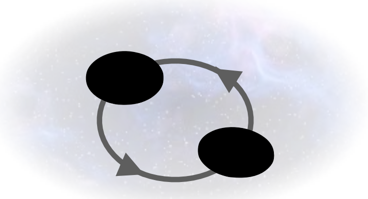
- **Inverse direction** with Bayesian inference

forward model      prior belief

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

# Inverse problems in science

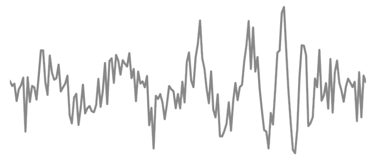
Physical system

$\theta \in \mathbb{R}^N$ 

**Forward model** →

← **Inference**

Measured data

$d \in \mathbb{R}^M$ 

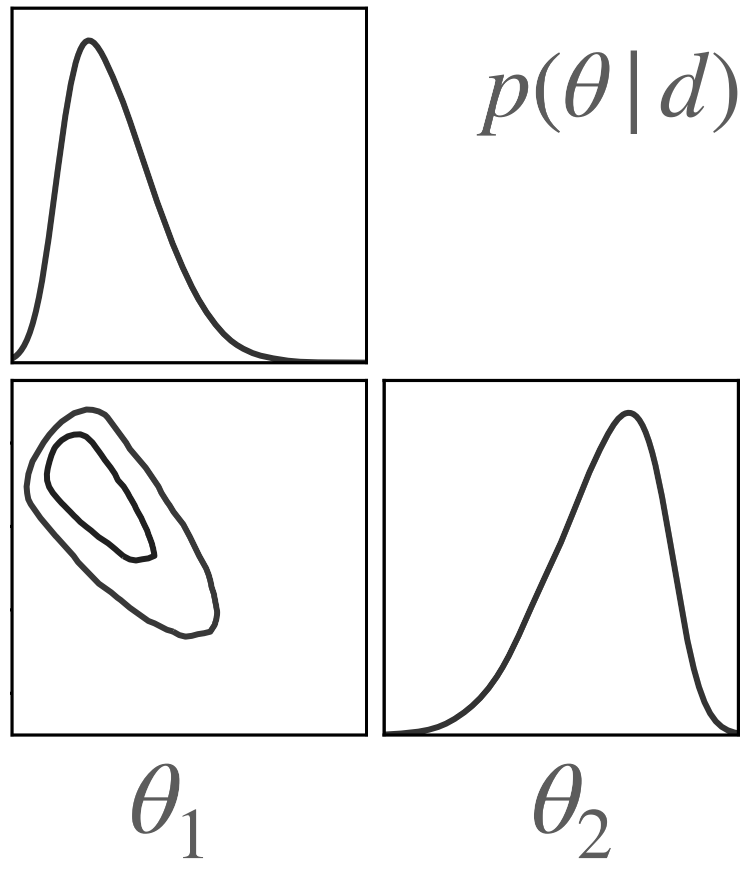
- **Forward direction $\theta \rightarrow d$** is defined by a simulator, $d \sim p(d \mid \theta)$

- **Inverse direction** with Bayesian inference

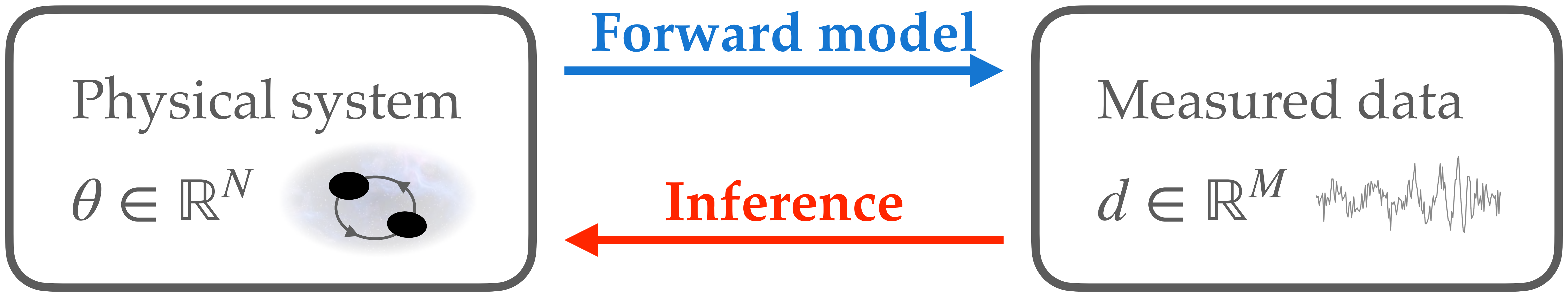
forward model ↘ ↙ prior belief

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

$p(\theta \mid d)$ **Bayesian posterior**



# Inverse problems in science

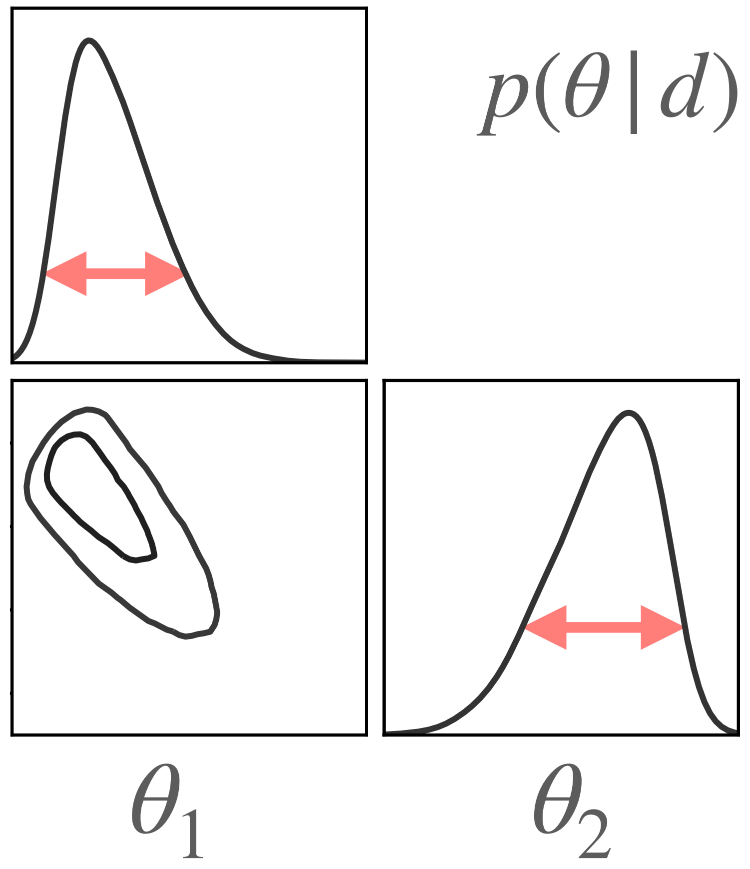



- **Forward direction** $\theta \to d$ is defined by a simulator, $d \sim p(d \mid \theta)$

- **Inverse direction** with Bayesian inference

forward model prior belief

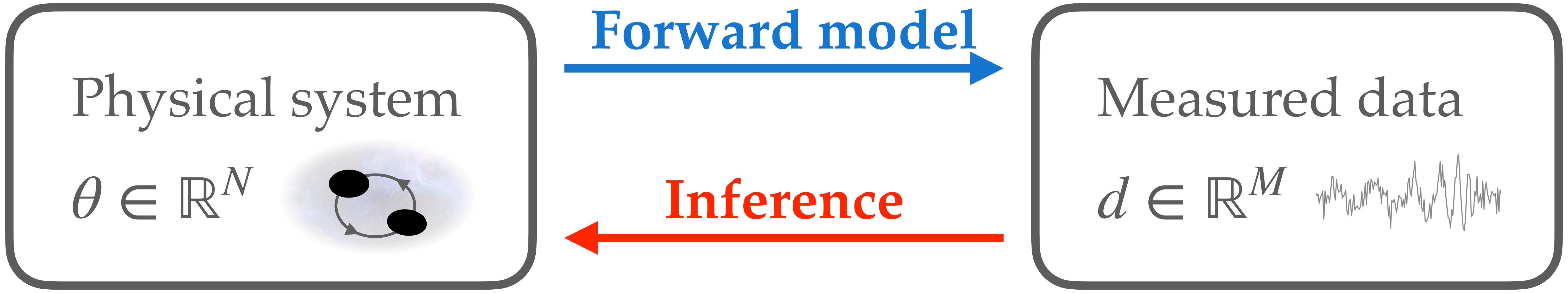
$$p(\theta \mid d) = \frac{p(d \mid \theta)\, p(\theta)}{p(d)}$$




$p(\theta \mid d)$ **Bayesian posterior**

$\rightarrow$ uncertainties

# Inverse problems in science

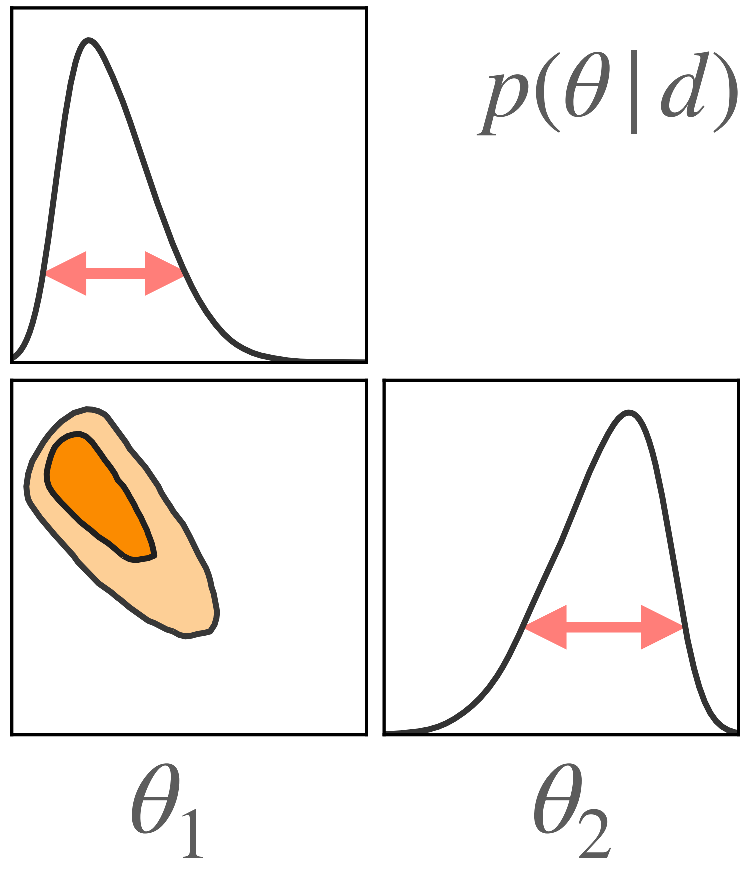



- **Forward direction $\theta \rightarrow d$** is defined by a simulator, $d \sim p(d\,|\,\theta)$

- **Inverse direction** with Bayesian inference

forward model      prior belief

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




$p(\theta\,|\,d)$ **Bayesian posterior**

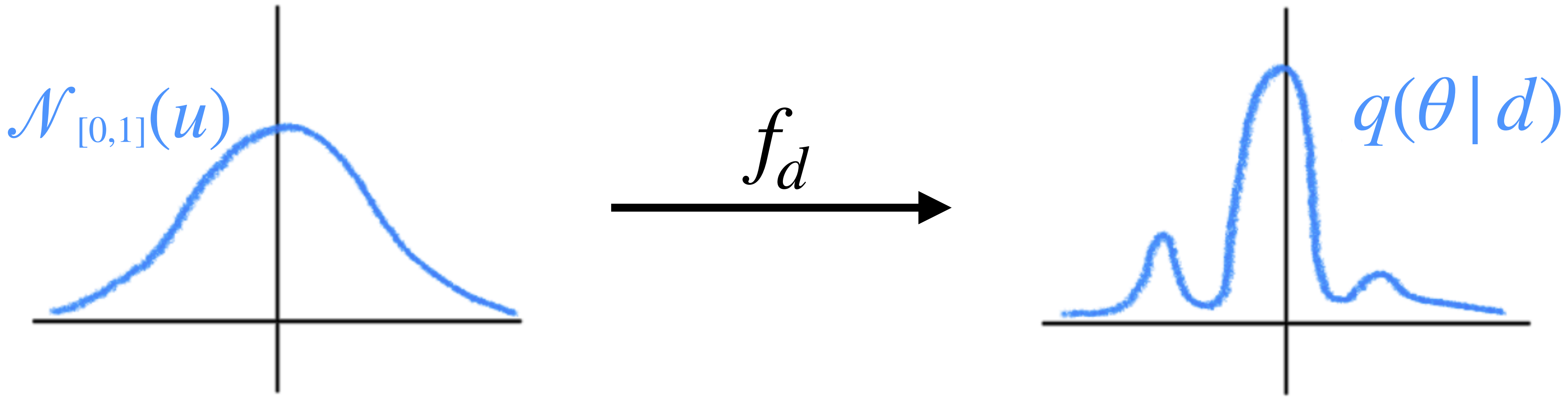
→ uncertainties

→ correlations

Rezende,Mohamed (ICML 2015)
Papamakarios,Murray (NeurIPS 2016)
Cranmer+ (PNAS 2020)

# Neural Posterior Estimation (NPE)

1) **Parameterize** posterior with normalizing flows $q(\theta|d)$



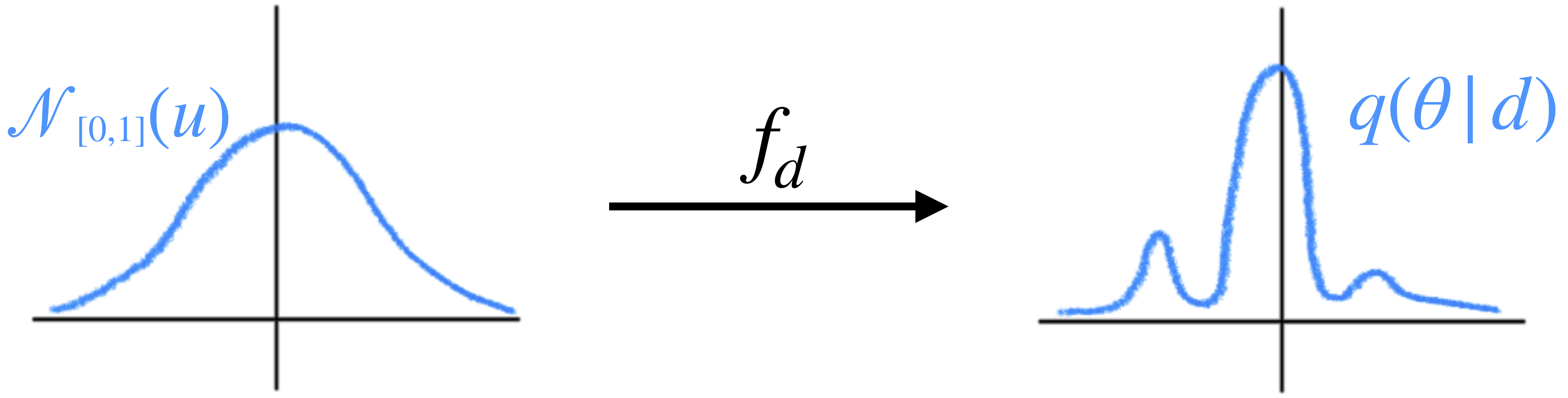

$$q(\theta|d) = \mathcal{N}_{[0,1]}\left(f_d^{-1}(\theta)\right)\left|\det J_{f_d}^{-1}\right|$$

- $f_d$ parameterized with neural net
- Arbitrarily **expressive**
- **Sampling & density** evaluation

Rezende,Mohamed (ICML 2015)
Papamakarios,Murray (NeurIPS 2016)
Cranmer+ (PNAS 2020)

# Neural Posterior Estimation (NPE)

1) **Parameterize** posterior with normalizing flows $q(\theta | d)$




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

- $f_d$ parameterized with neural net
- Arbitrarily **expressive**
- **Sampling & density** evaluation

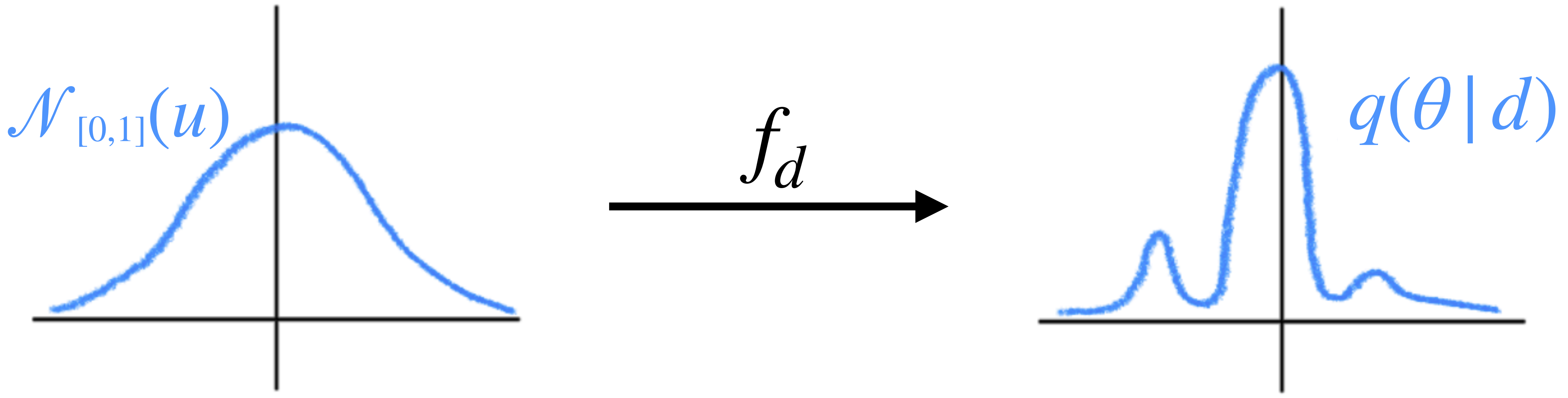
2) **Train flow** s.t. $q(\theta | d) \approx p(\theta | d)$

$$D_{\mathrm{KL}}\left(p(\theta | d) \,|\, q(\theta | d)\right) = -\,\mathbb{E}_{\theta \sim p(\theta)} \mathbb{E}_{d \sim p(d|\theta)} \left[\log q(\theta | d)\right] \quad + \text{const.}$$

Rezende,Mohamed (ICML 2015)
Papamakarios,Murray (NeurIPS 2016)
Cranmer+ (PNAS 2020)

# Neural Posterior Estimation (NPE)

1) **Parameterize** posterior with normalizing flows $q(\theta|d)$




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

- $f_d$ parameterized with neural net
- Arbitrarily **expressive**
- **Sampling & density** evaluation

2) **Train flow** s.t. $q(\theta|d) \approx p(\theta|d)$

$$D_{\mathrm{KL}}\left(p(\theta|d)\,|\,q(\theta|d)\right) = -\mathbb{E}_{\theta\sim p(\theta)}\mathbb{E}_{d\sim p(d|\theta)}\left[\log q(\theta|d)\right] \quad + \text{const.}$$

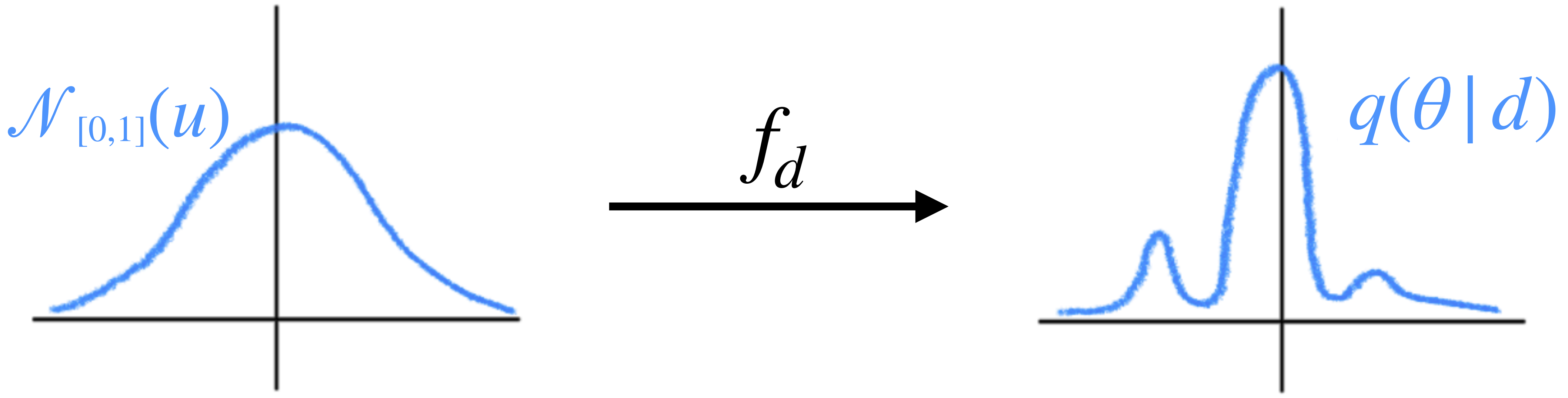
$$L = -\log q(\theta|d), \quad \theta \sim p(\theta),\ d \sim p(d|\theta)$$

- NPE **minimizes KL** divergence
- Converges to $q(\theta|d) = p(\theta|d)$
- Training based only on simulations

Rezende,Mohamed (ICML 2015)
Papamakarios,Murray (NeurIPS 2016)
Cranmer+ (PNAS 2020)

# Neural Posterior Estimation (NPE)

1) **Parameterize** posterior with normalizing flows $q(\theta|d)$




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

- $f_d$ parameterized with neural net
- Arbitrarily **expressive**
- **Sampling & density** evaluation

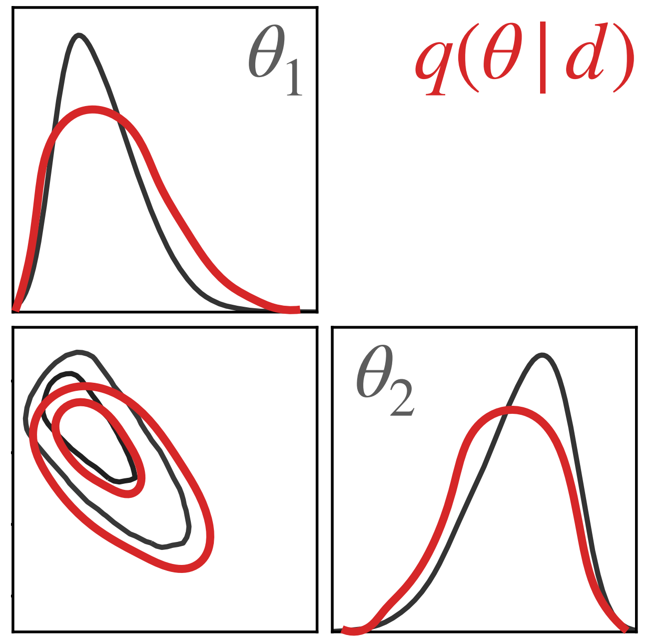
2) **Train flow** s.t. $q(\theta|d) \approx p(\theta|d)$

$$D_{\mathrm{KL}}\left(p(\theta|d)\,|\,q(\theta|d)\right) = -\mathbb{E}_{\theta\sim p(\theta)}\mathbb{E}_{d\sim p(d|\theta)}\left[\log q(\theta|d)\right] + \text{const.}$$

$$L = -\log q(\theta|d), \quad \theta \sim p(\theta),\ d \sim p(d|\theta)$$

- NPE **minimizes KL** divergence
- Converges to $q(\theta|d) = p(\theta|d)$
- Training based only on simulations

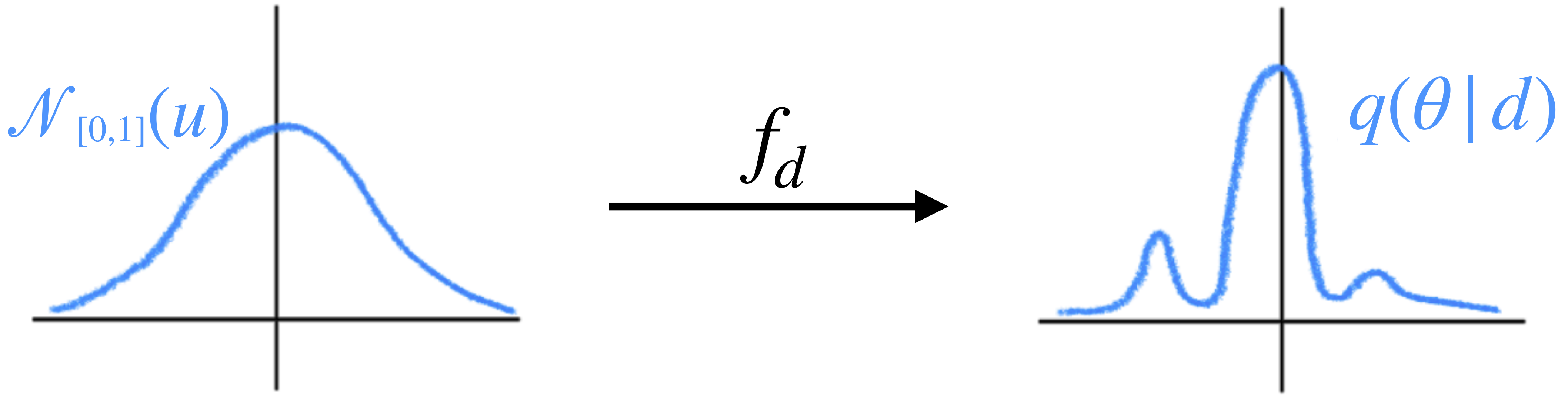
3) At inference, estimate $p(\theta|d)$ by **sampling** $\theta \sim q(\theta|d)$



# Neural Posterior Estimation (NPE)

Rezende,Mohamed (ICML 2015)
Papamakarios,Murray (NeurIPS 2016)
Cranmer+ (PNAS 2020)

1) **Parameterize** posterior with normalizing flows $q(\theta|d)$




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

- $f_d$ parameterized with neural net
- Arbitrarily **expressive**
- **Sampling & density** evaluation

2) **Train flow** s.t. $q(\theta|d) \approx p(\theta|d)$

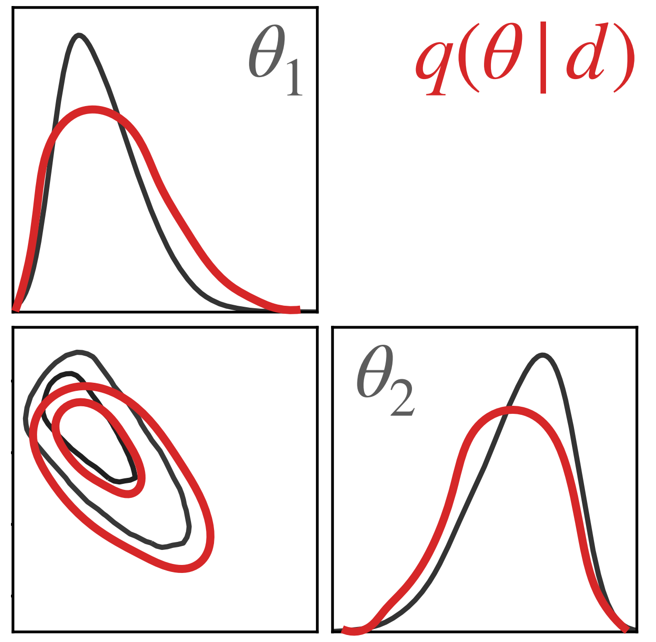
$$D_{\mathrm{KL}}\left(p(\theta|d)\,|\,q(\theta|d)\right) = -\,\mathbb{E}_{\theta\sim p(\theta)}\mathbb{E}_{d\sim p(d|\theta)}\left[\log q(\theta|d)\right] \quad + \text{const.}$$

$$L = -\log q(\theta|d), \quad \theta \sim p(\theta),\; d \sim p(d|\theta)$$

- NPE **minimizes KL** divergence
- Converges to $q(\theta|d) = p(\theta|d)$
- Training based only on simulations

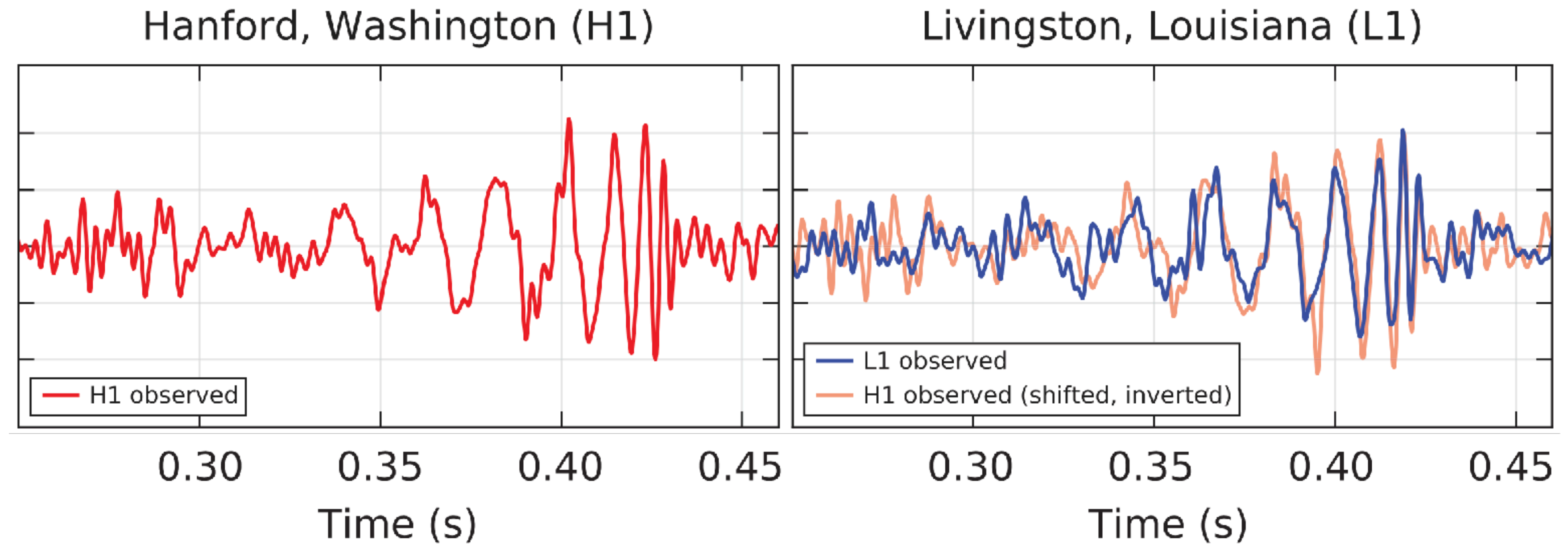
3) At inference, estimate $p(\theta|d)$ by **sampling** $\theta \sim q(\theta|d)$

***Amortized* inference**: $q(\theta|d)$ applicable to all data $d \sim p(d)$

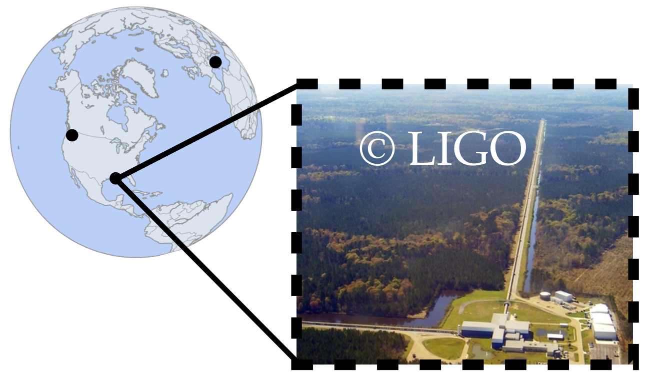


# NPE for binary black holes

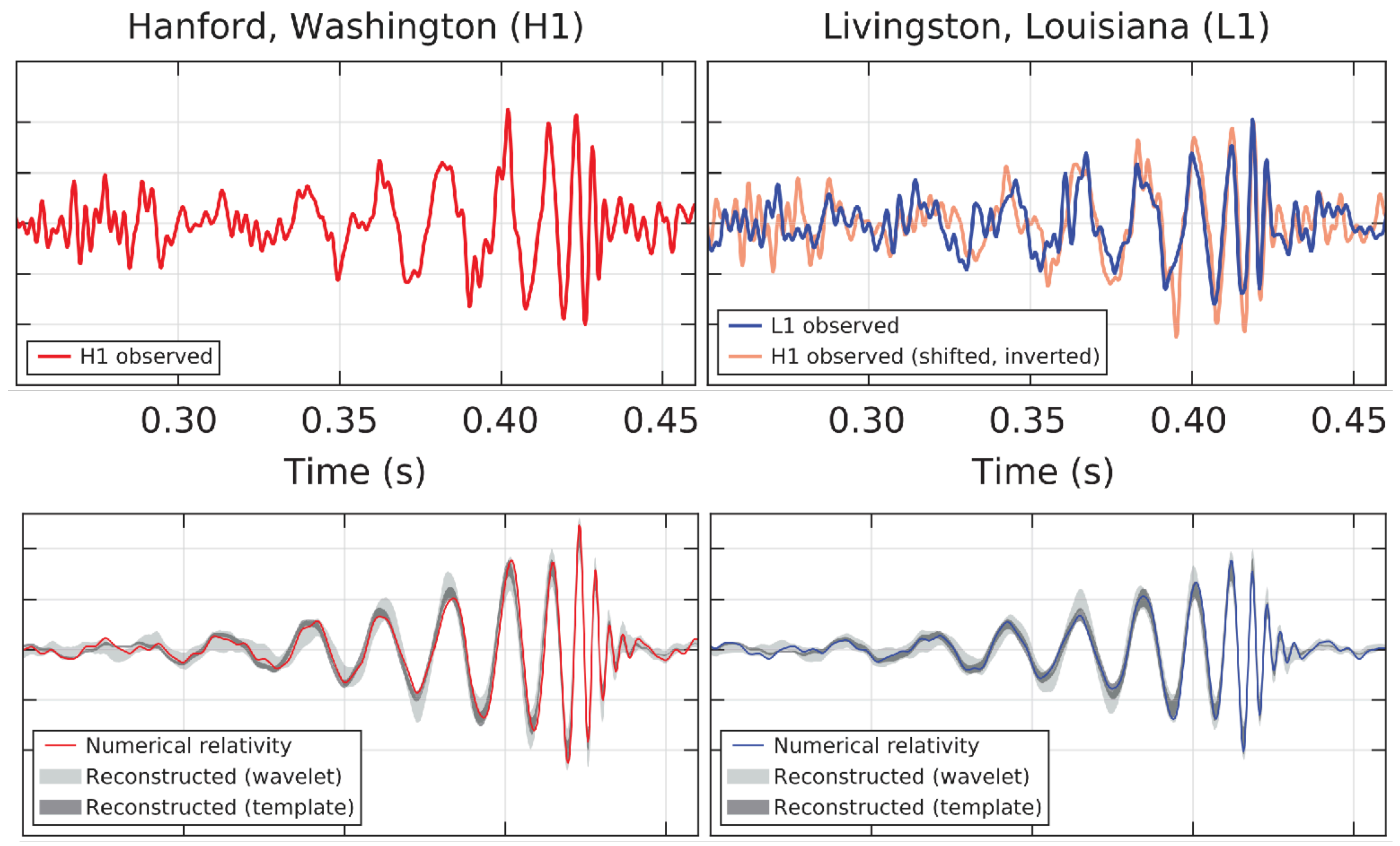
# GW forward model




observed data



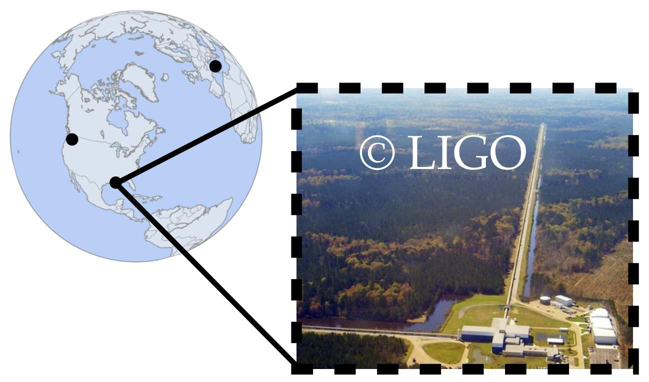
# GW forward model



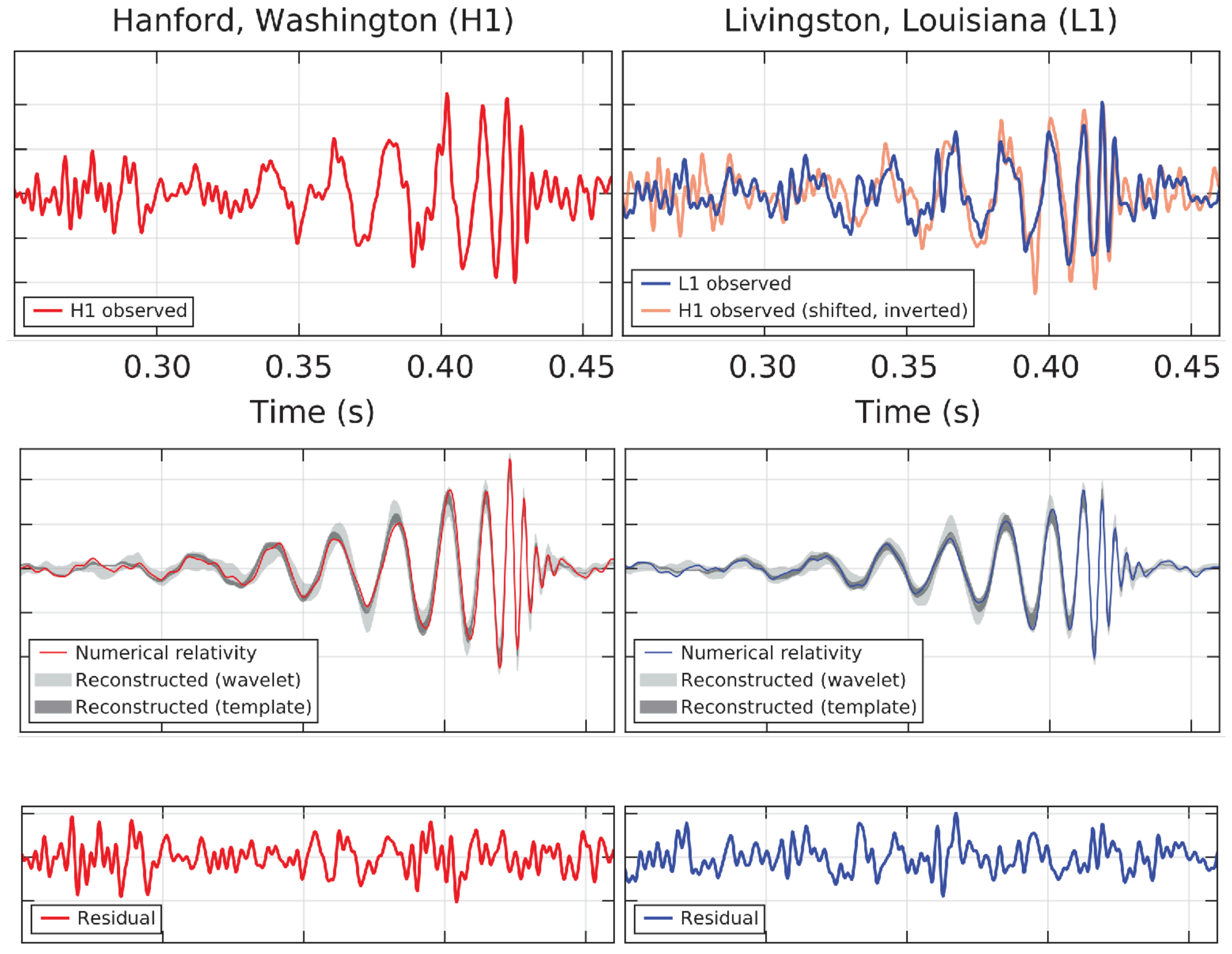

observed data

=

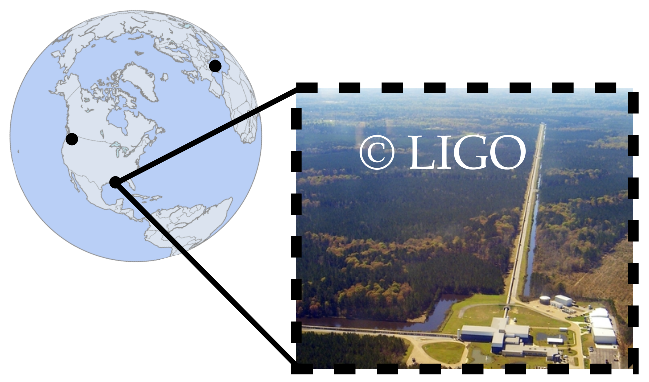
signal



# GW forward model




observed data

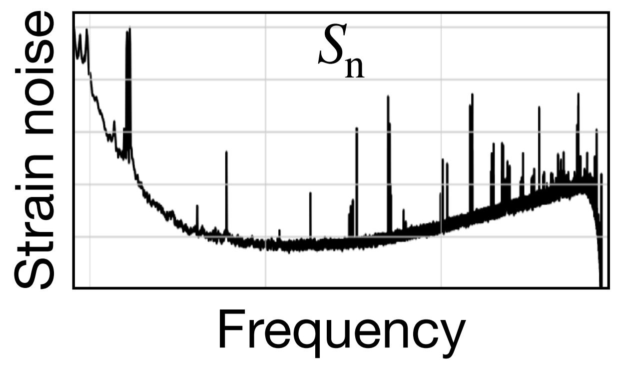



=

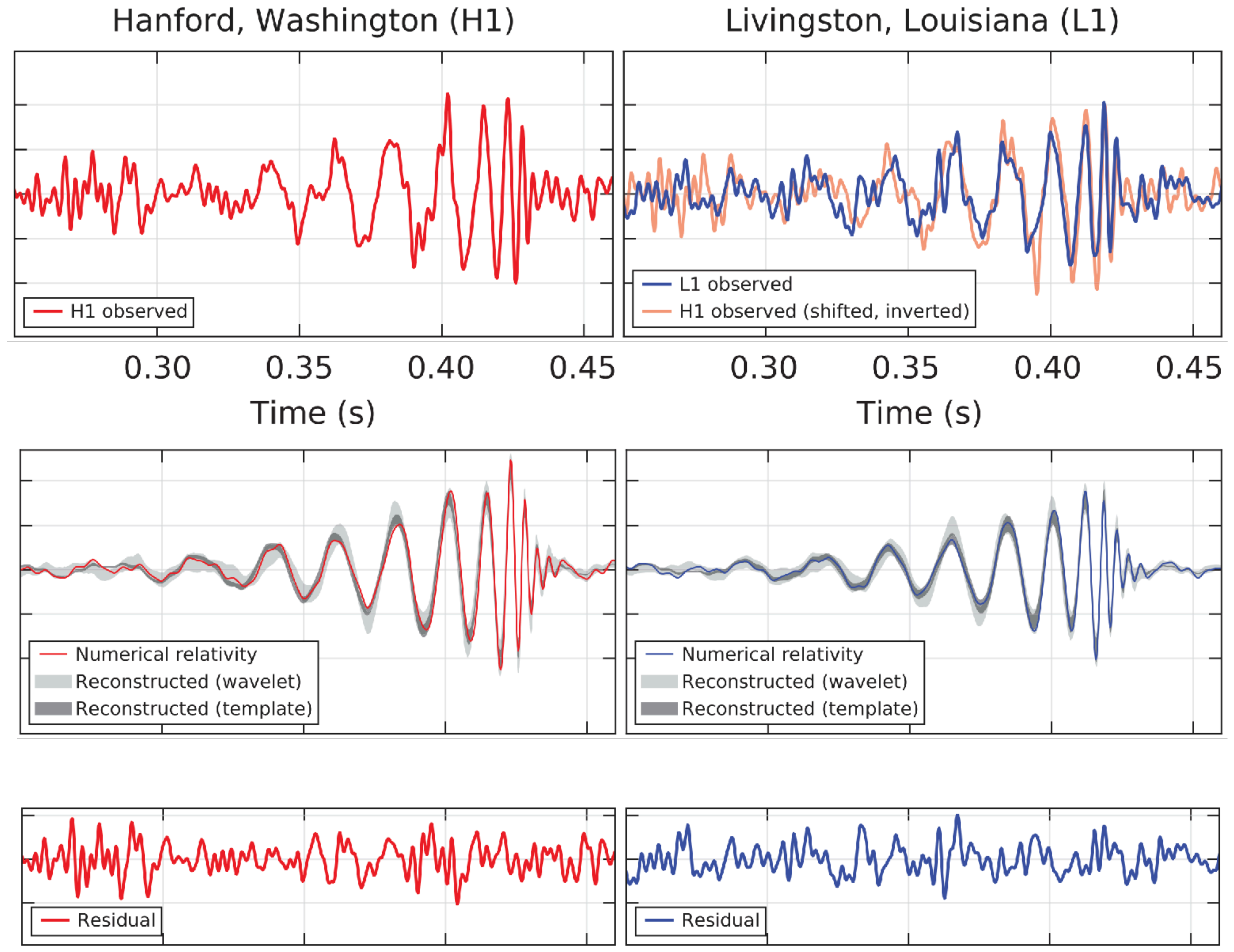
signal

+

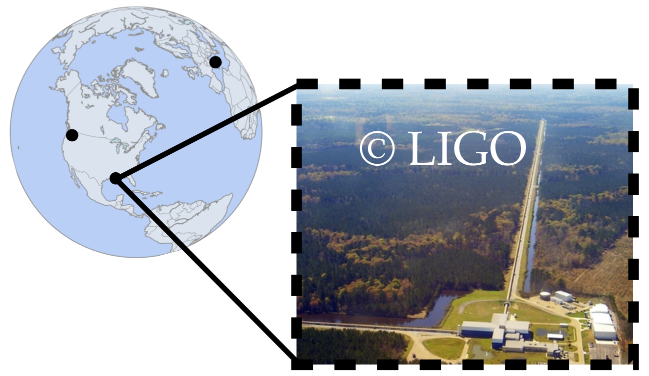
noise



# GW forward model




observed data




$d \sim p(d \mid \theta, S_\mathrm{n})$

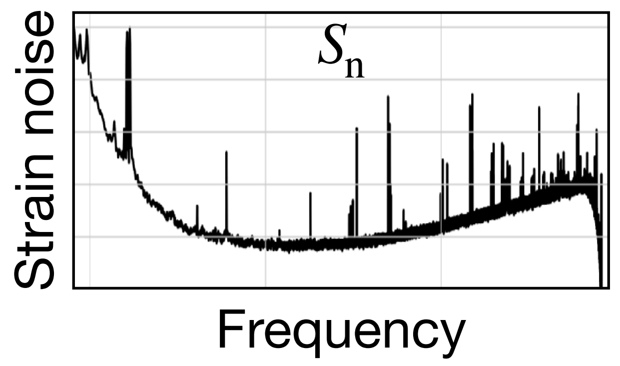
$=$

signal

$h(\theta)$

$+$

noise

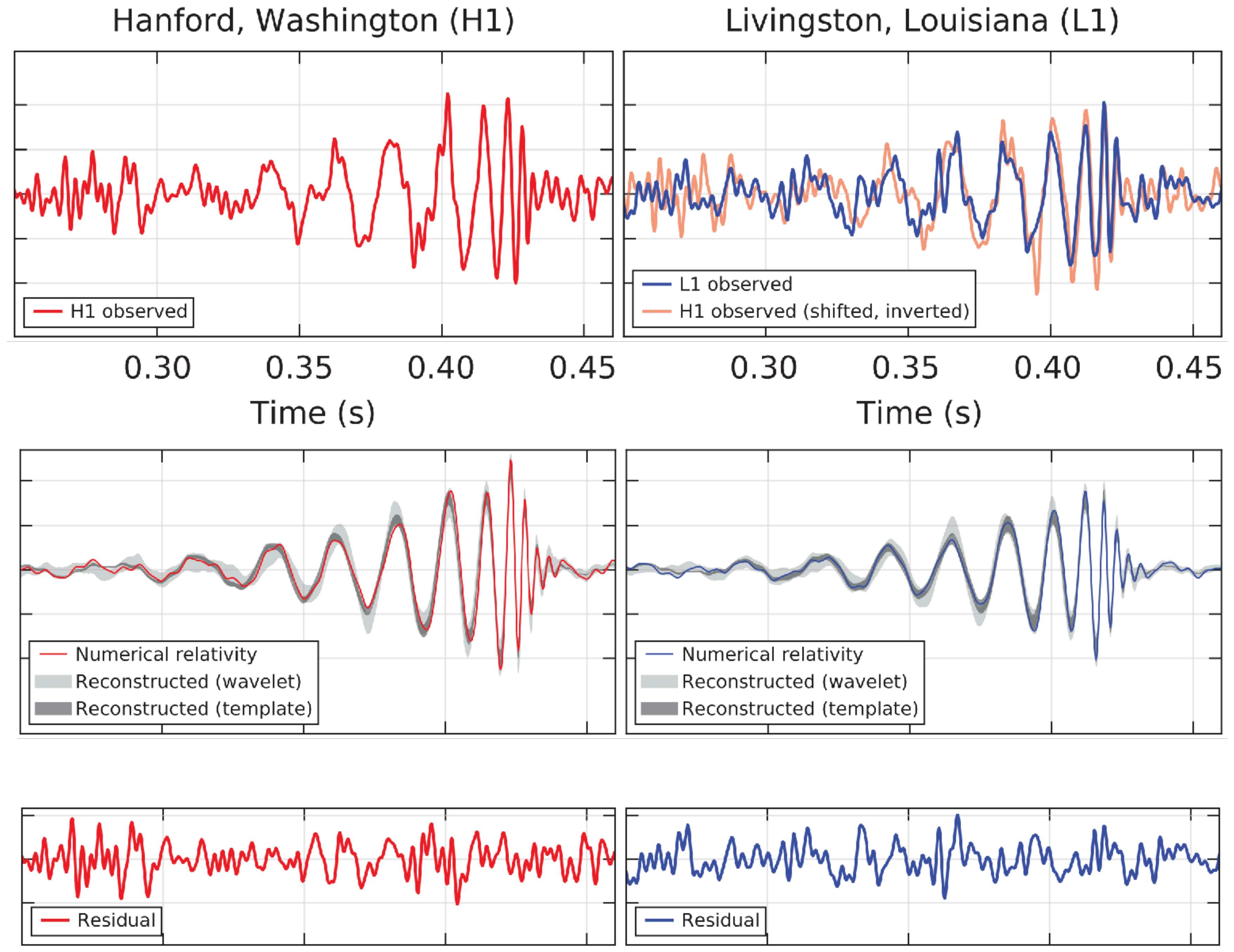



$n \sim N(0, S_\mathrm{n})$

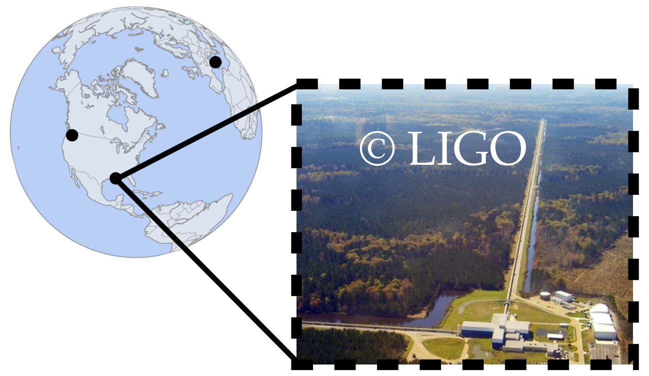
Parameters of GW model:

- Binary parameters $\theta \in \mathbb{R}^{15}$ → Parameters of interest
- Detector noise spectrum $S_\mathrm{n} \in \mathbb{R}^{k}$ → Detector property, from external data

# GW forward model




observed data




$d \sim p(d \mid \theta, S_\mathrm{n})$

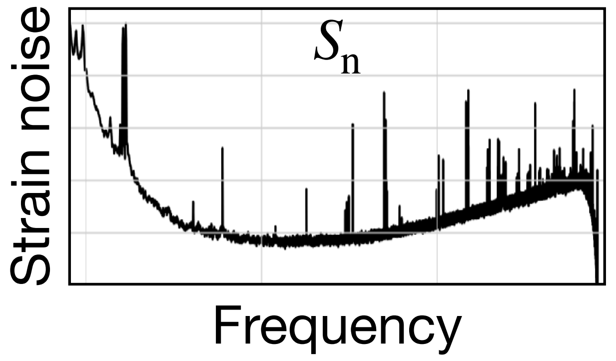
=

signal

$h(\theta)$

+

noise



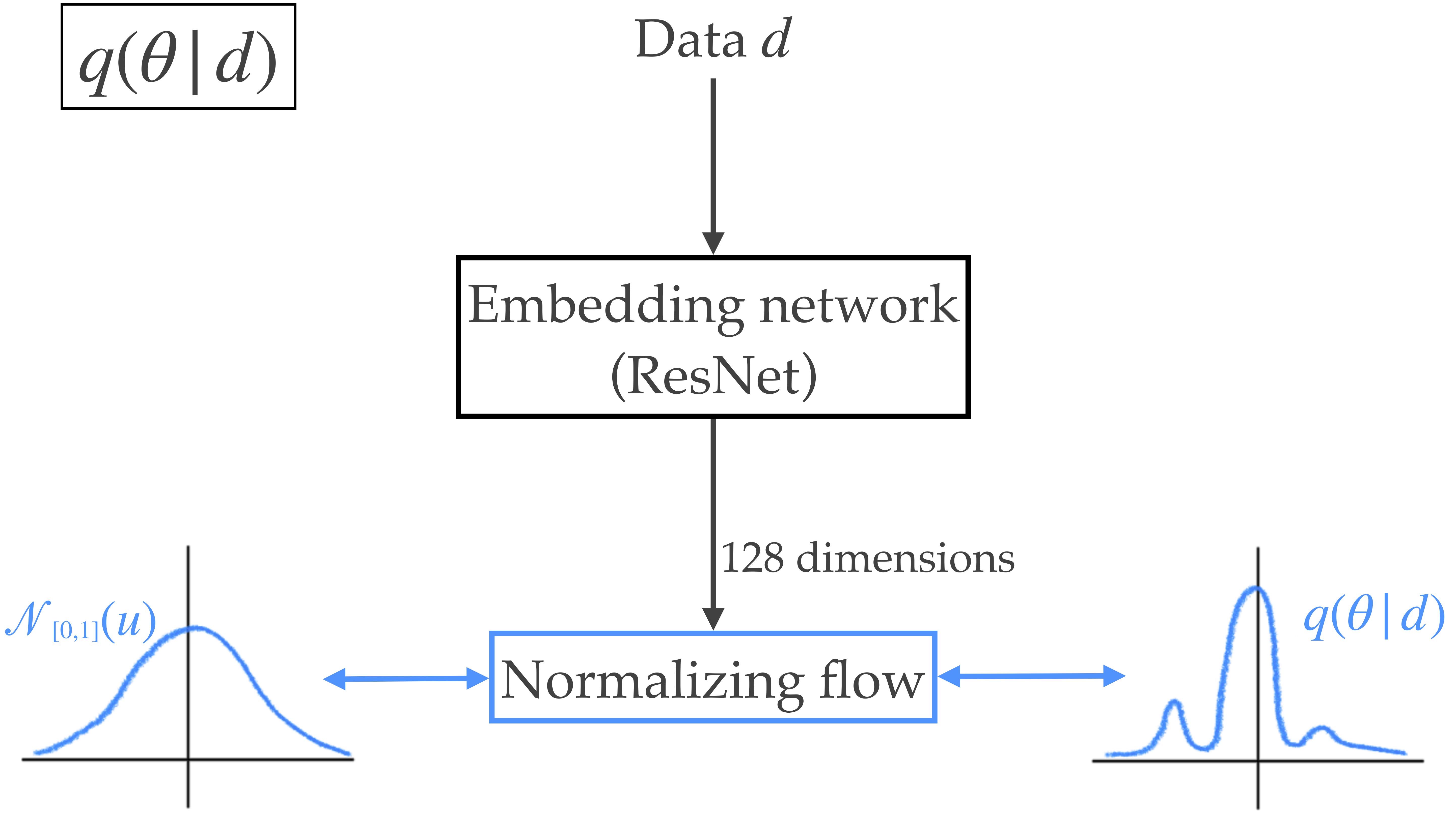

$n \sim N(0, S_\mathrm{n})$

Parameters of GW model:

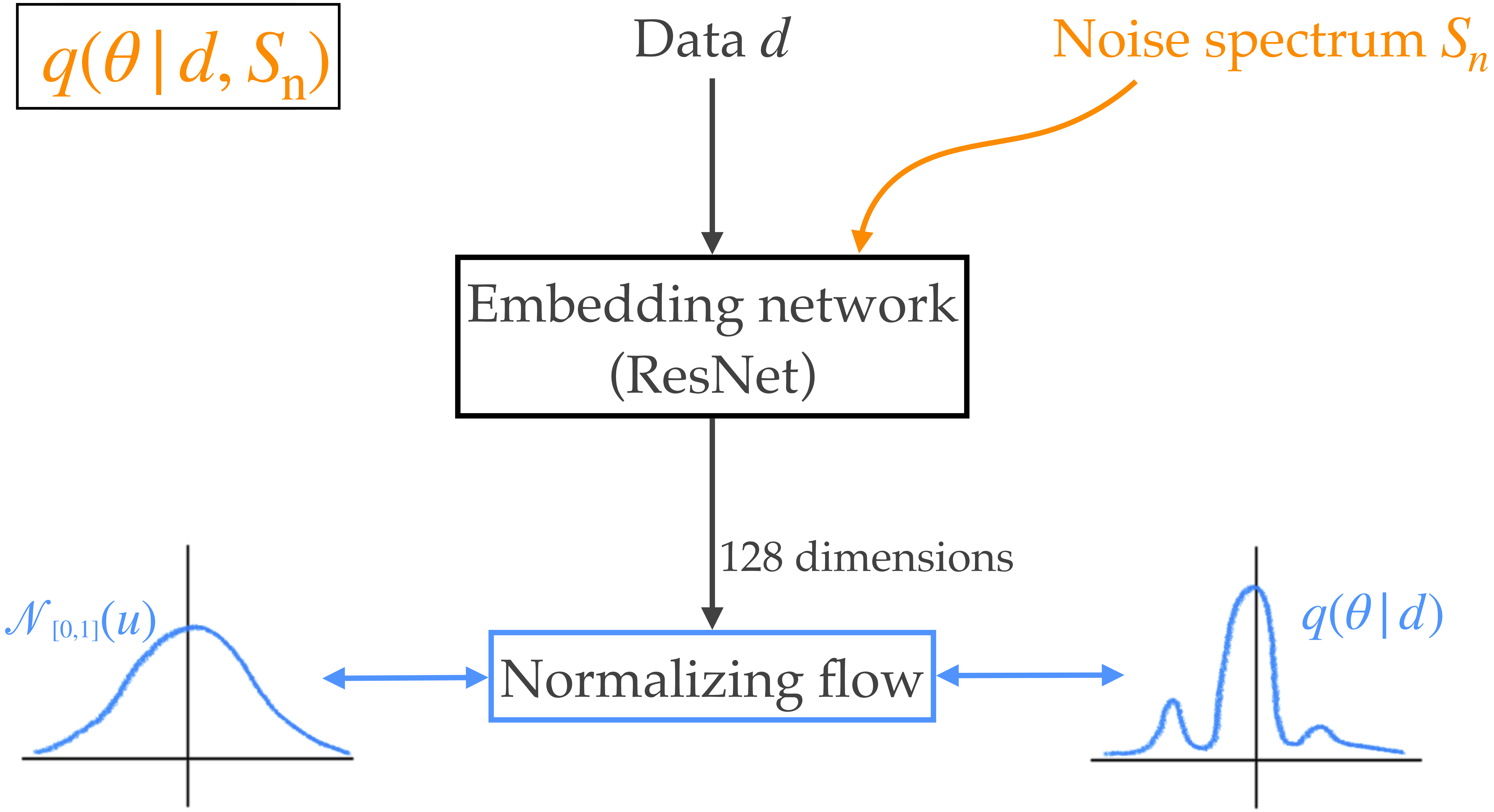
- Binary parameters $\theta \in \mathbb{R}^{15}$ → Parameters of interest
- Detector noise spectrum $S_\mathrm{n} \in \mathbb{R}^k$ → Detector property, from external data

$\Rightarrow p(\theta \mid d, S_\mathrm{n})$

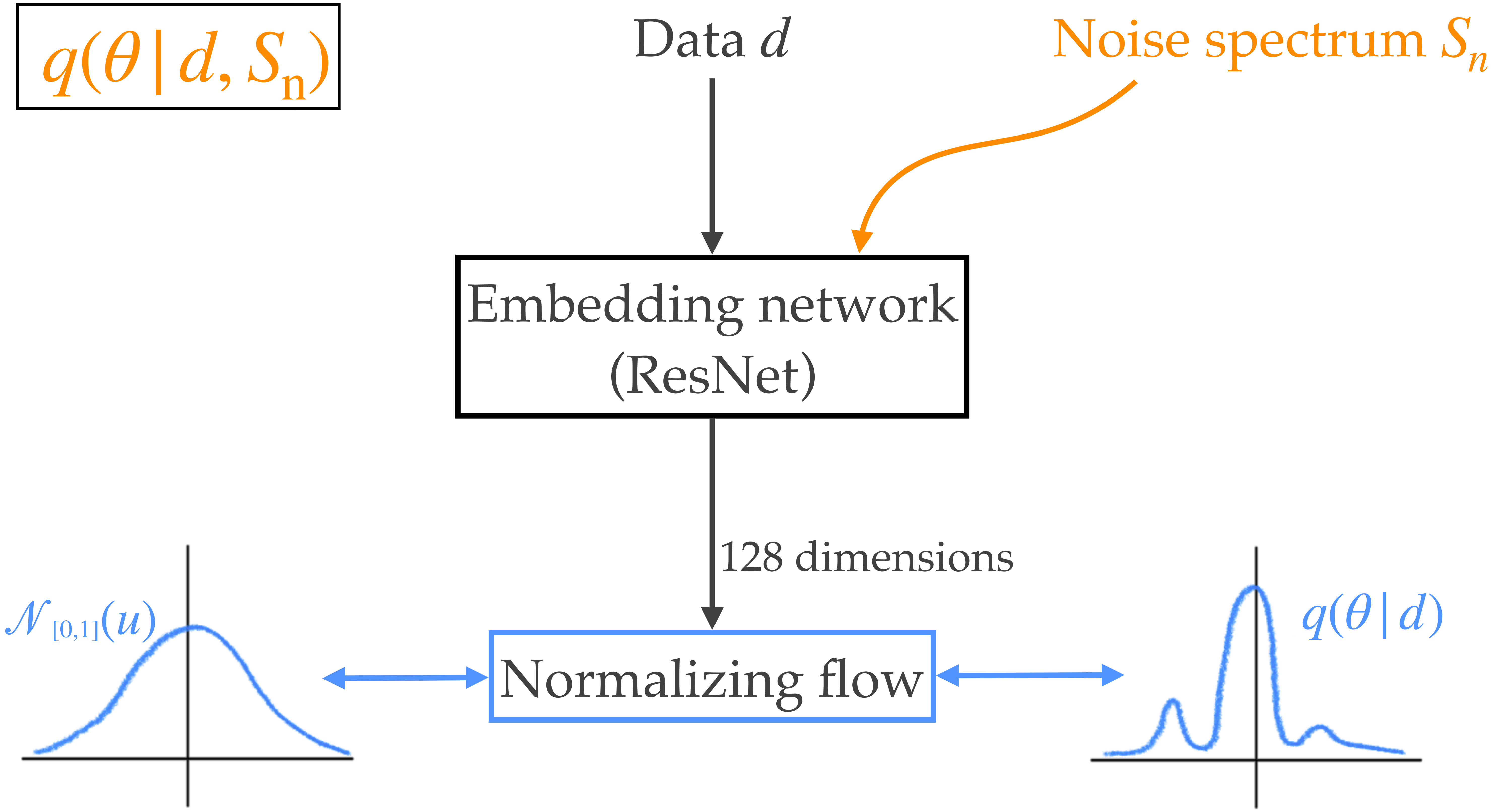
# NPE model


$q(\theta | d)$
Data $d$
Embedding network
(ResNet)
128 dimensions
$\mathcal{N}_{[0,1]}(u)$
Normalizing flow
$q(\theta | d)$

# NPE model


$q(\theta \mid d, S_n)$
Data $d$
Noise spectrum $S_n$
Embedding network
(ResNet)
128 dimensions
$\mathcal{N}_{[0,1]}(u)$
Normalizing flow
$q(\theta \mid d)$

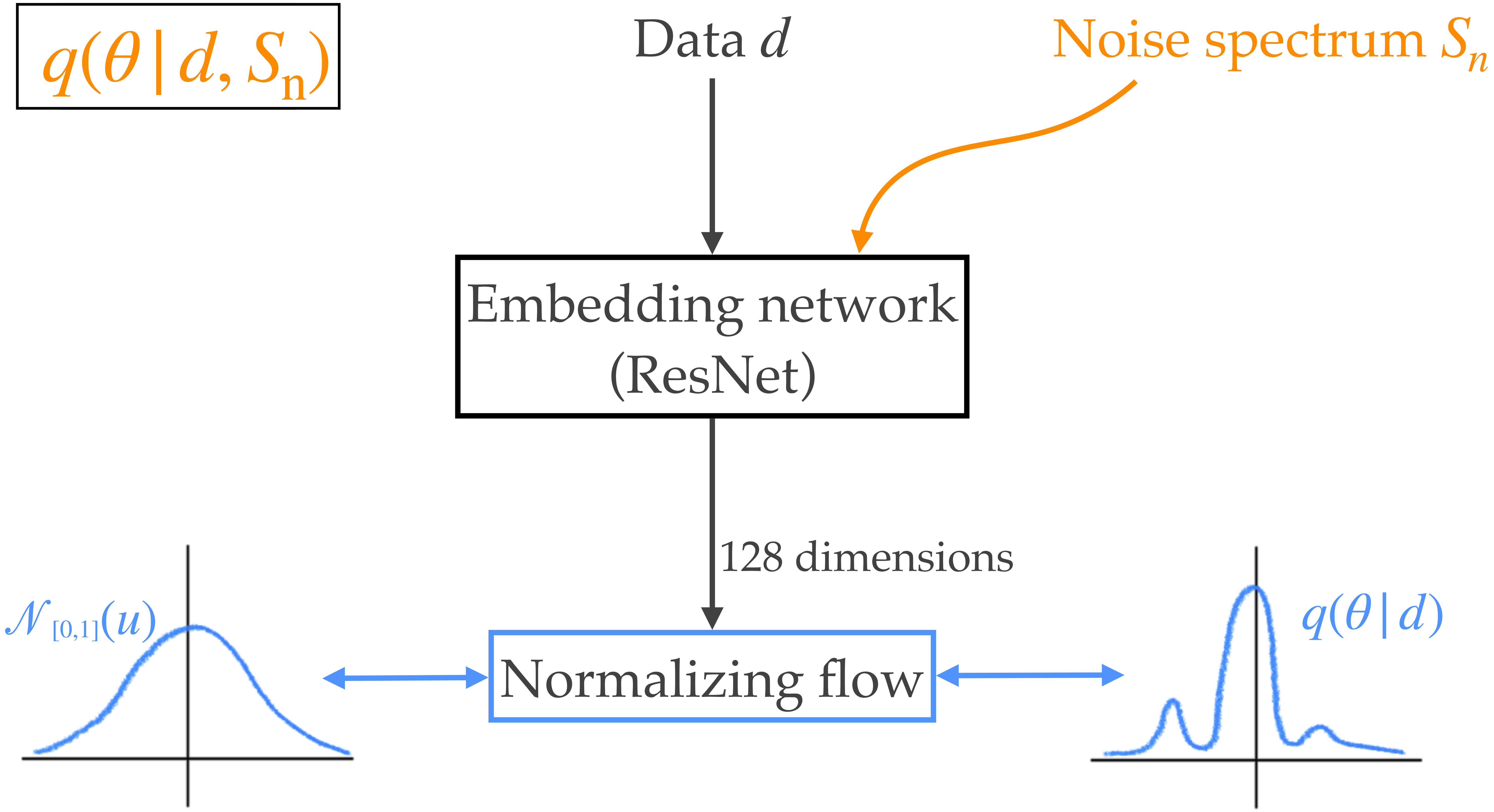
# NPE model




**Simulation-based training**

$$\theta \sim p(\theta),$$
$$S_n \sim p(S_n),$$
$$d \sim p(d\,|\,\theta, S_n)$$

# NPE model

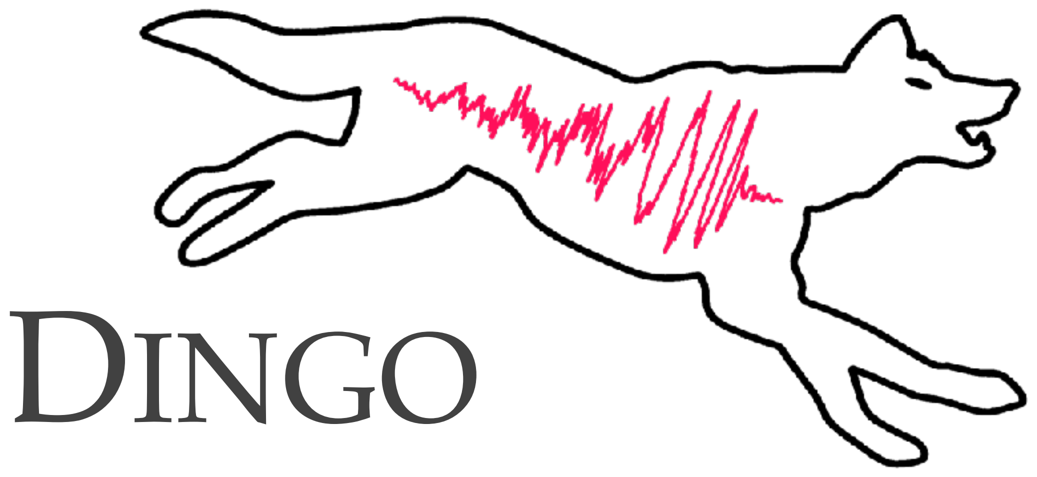



**Simulation-based training**

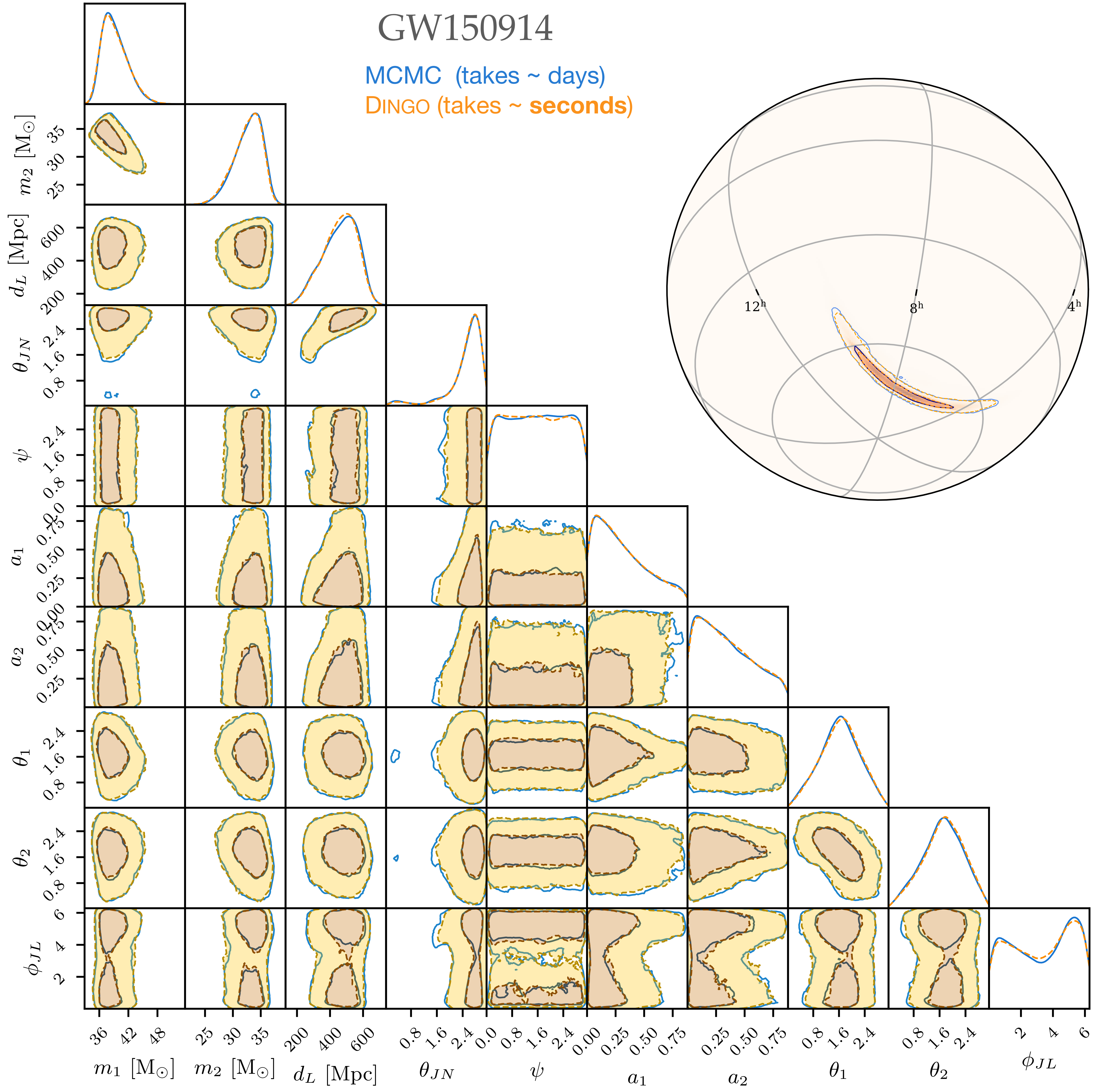
$$\theta \sim p(\theta),$$
$$S_\mathrm{n} \sim p(S_\mathrm{n}),$$
$$d \sim p(d \mid \theta, S_\mathrm{n})$$

Inference result independent of $p(S_\mathrm{n})$ due to conditioning on $S_\mathrm{n}$

# Binary black holes

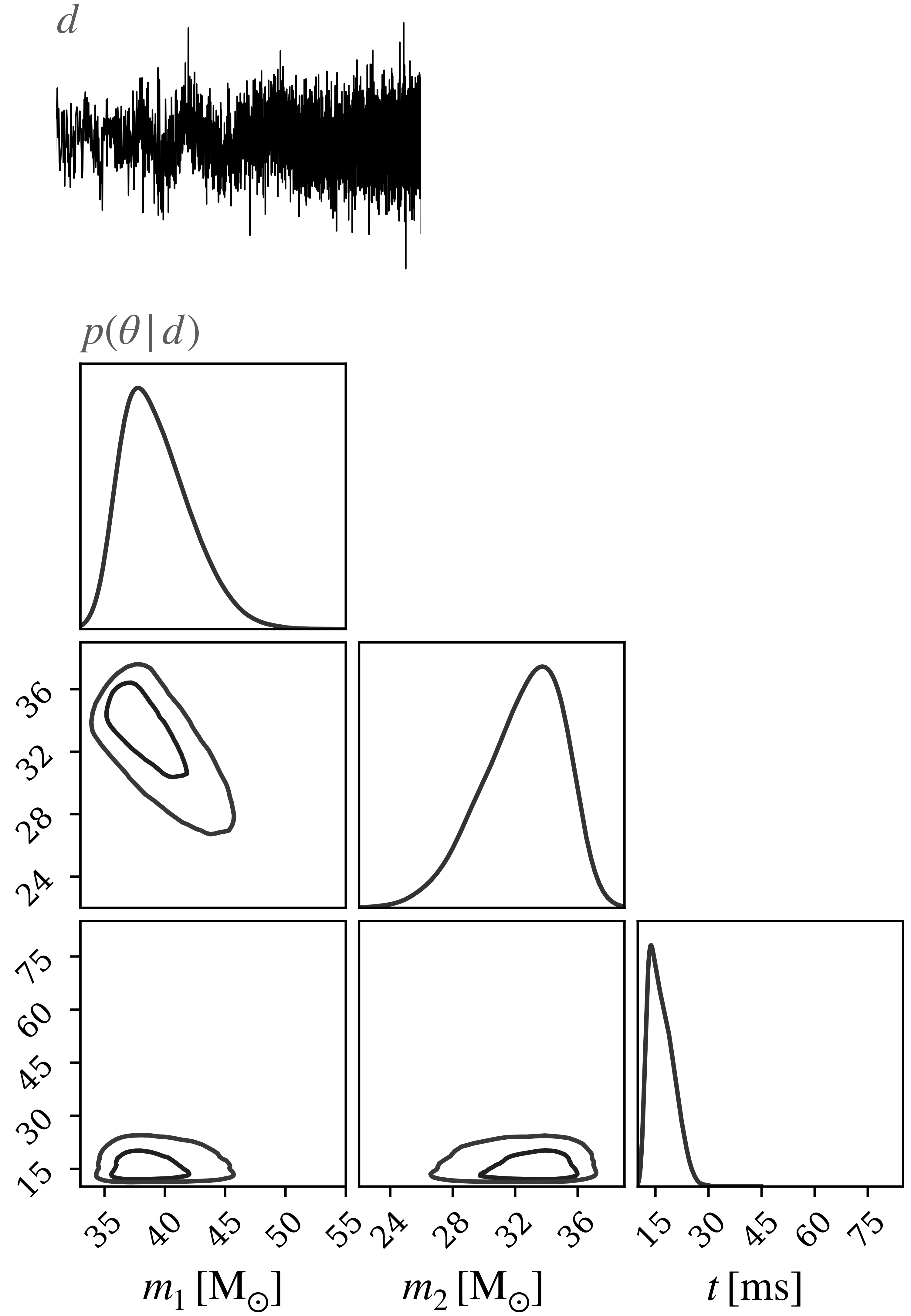



- Inference in seconds to minutes using pre-trained networks (1000x speed up)
- Extremely good agreement with standard samplers
- Likelihood-free



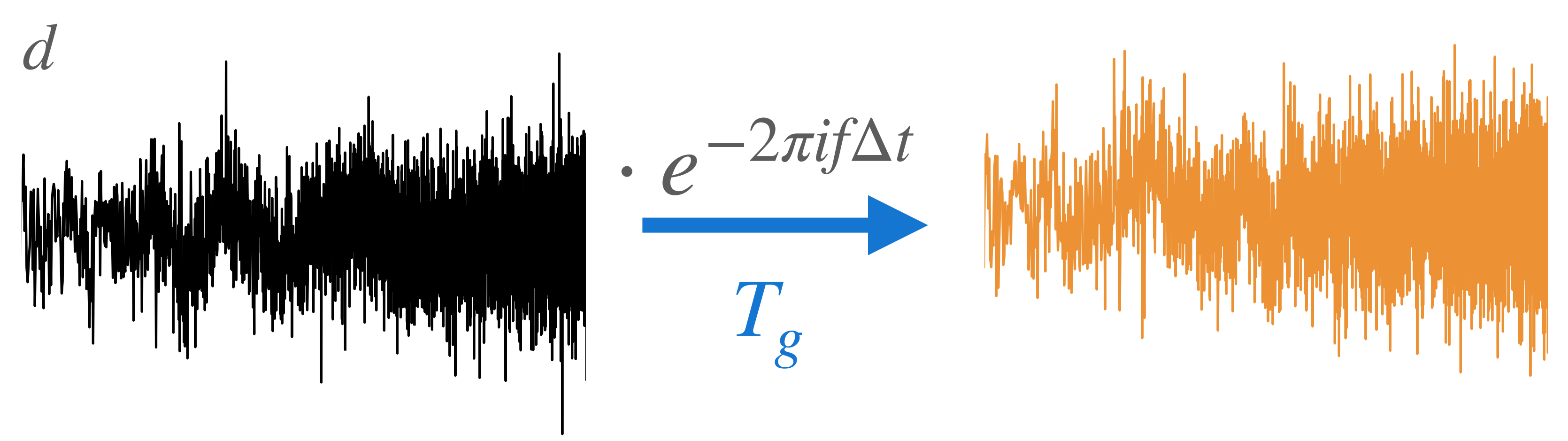
# NPE with symmetries: Group-equivariant NPE

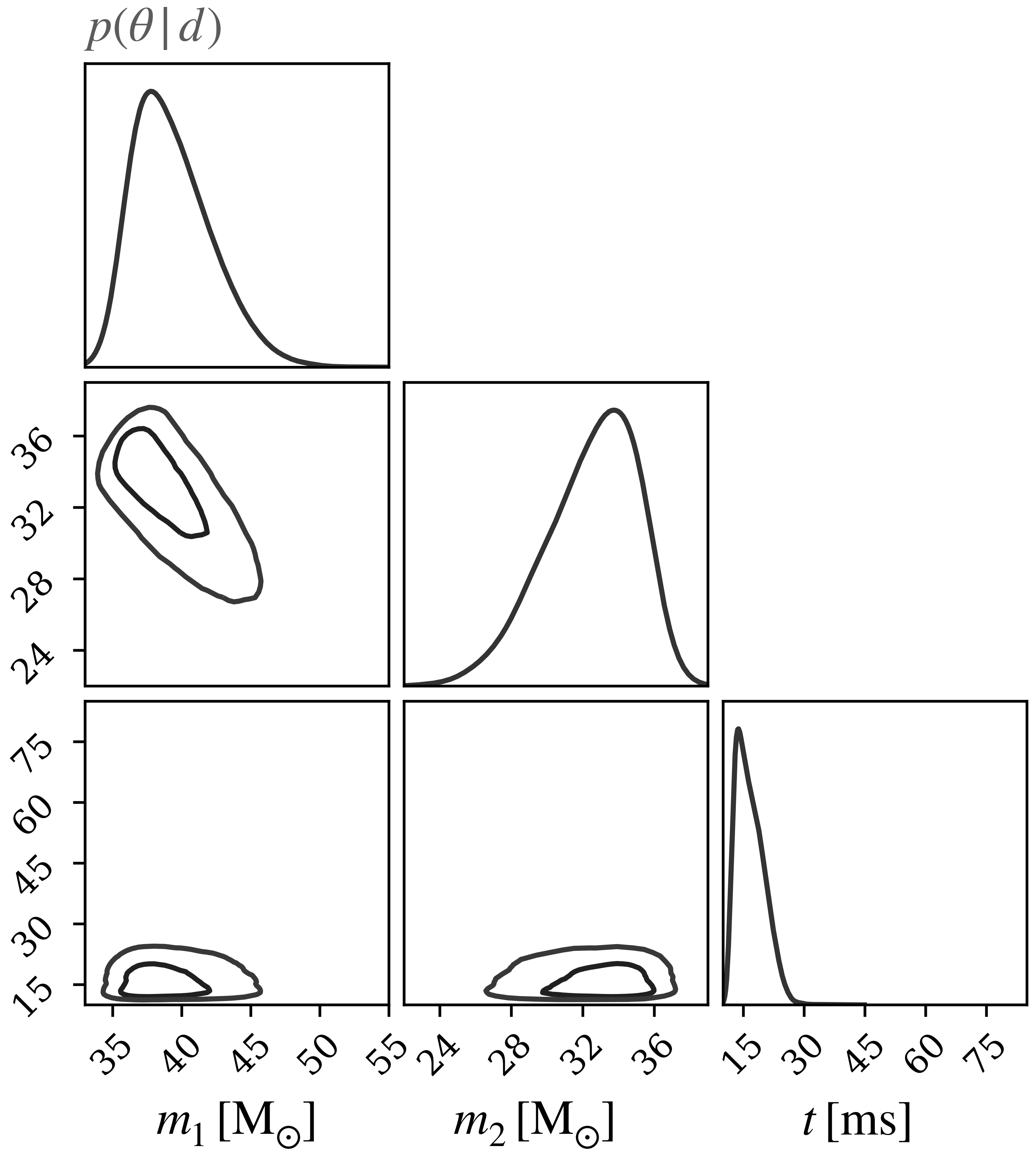
- Equivariance (covariance) under time shift



# NPE with symmetries: Group-equivariant NPE

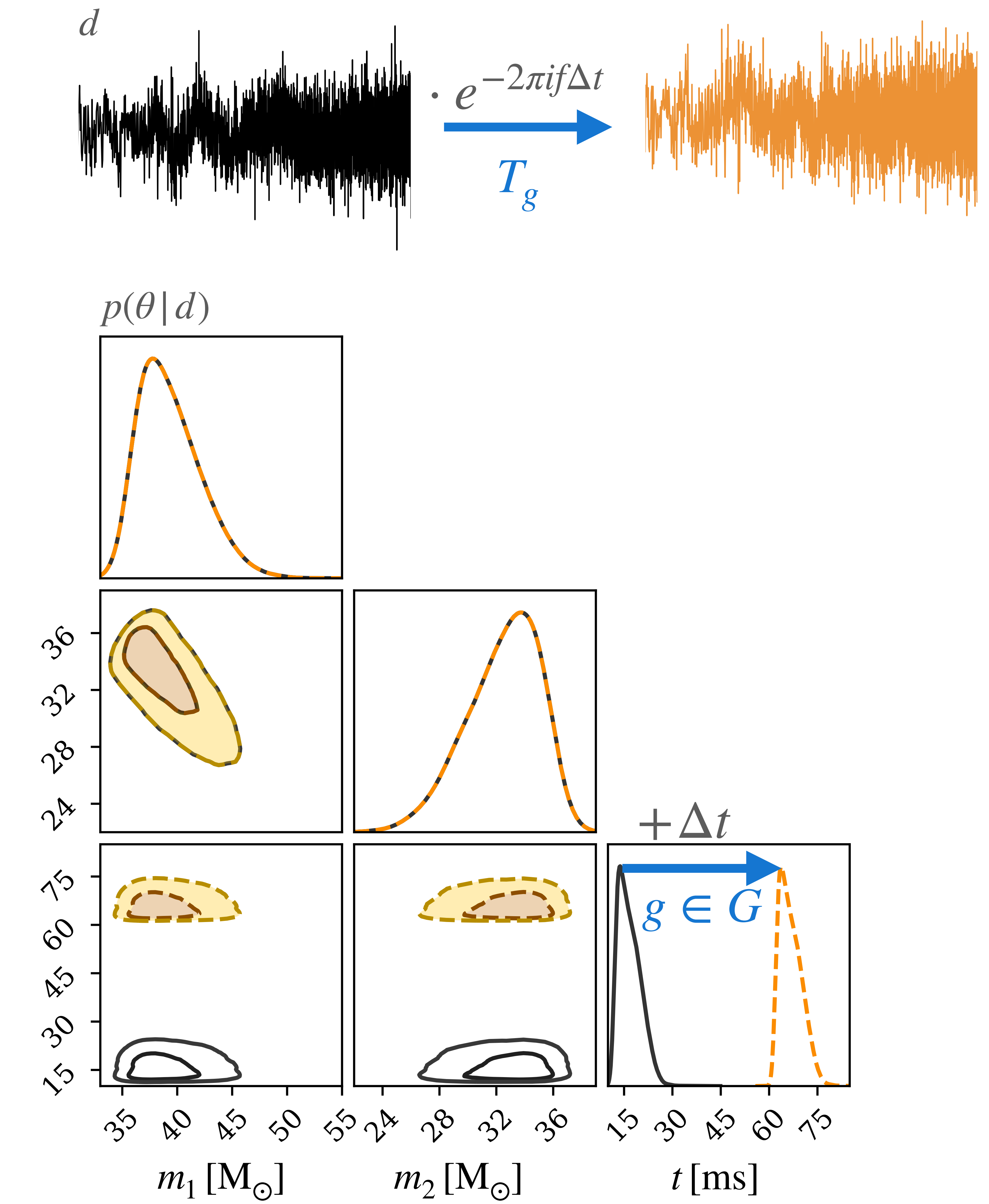
- Equivariance (covariance) under time shift



# NPE with symmetries: Group-equivariant NPE

- Equivariance (covariance) under time shift



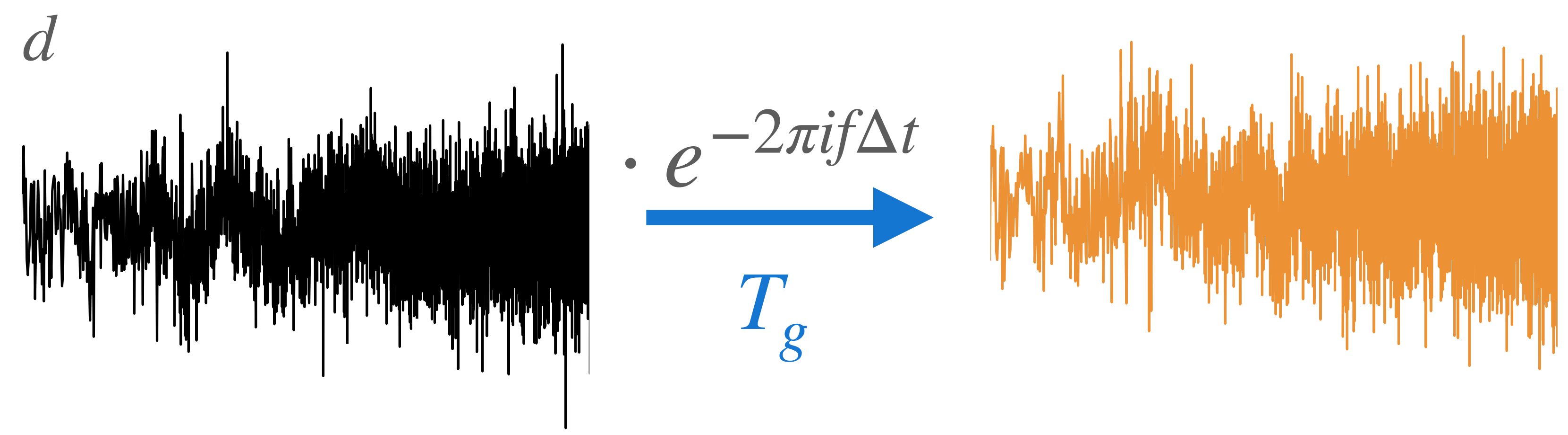
Dax+ (PRL 2021)

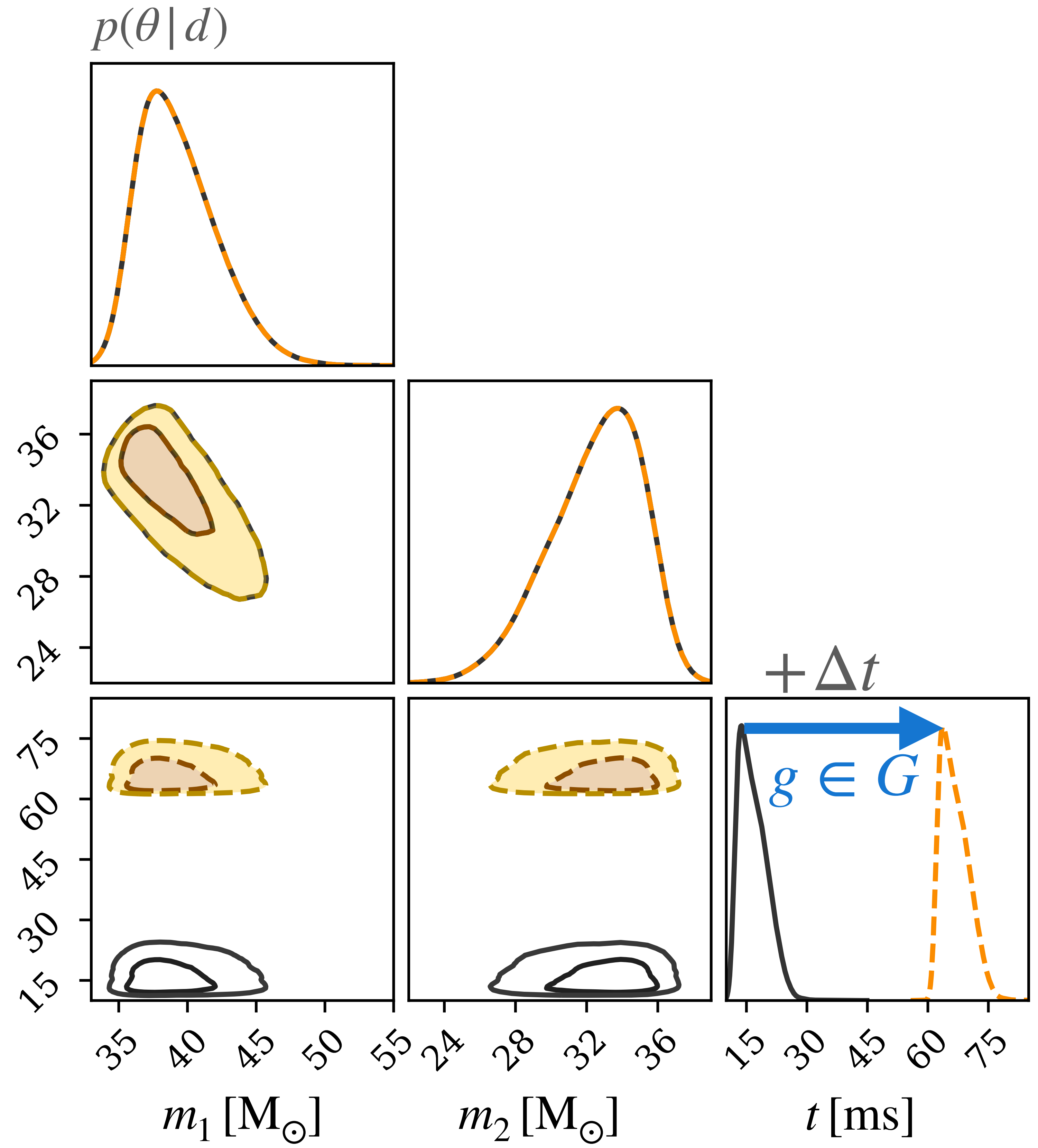
Dax+ (ICLR 2022)

# NPE with symmetries: Group-equivariant NPE

- Equivariance (covariance) under time shift

$$p(\theta|d) = p(g\theta|T_g d)\,|\det J_g| \qquad \forall g \in G$$



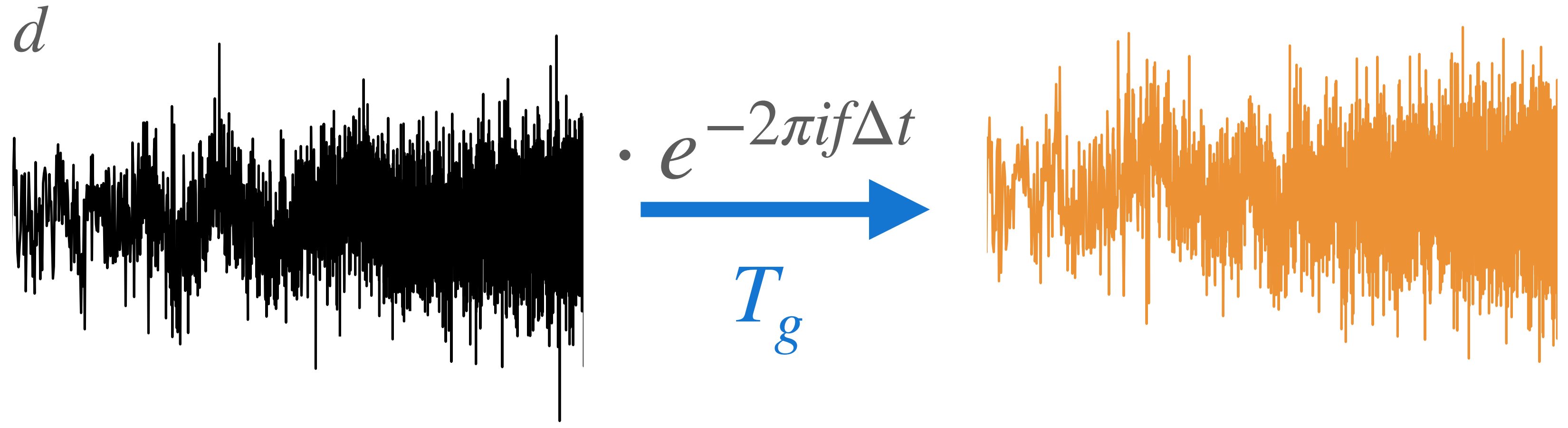



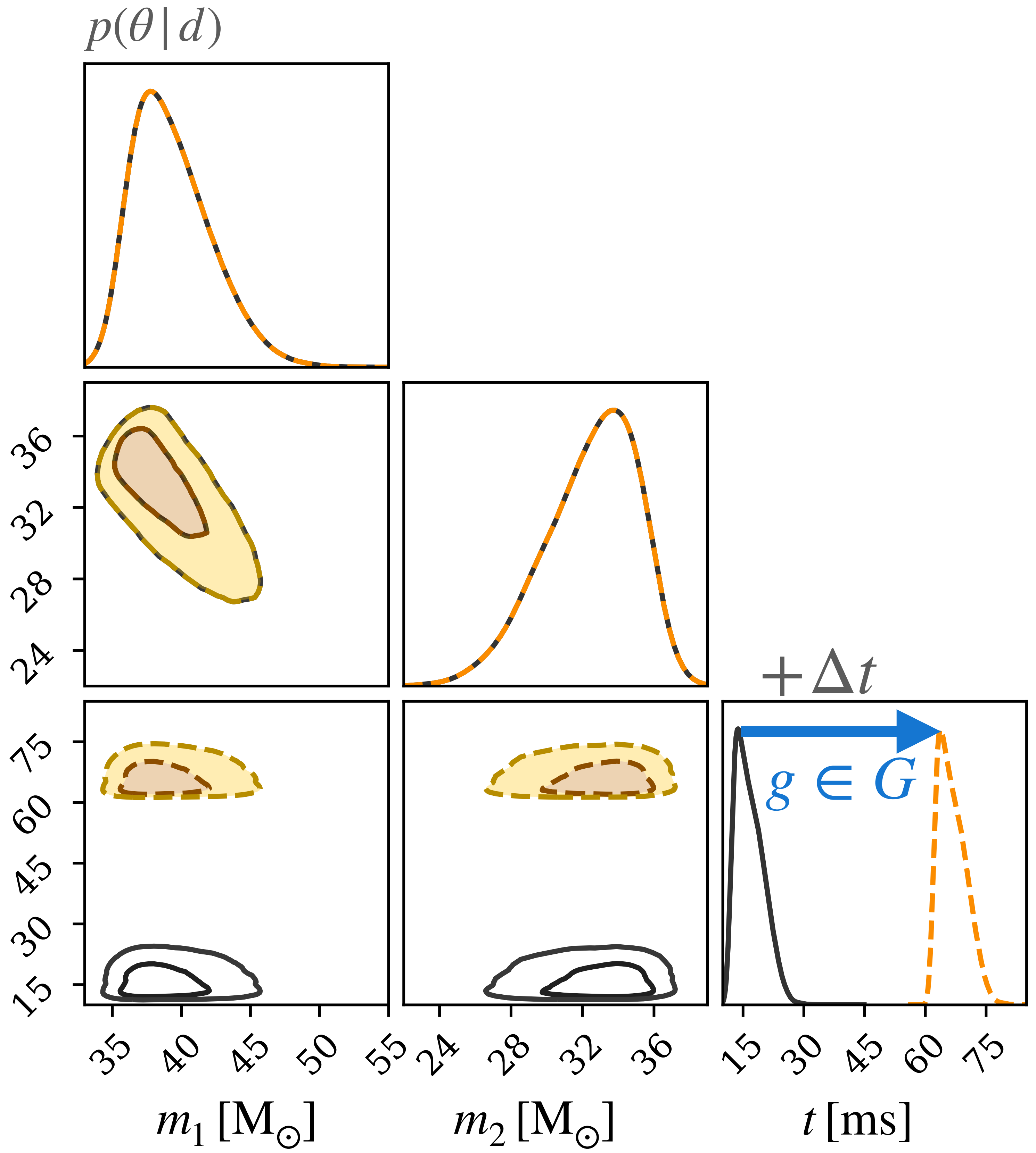
# NPE with symmetries: Group-equivariant NPE

- Equivariance (covariance) under time shift

$$p(\theta|d) = p(g\theta|T_g d)\,|\det J_g| \qquad \forall g \in G$$

- NPE learns such symmetries from simulation data
  ⇒ requires network and training capacity
  ⇒ can we instead **enforce such symmetries**?



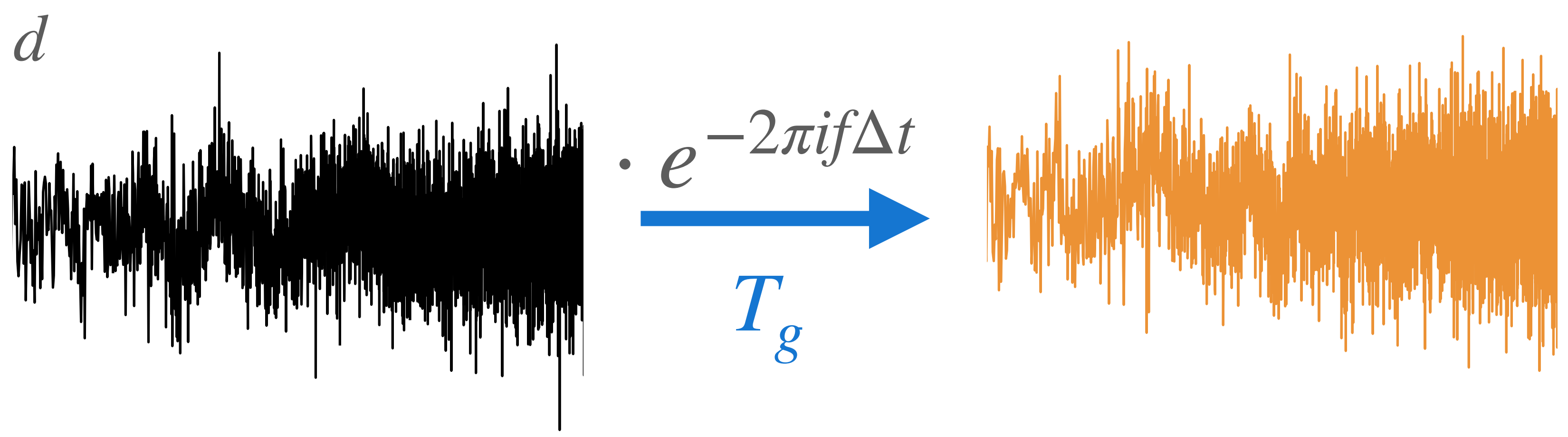
# NPE with symmetries: Group-equivariant NPE

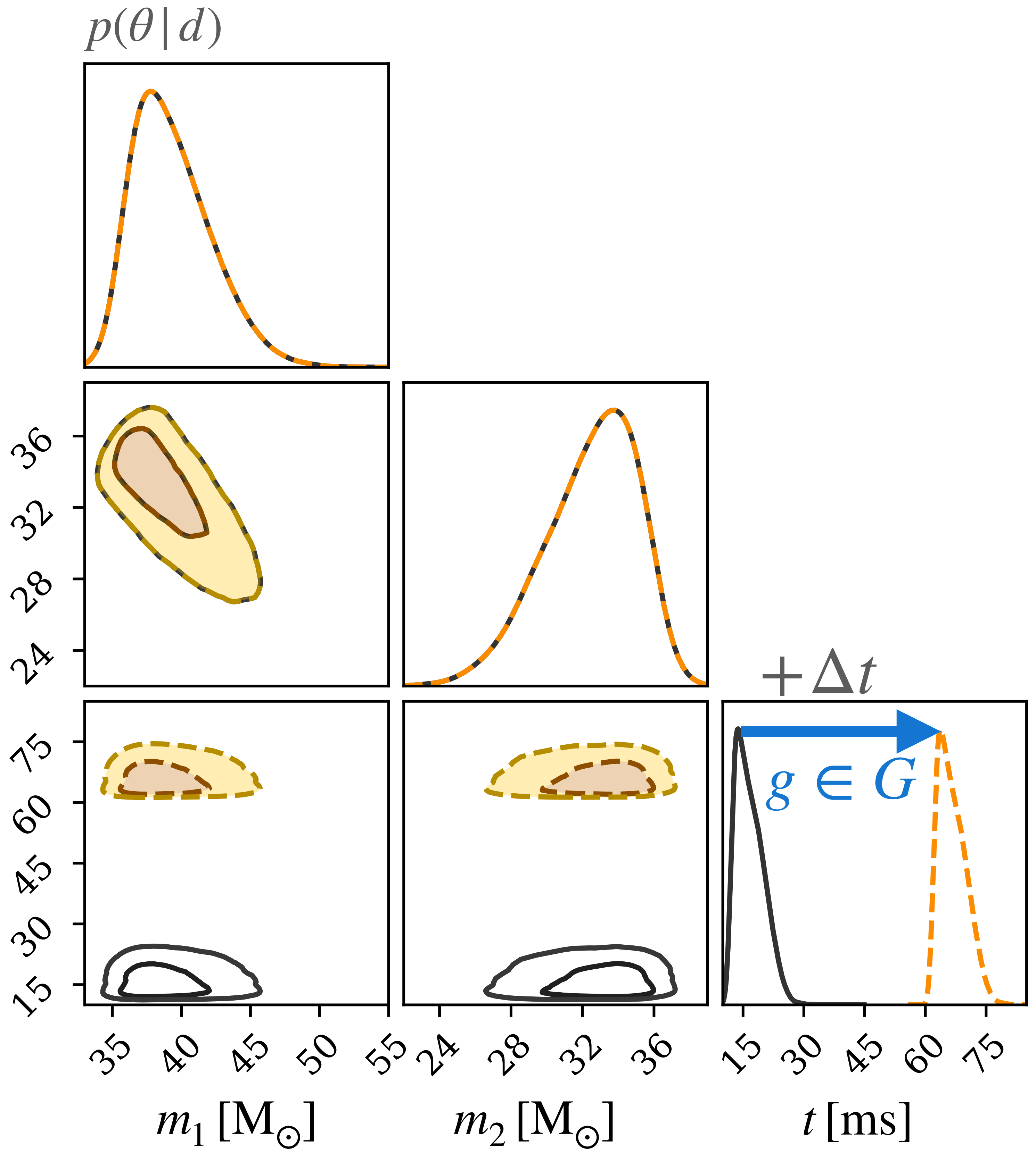
- Equivariance (covariance) under time shift

$$p(\theta|d) = p(g\theta|T_g d)\,|\det J_g| \qquad \forall g \in G$$

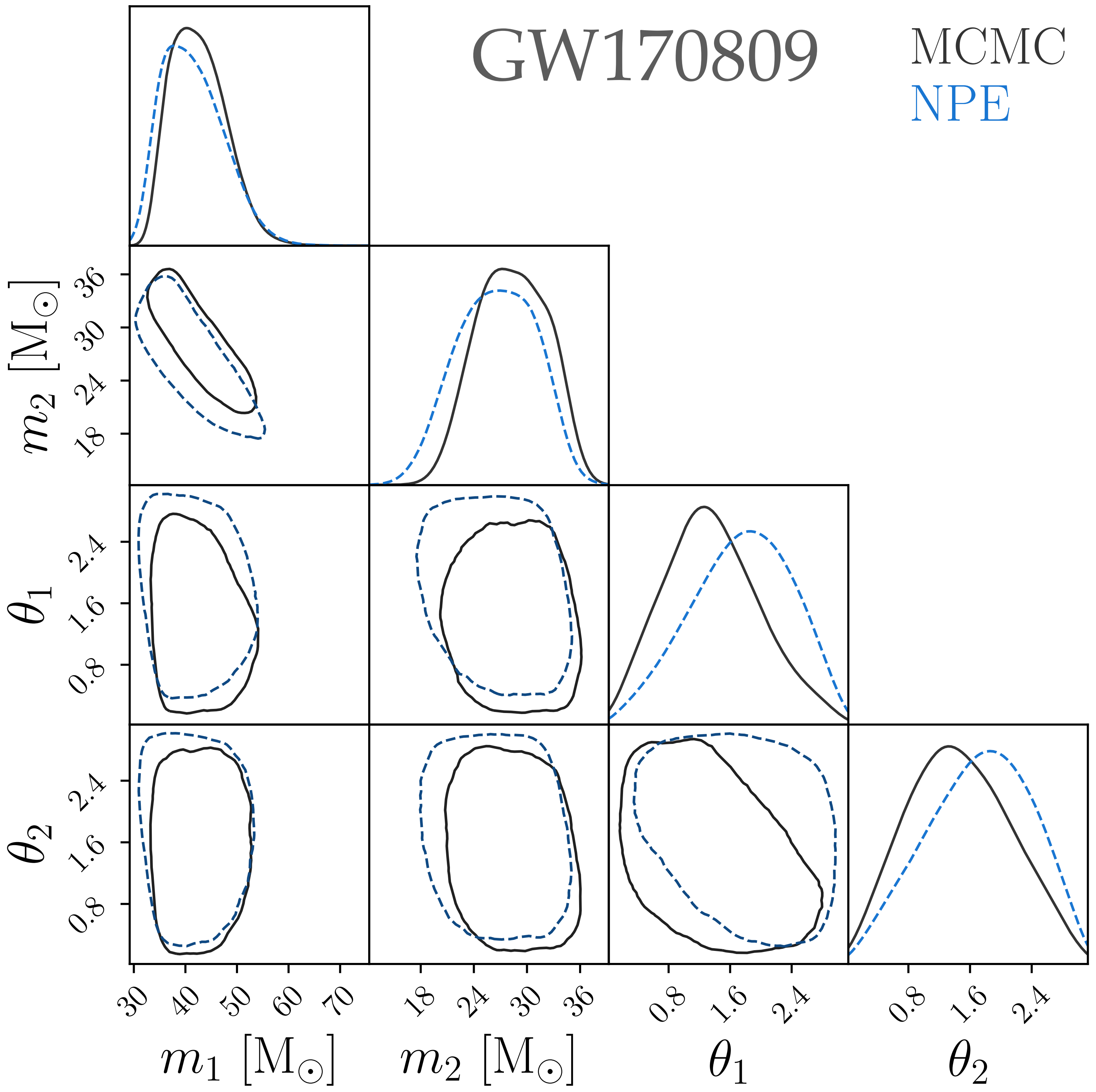
- NPE learns such symmetries from simulation data
  ⇒ requires network and training capacity
  ⇒ can we instead **enforce such symmetries**?

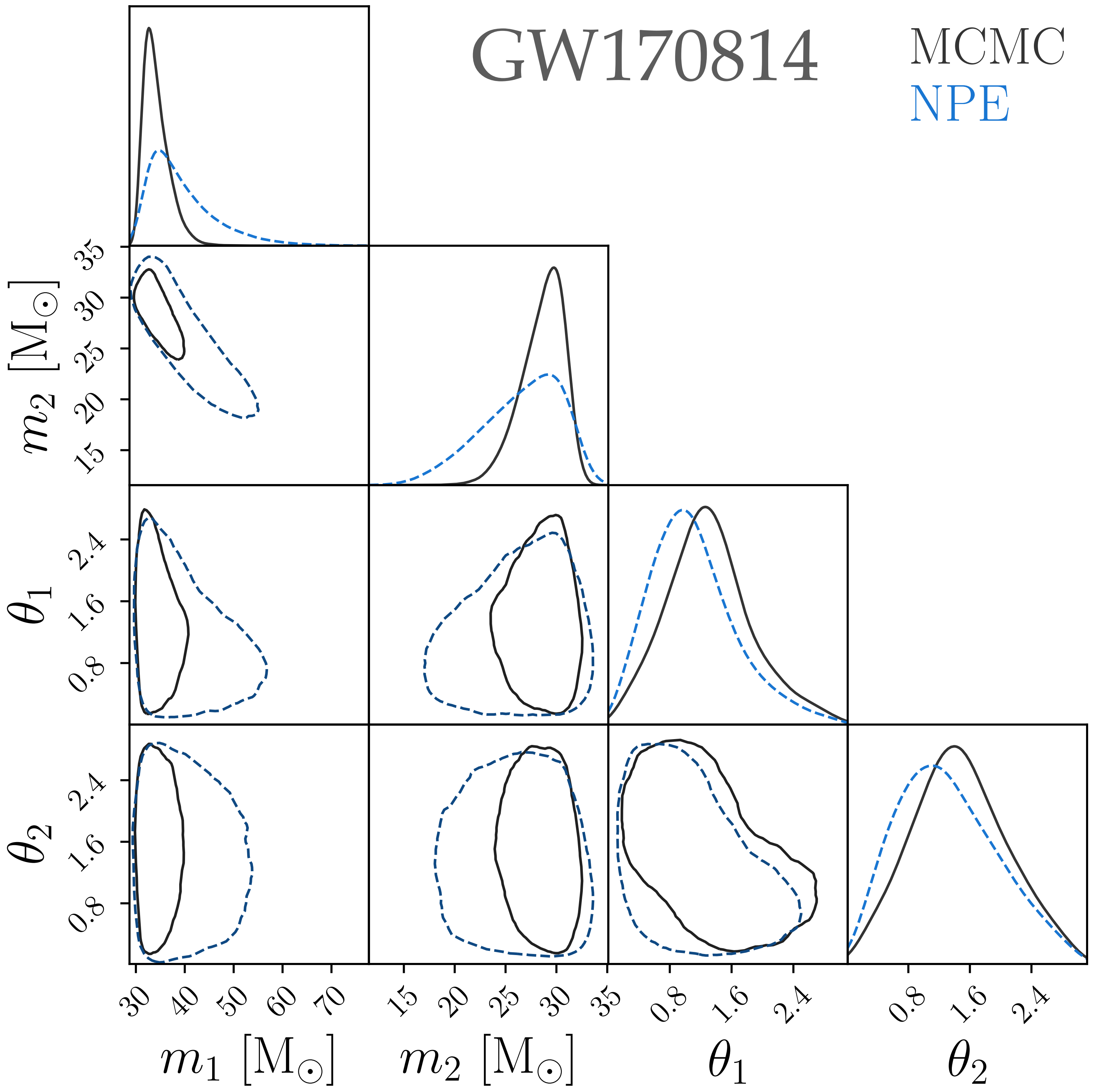
- Group-equivariant NPE (GNPE)
  - Integrate **symmetries via data-standardisation**
    - Define proxy parameter $\hat{t} \approx t$ via a kernel $p(\hat{t}|t) = \kappa(\hat{t} - t)$
    - Train model $q(t|d_{-\hat{t}}, \hat{t})$ conditional on time-shifted strain $d_{-\hat{t}}$
    - Inference with Gibbs sampling using $q(t|d_{-\hat{t}}, \hat{t})$ and $p(\hat{t}|t)$
  - For GWs: great **accuracy improvements**
  - Compatible with **exact** and **approximate** symmetries



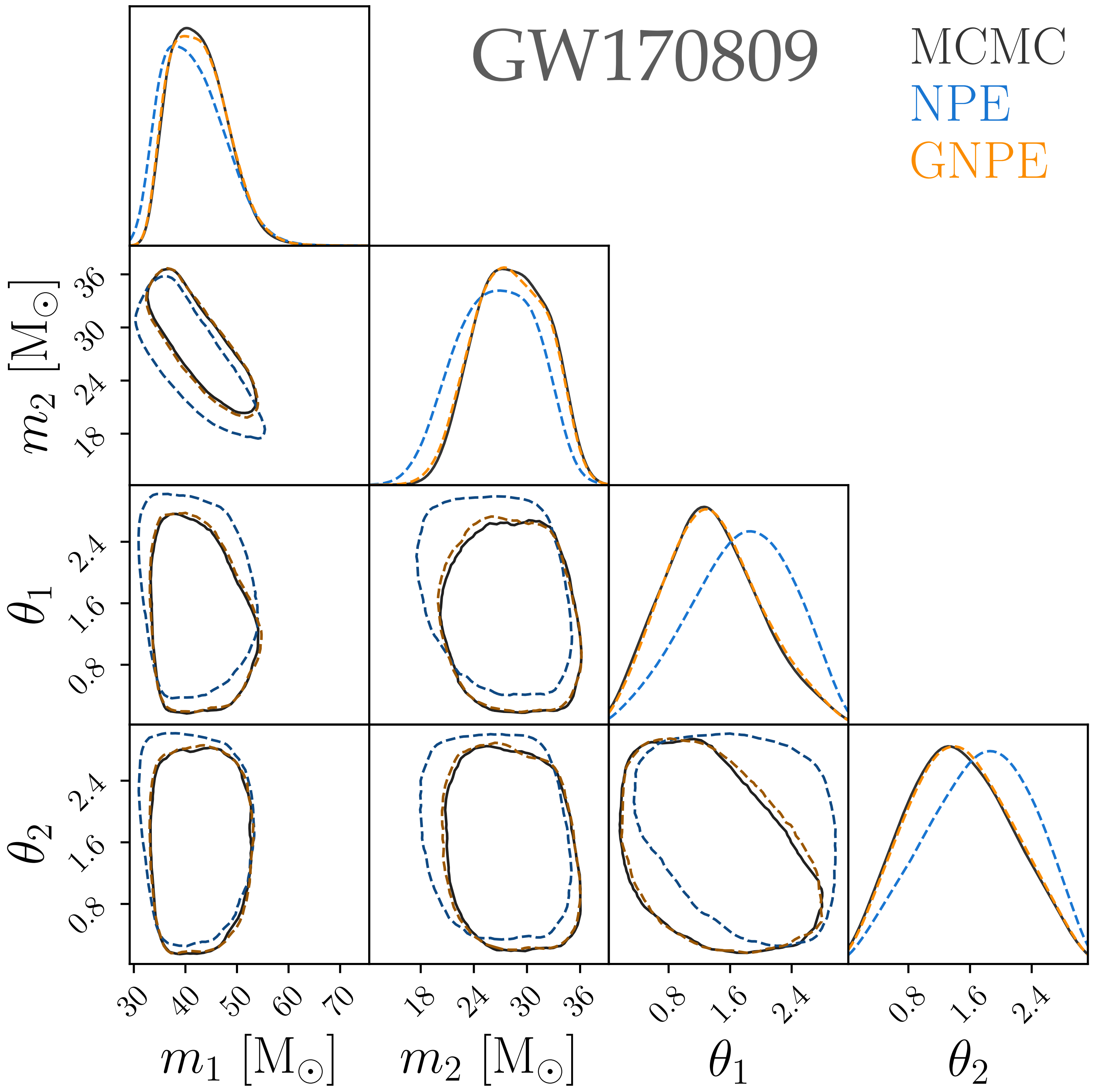



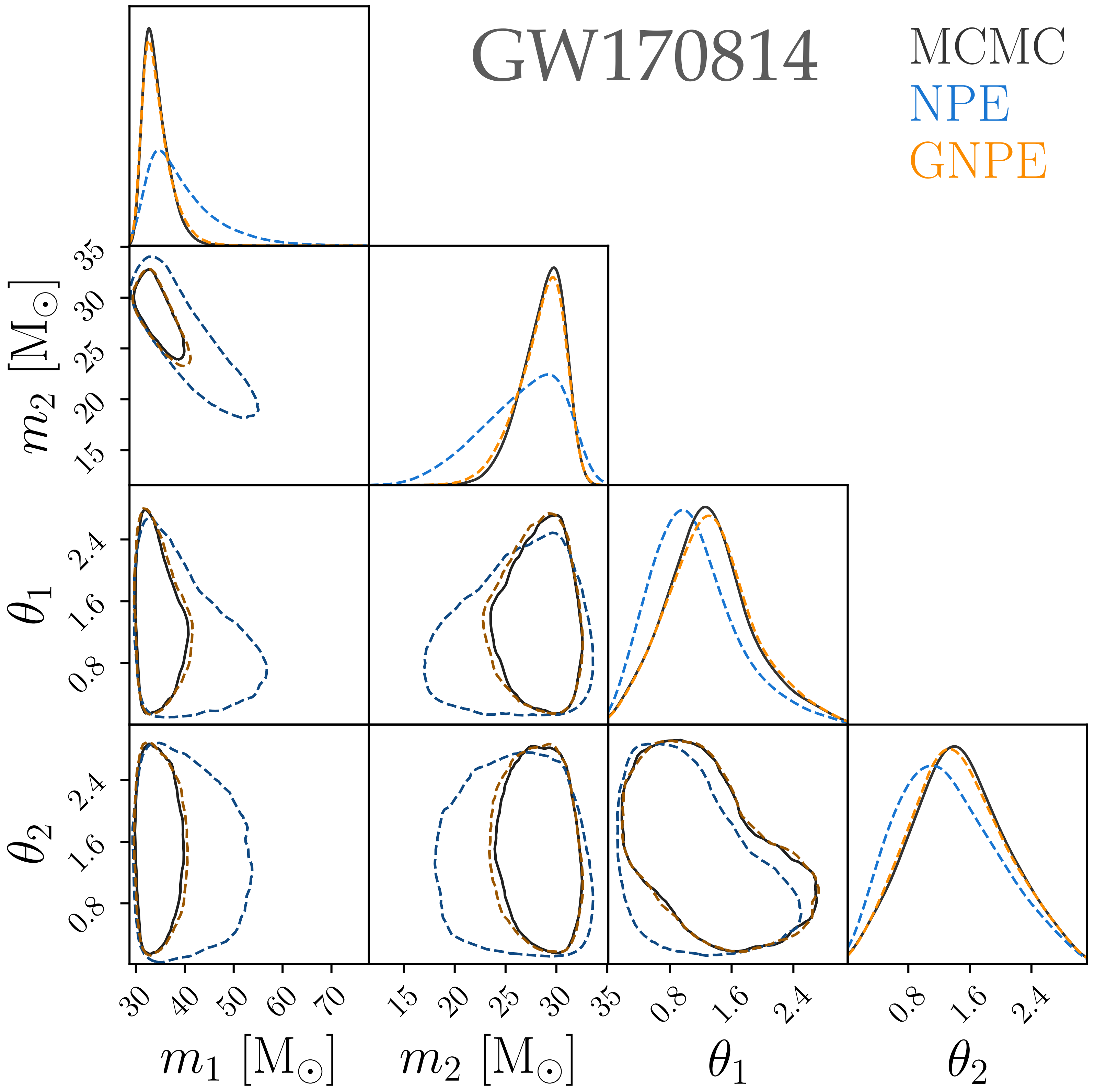
# NPE with symmetries: Group-equivariant NPE



# NPE with symmetries: Group-equivariant NPE

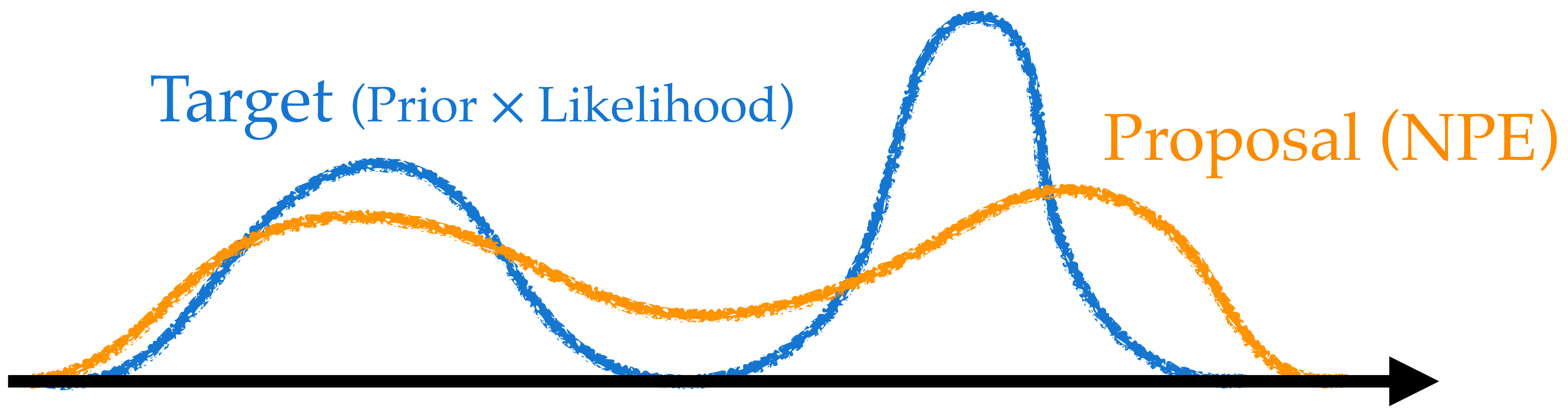


# Importance-sampled NPE (NPE-IS)

# Importance-sampled NPE (NPE-IS)

- If likelihood is tractable, can reweight NPE results

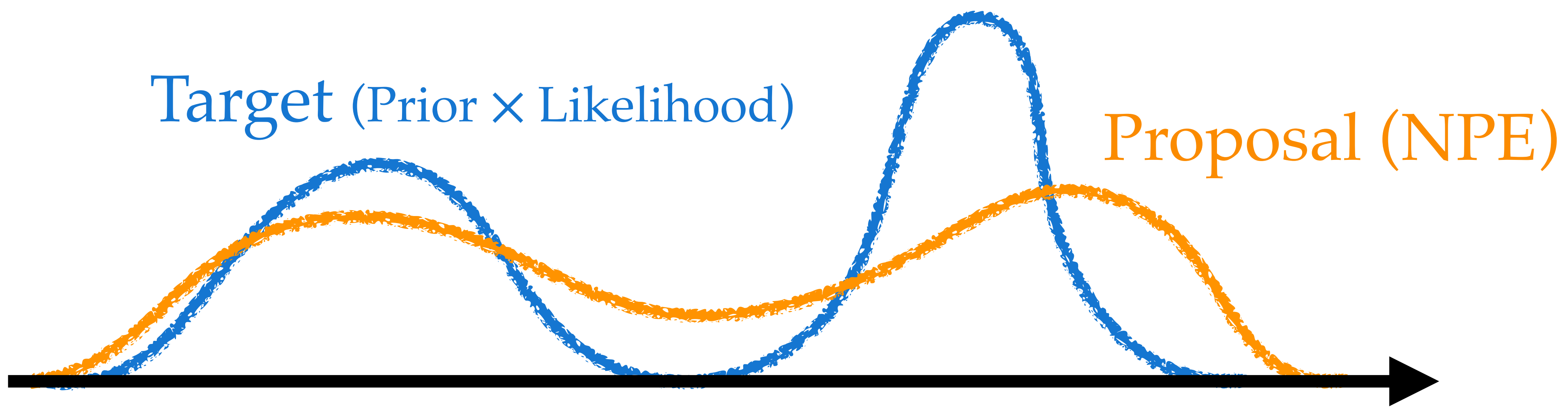



$$\theta_i \sim q(\theta | d)$$

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

# Importance-sampled NPE (NPE-IS)

- If likelihood is tractable, can reweight NPE results




$$\theta_i \sim q(\theta|d)$$

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

- **Effective number of samples** as performance metric

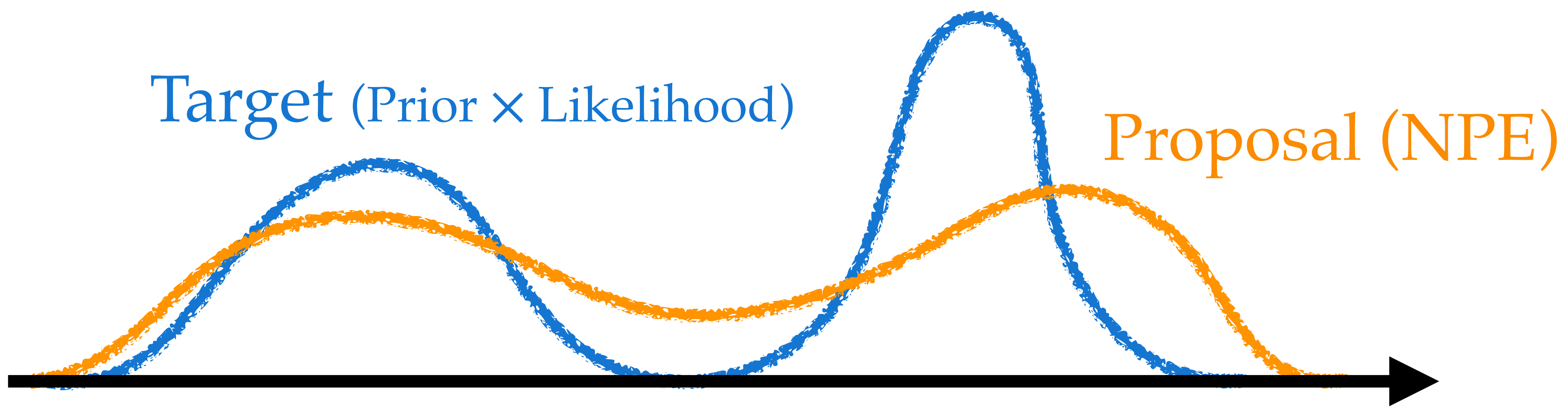
$$n_{\text{eff}} = \left(\Sigma_i w_i\right)^2 / \Sigma_i \left(w_i^2\right) \qquad \epsilon = n_{\text{eff}}/n \in (0,1]$$

- Estimate of **Bayesian evidence**

$$p(d) = \frac{1}{n}\sum_i w_i \qquad \sigma_{\log p(d)} = \sqrt{(1-\epsilon)/(n \cdot \epsilon)}$$

# Importance-sampled NPE (NPE-IS)

- If likelihood is tractable, can reweight NPE results




$$\theta_i \sim q(\theta|d)$$

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

⇒ **asymptotically exact**

- **Effective number of samples** as performance metric

$$n_\text{eff} = \left(\Sigma_i w_i\right)^2 / \Sigma_i \left(w_i^2\right) \qquad \epsilon = n_\text{eff}/n \in (0,1]$$

⇒ **verification** without for ground truth posterior

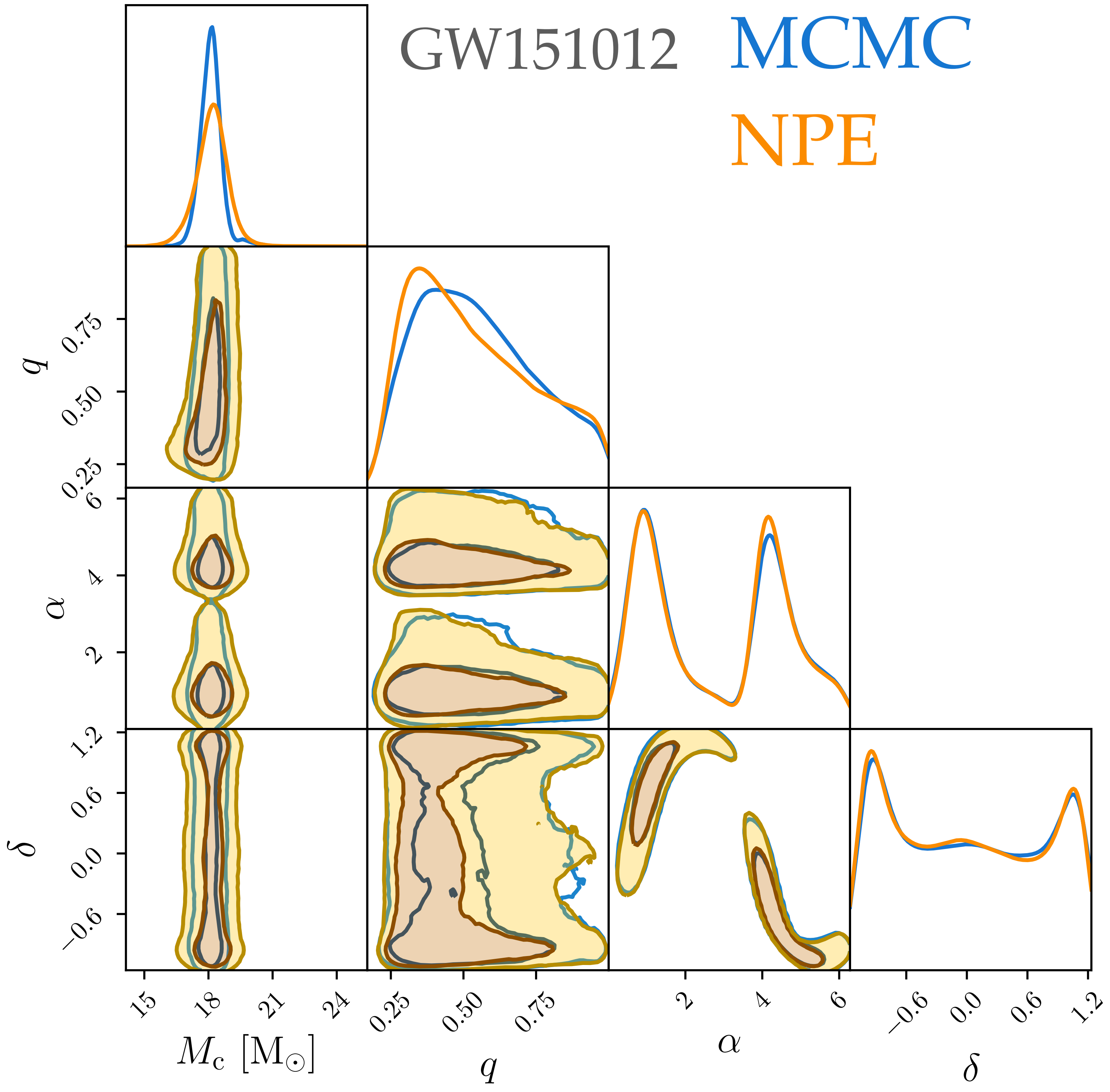
- Estimate of **Bayesian evidence**

$$p(d) = \frac{1}{n}\sum_i w_i \qquad \sigma_{\log p(d)} = \sqrt{(1-\epsilon)/(n\cdot\epsilon)}$$

⇒ unbiased & precise estimate of **evidence**

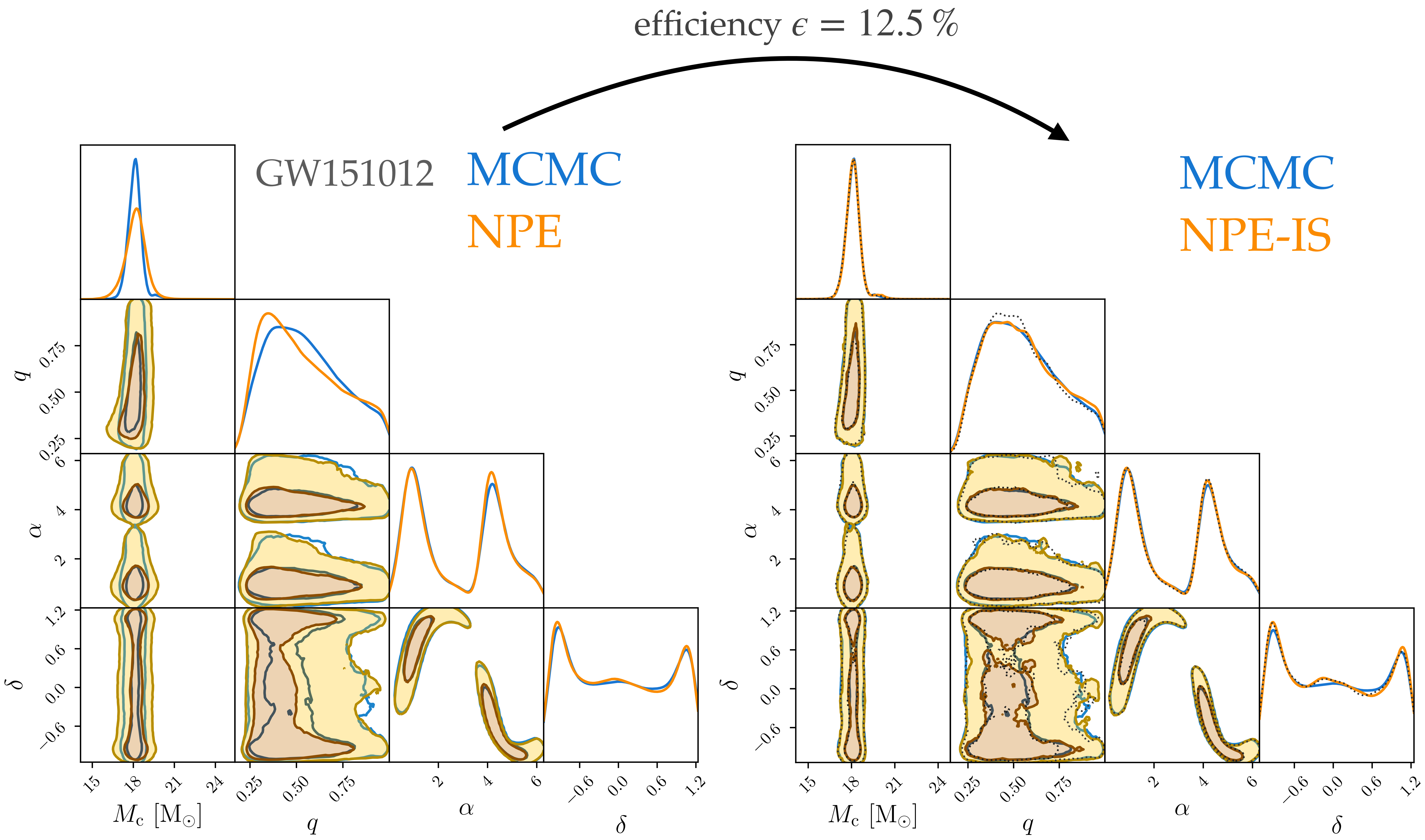
# Importance-sampled NPE (NPE-IS)

- Evaluation on 42 real GW events, **efficiencies of ≈ 10%**



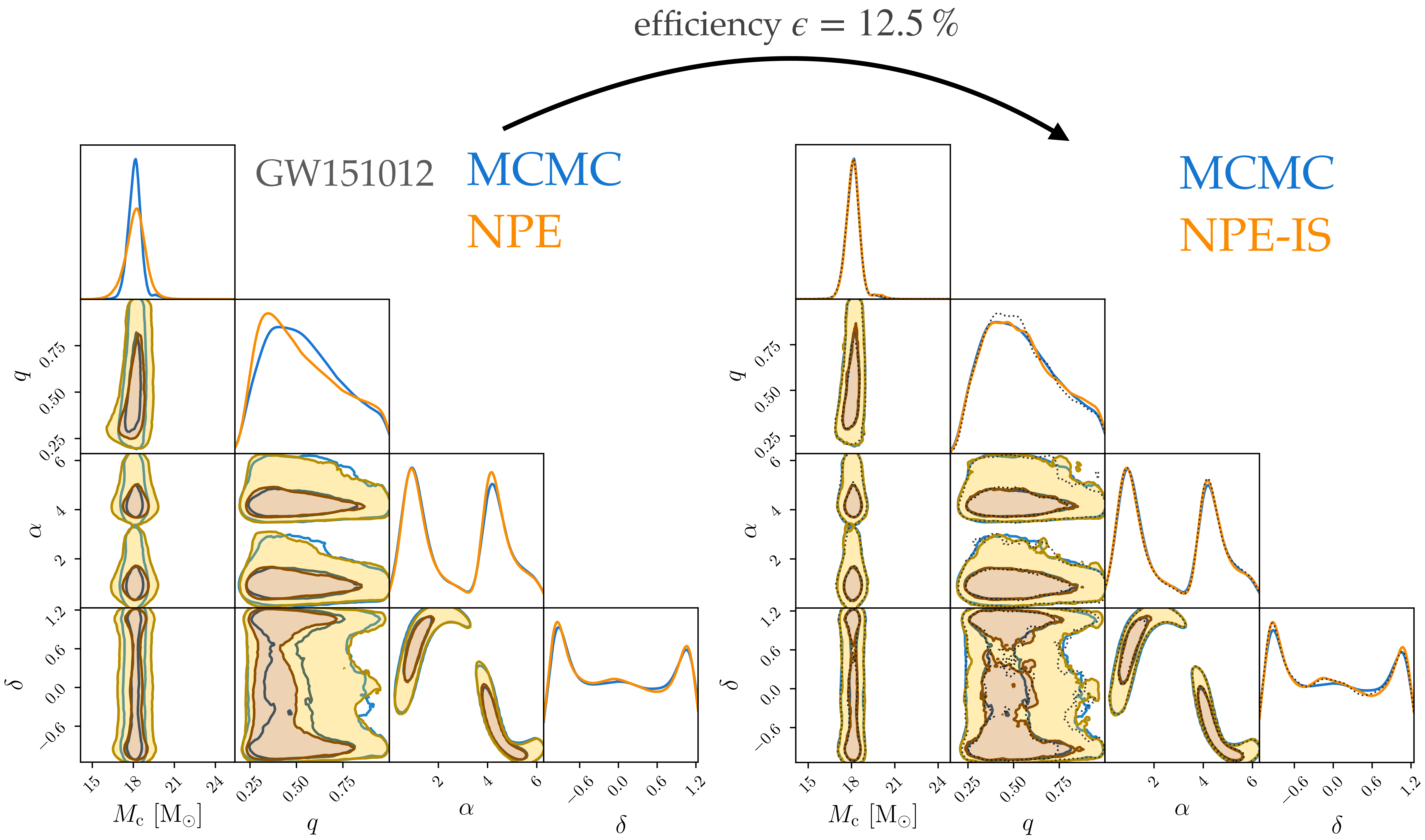
# Importance-sampled NPE (NPE-IS)

- Evaluation on 42 real GW events, **efficiencies of ≈ 10%**



# Importance-sampled NPE (NPE-IS)

- Evaluation on 42 real GW events, **efficiencies of ≈ 10%**
- Can use GW models for which MCMC is too costly
- Low efficiencies **flag OOD data** and adversarial attacks
- **Evidences** consistent with nested sampling, but **10x more precise**



# Importance-sampled NPE (NPE-IS)

- Whenever the likelihood is tractable, we can combine **NPE** with **importance sampling**

- This provides an **independent verification and correction** of results
  - Improve performance at inference
  - Sample efficiency as independent performance metric
  - Precise and unbiased estimate of Bayesian evidence

- IS is applicable as NPE results are probability mass covering

- Caveat: **NPE-IS is extremely aggressive**. When it works, it has strong guarantees, but when it fails it does not mean that the initial NPE results are bad.

# NPE for binary neutron stars

# Masses in the Stellar Graveyard

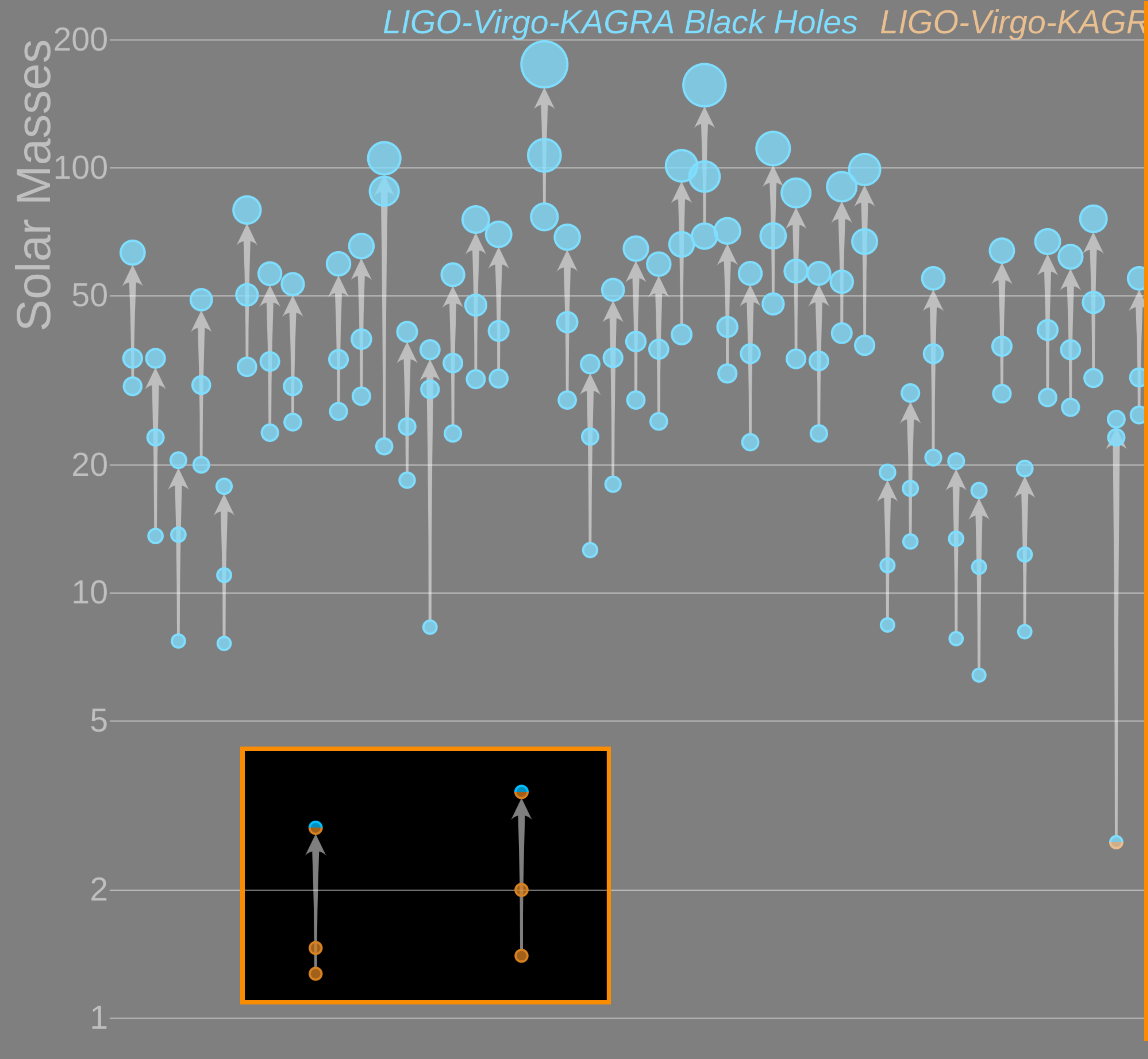

LIGO-Virgo-KAGRA | Aaron Geller | Northwestern

# Masses in the Stellar Graveyard


**Binary neutron stars (BNS)**

- Signals 10-20x longer than BBH
  $\Rightarrow$ Extremely **challenging for ML**
- May emit electromagnetic follow-up
  $\Rightarrow$ **Fast inference critical** to enable EM search

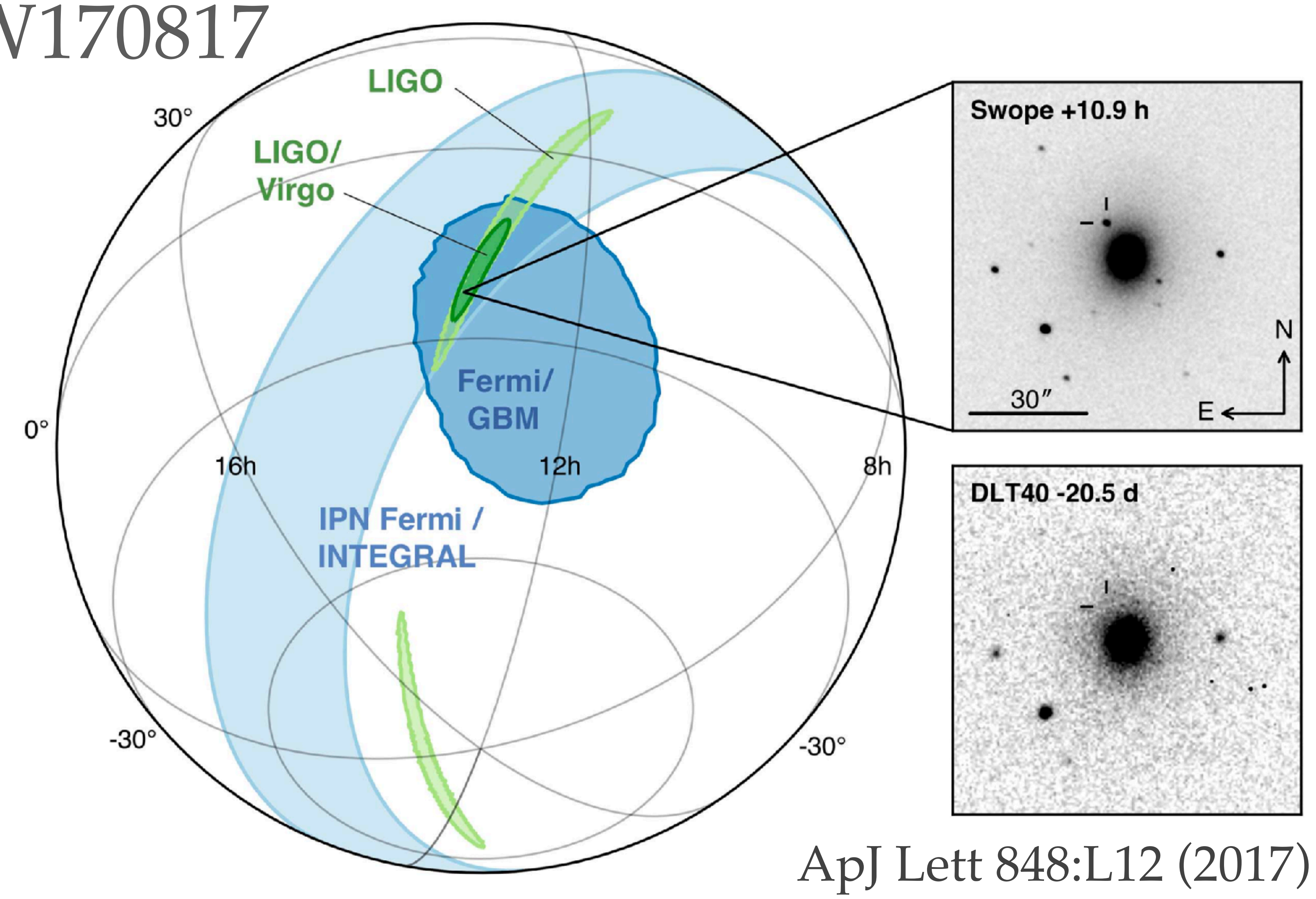
# Masses in the Stellar Graveyard




**Binary neutron stars (BNS)**

- Signals 10-20x longer than BBH
  ⇒ Extremely **challenging for ML**
- May emit electromagnetic follow-up
  ⇒ **Fast inference critical** to enable EM search

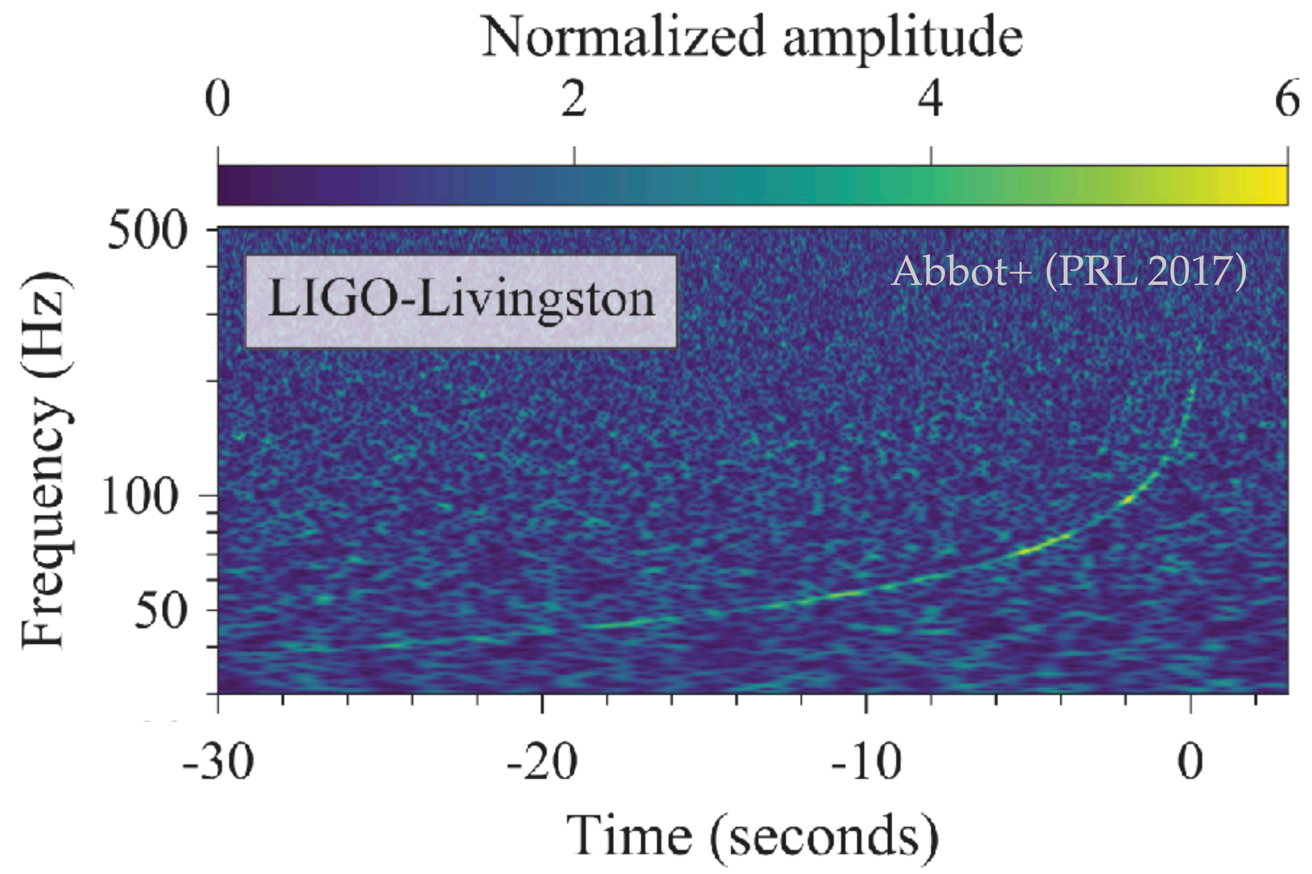
GW170817




ApJ Lett 848:L12 (2017)

# BNS: Prior-conditioning

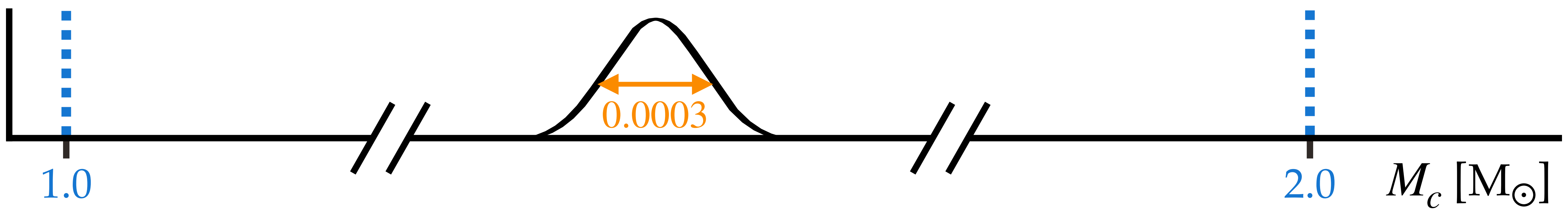
- BNS data constrains the chirp mass $M_c$ extremely well
  - **Between events**: large chirp mass range, e.g. [1.0, 2.0]
  - **Specific event**: tightly constrained posterior

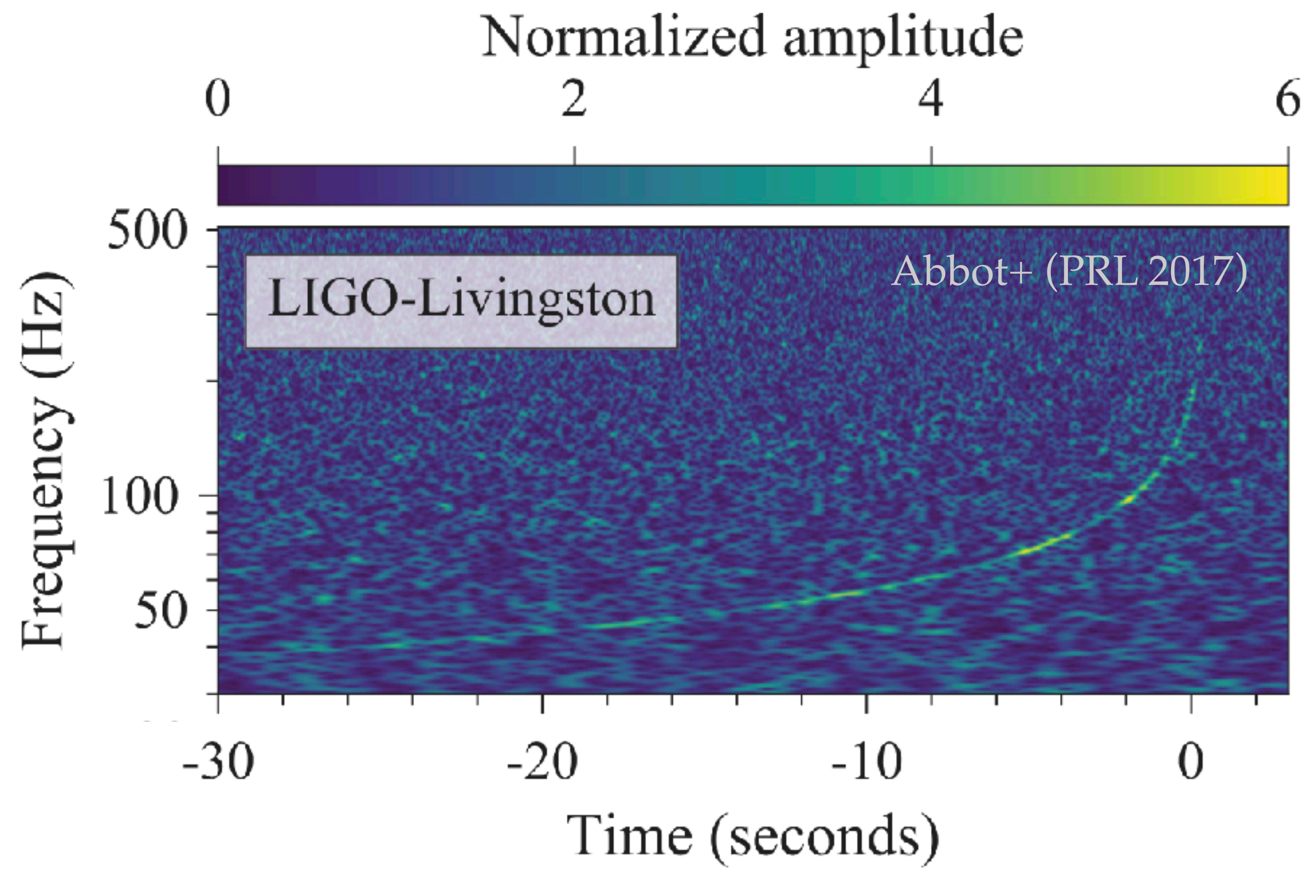




$$M_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

# BNS: Prior-conditioning

- BNS data constrains the chirp mass $M_c$ extremely well
  - **Between events**: large chirp mass range, e.g. [1.0, 2.0]
  - **Specific event**: tightly constrained posterior

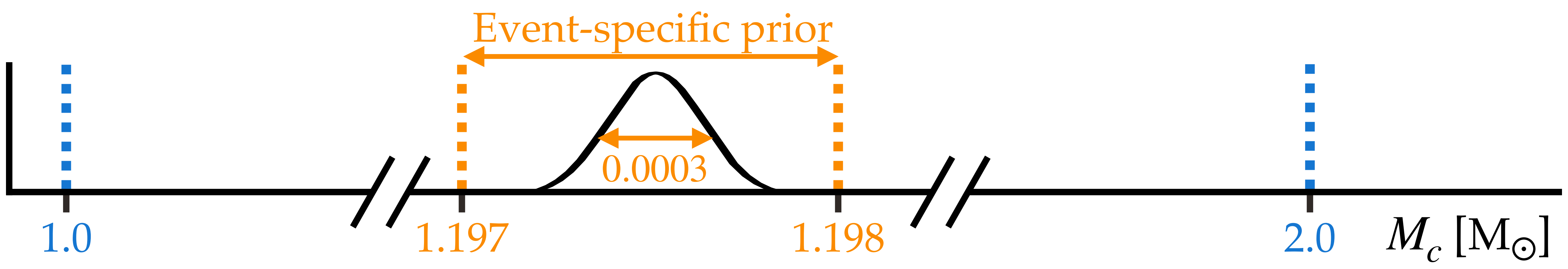


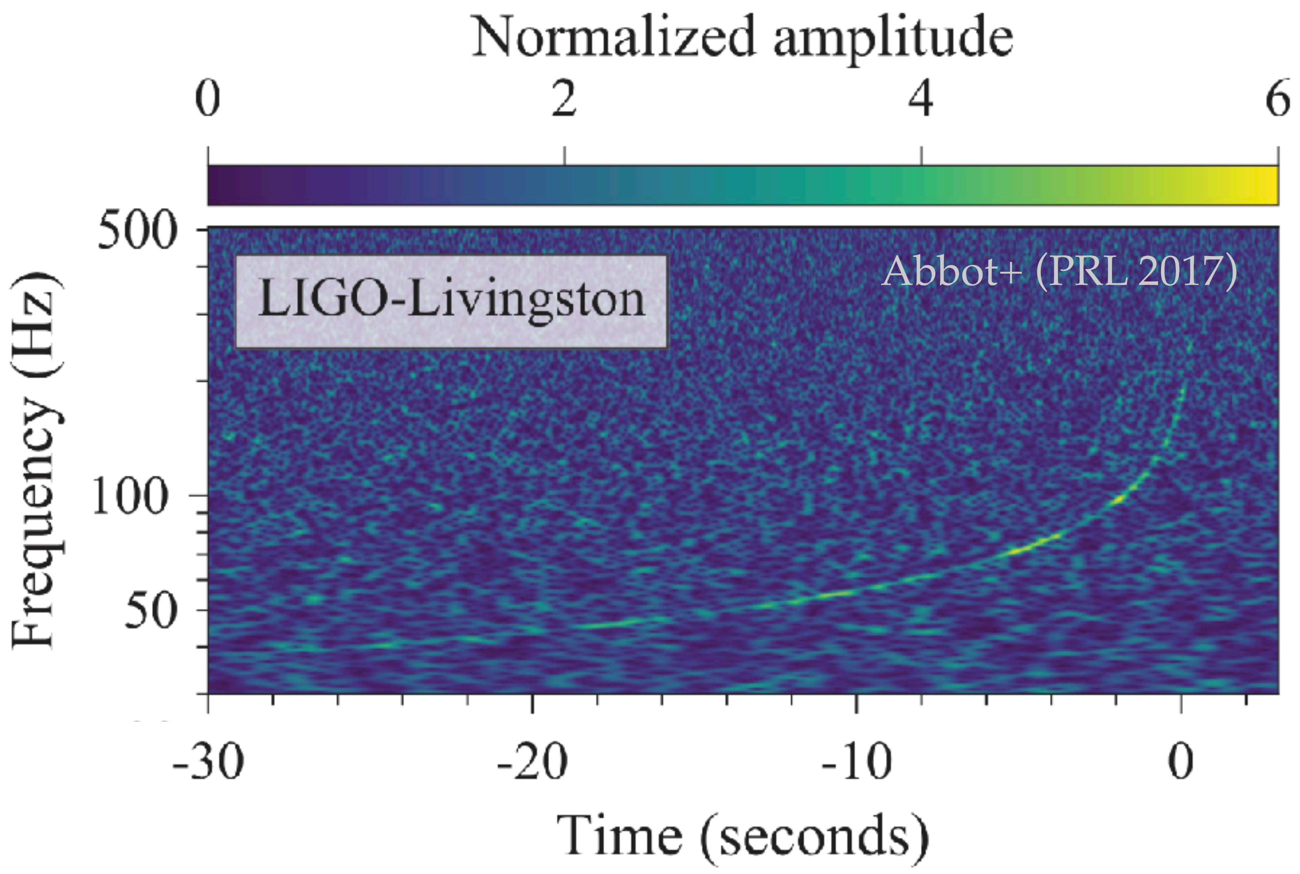




$$M_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

# BNS: Prior-conditioning

- BNS data constrains the chirp mass $M_c$ extremely well
  - **Between events**: large chirp mass range, e.g. [1.0, 2.0]
  - **Specific event**: tightly constrained posterior

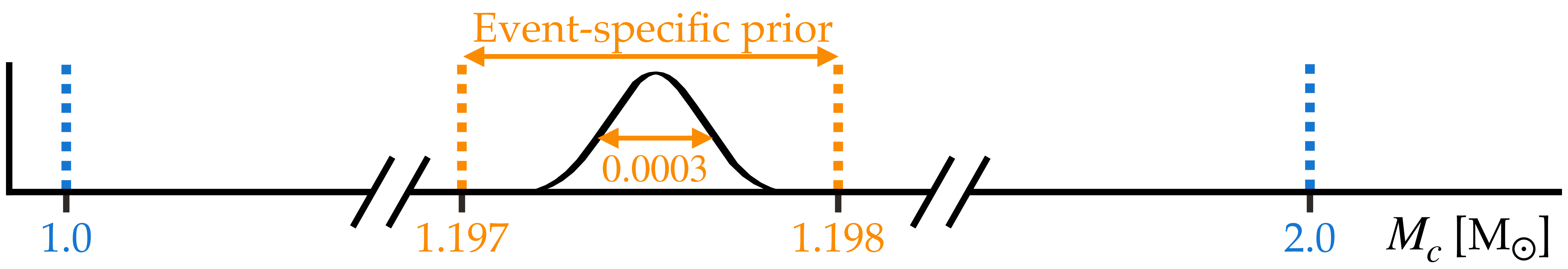


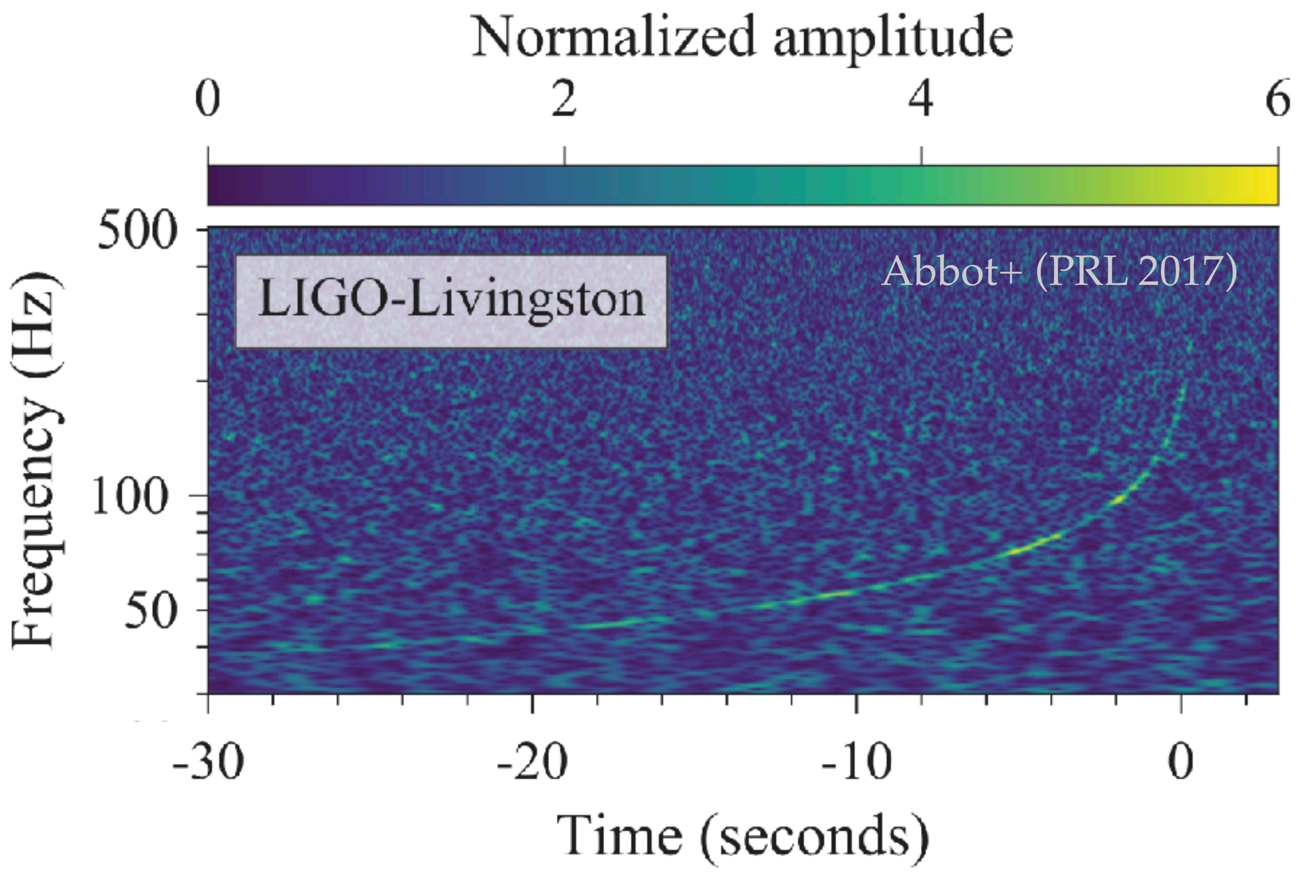




$$M_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

# BNS: Prior-conditioning

- BNS data constrains the chirp mass $M_c$ extremely well
  - **Between events**: large chirp mass range, e.g. [1.0, 2.0]
  - **Specific event**: tightly constrained posterior



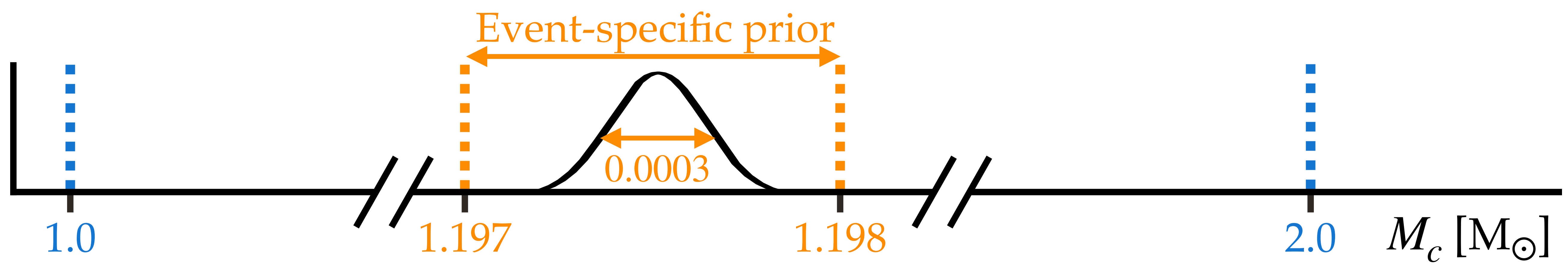

$$M_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

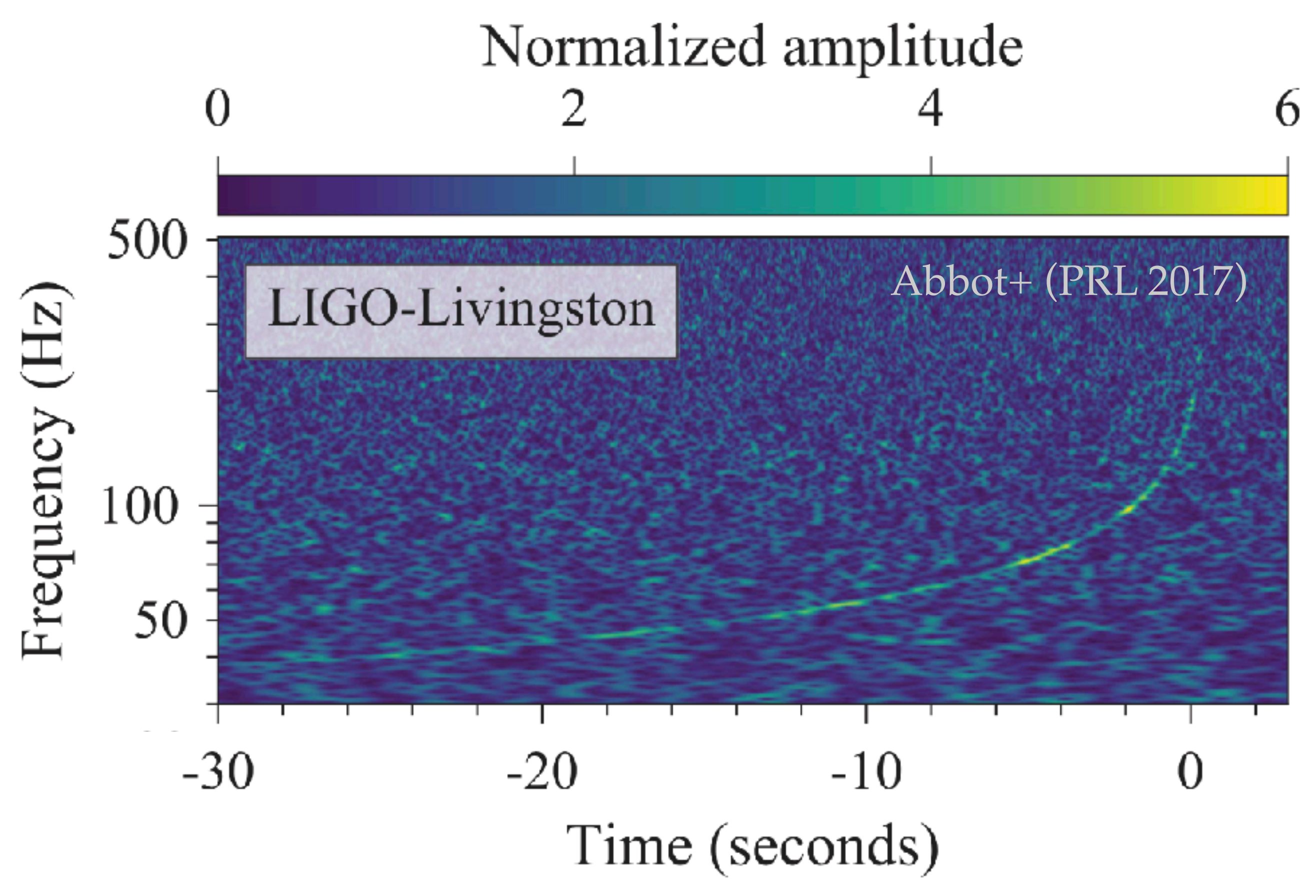
- **Prior-conditioning** enables prior-tunable SBI networks

1) Sample the training prior hierarchically $\rho_i \sim \hat{p}(\rho),\, \theta_i \sim p_{\rho_i}(\theta)$

# BNS: Prior-conditioning

- BNS data constrains the chirp mass $M_c$ extremely well
  - **Between events**: large chirp mass range, e.g. [1.0, 2.0]
  - **Specific event**: tightly constrained posterior




$$M_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

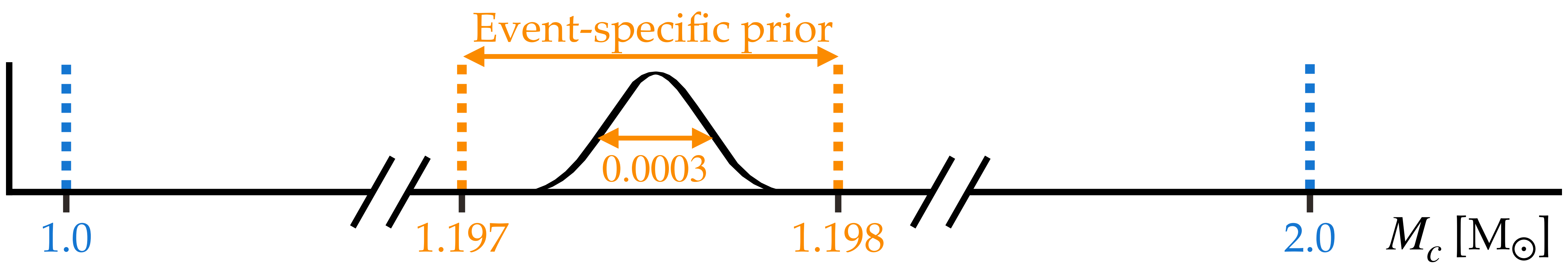
- **Prior-conditioning** enables prior-tunable SBI networks

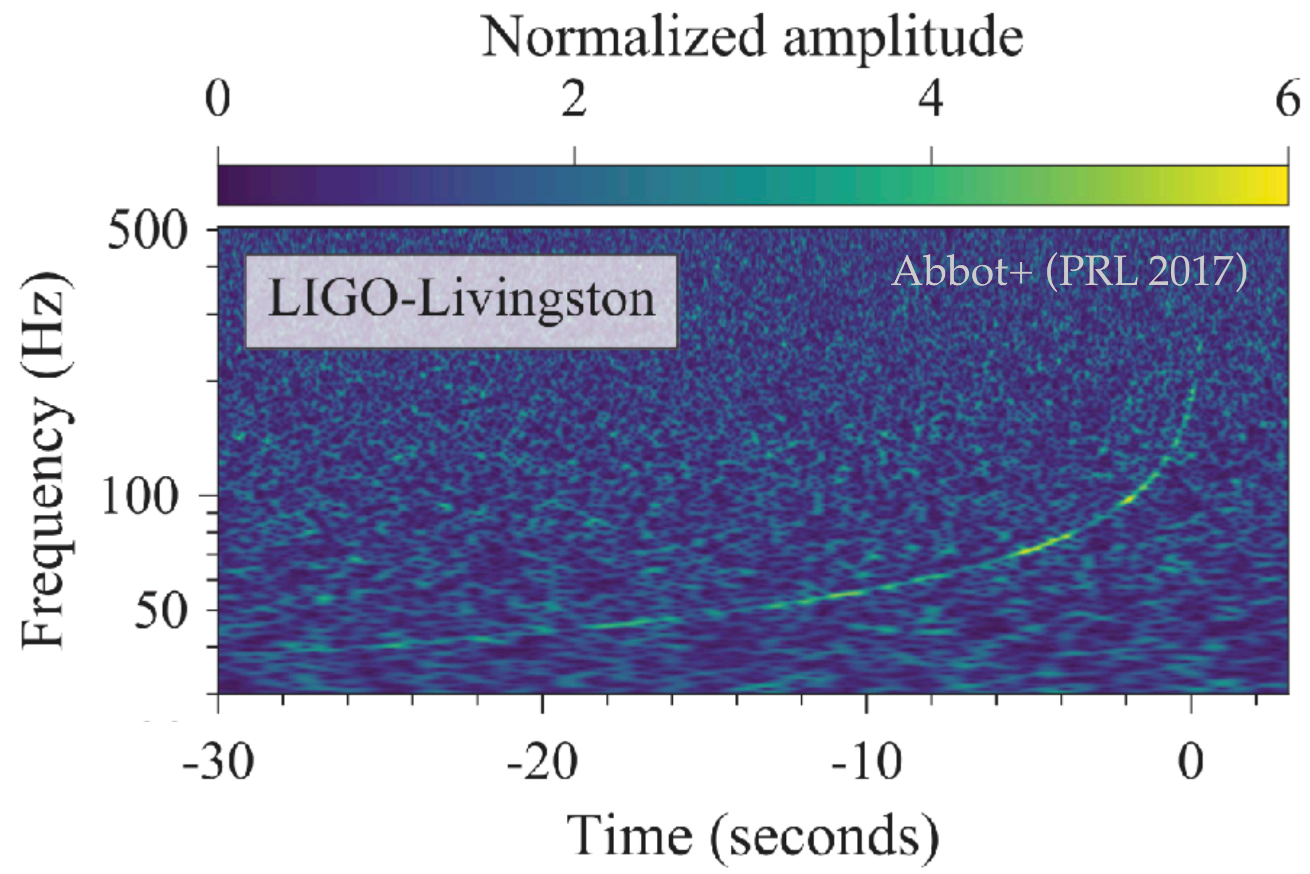
1) Sample the training prior hierarchically $\rho_i \sim \hat{p}(\rho),\, \theta_i \sim p_{\rho_i}(\theta)$

2) Condition SBI network on choice of prior $q(\theta \,|\, d, \rho)$

# BNS: Prior-conditioning

- BNS data constrains the chirp mass $M_c$ extremely well
  - **Between events**: large chirp mass range, e.g. [1.0, 2.0]
  - **Specific event**: tightly constrained posterior




$$M_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

- **Prior-conditioning** enables prior-tunable SBI networks

1) Sample the training prior hierarchically $\rho_i \sim \hat{p}(\rho),\, \theta_i \sim p_{\rho_i}(\theta)$

2) Condition SBI network on choice of prior $q(\theta \,|\, d, \rho)$

**For BNS:**

$\hat{p}(\rho) = U[1.0, 2.0]\ \mathrm{M}_\odot$

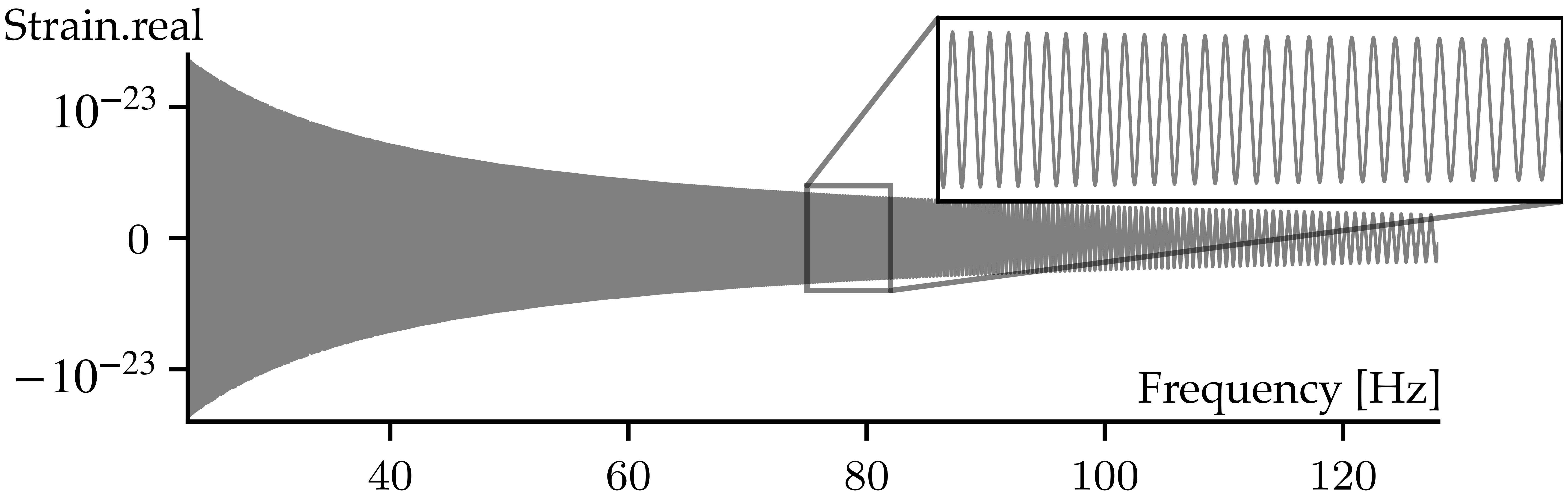
$p_\rho(M_c) = U[\rho - 0.005\,\mathrm{M}_\odot, \rho + 0.005\,\mathrm{M}_\odot]$

# BNS: Data compression

- **Challenge:** BNS signals are longer and more complex than BBH
- **Solution:** Prior conditioning enables **$M_c$-based compression**

Prior-conditioned network includes $M_c$ estimate $\rho = M_c^{\text{est}}$

$$q(\theta \,|\, d, M_c^{\text{est}})$$



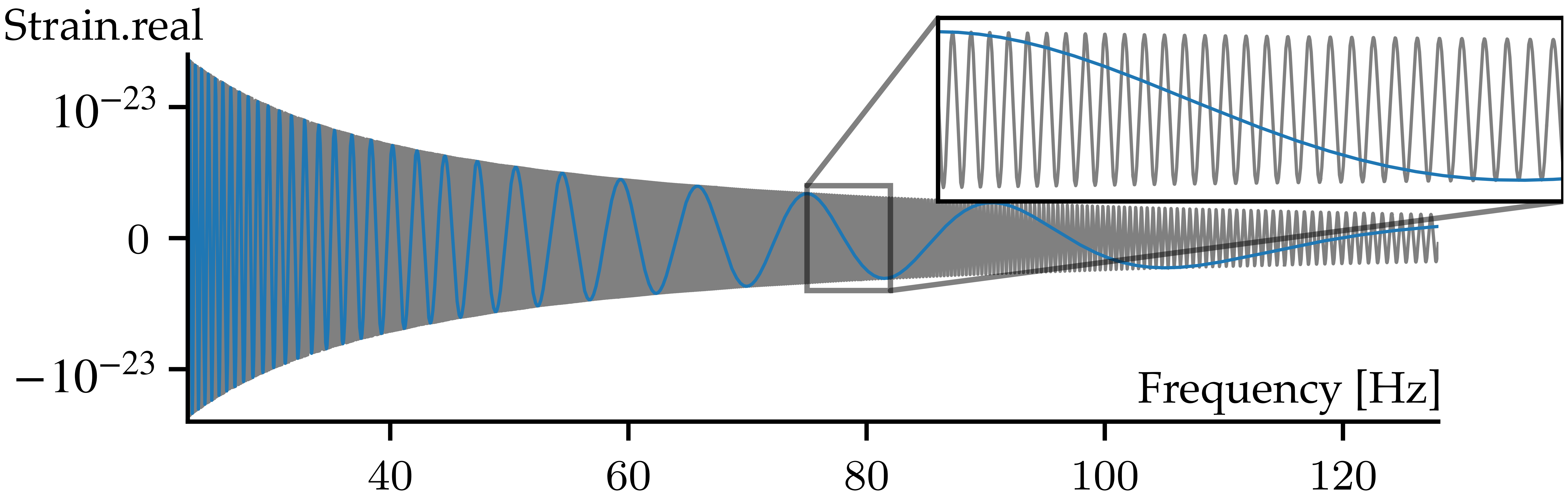
# BNS: Data compression

- **Challenge:** BNS signals are longer and more complex than BBH
- **Solution:** Prior conditioning enables **$M_c$-based compression**

1. **Heterodyning** (Cornish 2010) — factor out overall phase $\propto (M_c^{\text{est}} f)^{-5/3}$

Prior-conditioned network includes $M_c$ estimate $\rho = M_c^{\text{est}}$

$q(\theta \,|\, d, M_c^{\text{est}})$



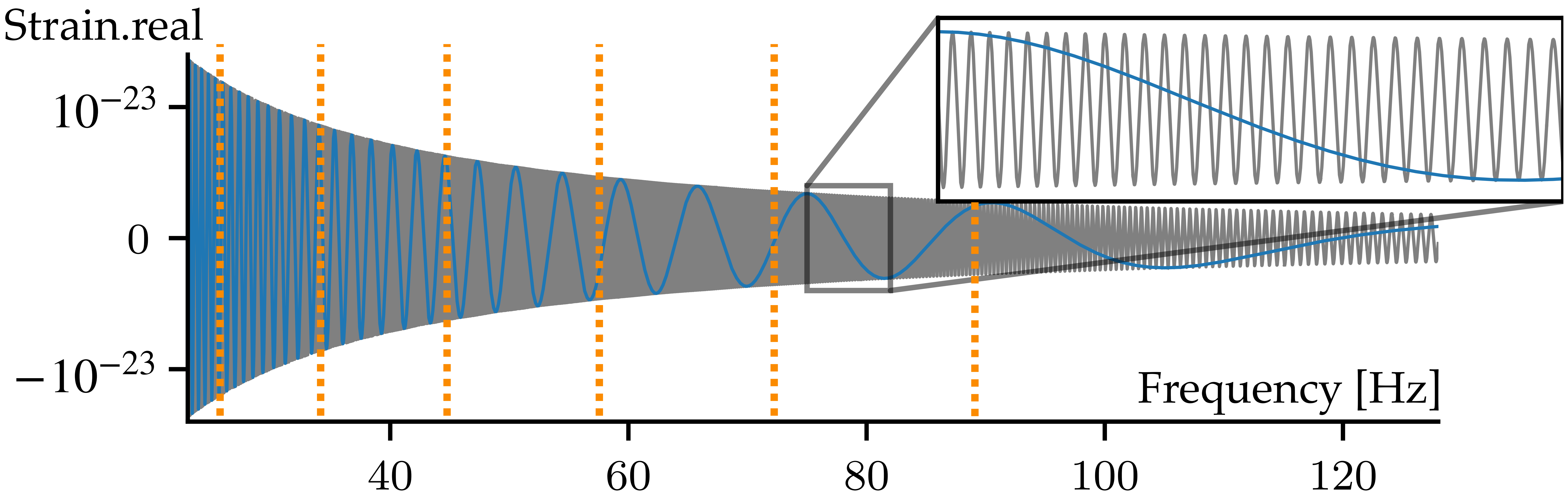
# BNS: Data compression

- **Challenge:** BNS signals are longer and more complex than BBH
- **Solution:** Prior conditioning enables **$M_c$-based compression**

1. **Heterodyning** (Cornish 2010) — factor out overall phase $\propto (M_c^{\text{est}} f)^{-5/3}$
2. **Multibanding** (Vinciguerra+, 2017) — use reduced resolution at higher $f$

Prior-conditioned network includes $M_c$ estimate $\rho = M_c^{\text{est}}$

$q(\theta \,|\, d, M_c^{\text{est}})$



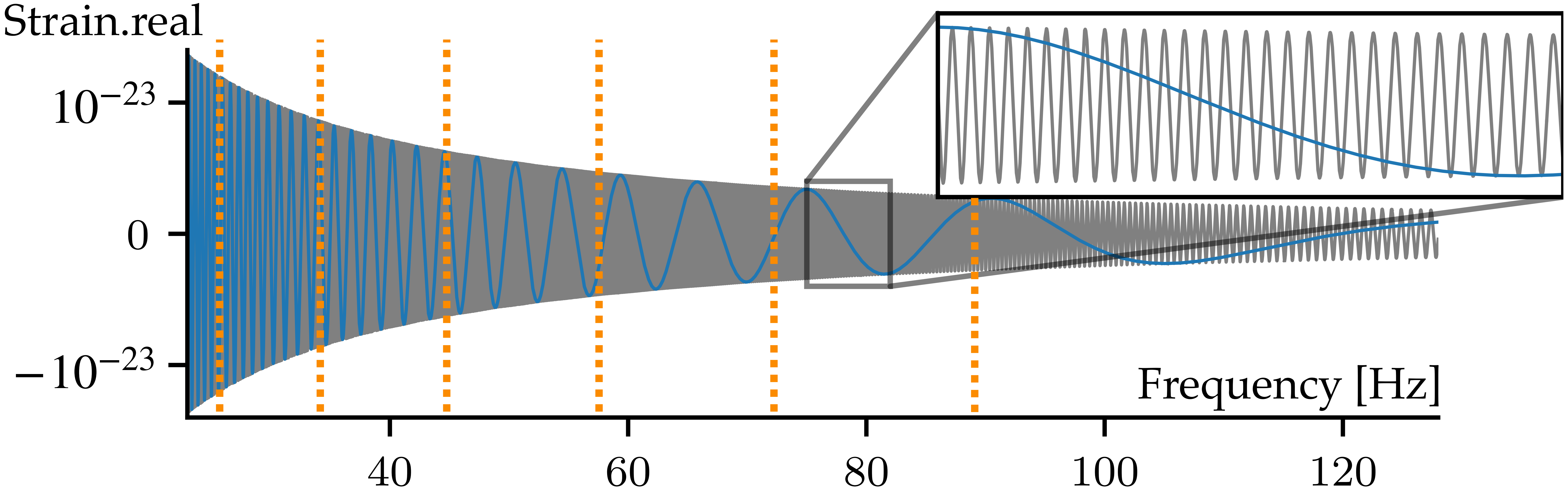
# BNS: Data compression

- **Challenge:** BNS signals are longer and more complex than BBH
- **Solution:** Prior conditioning enables **$M_c$-based compression**

1. **Heterodyning** (Cornish 2010) — factor out overall phase $\propto (M_c^{\text{est}} f)^{-5/3}$
2. **Multibanding** (Vinciguerra+, 2017) — use reduced resolution at higher $f$

Prior-conditioned network includes $M_c$ estimate $\rho = M_c^{\text{est}}$

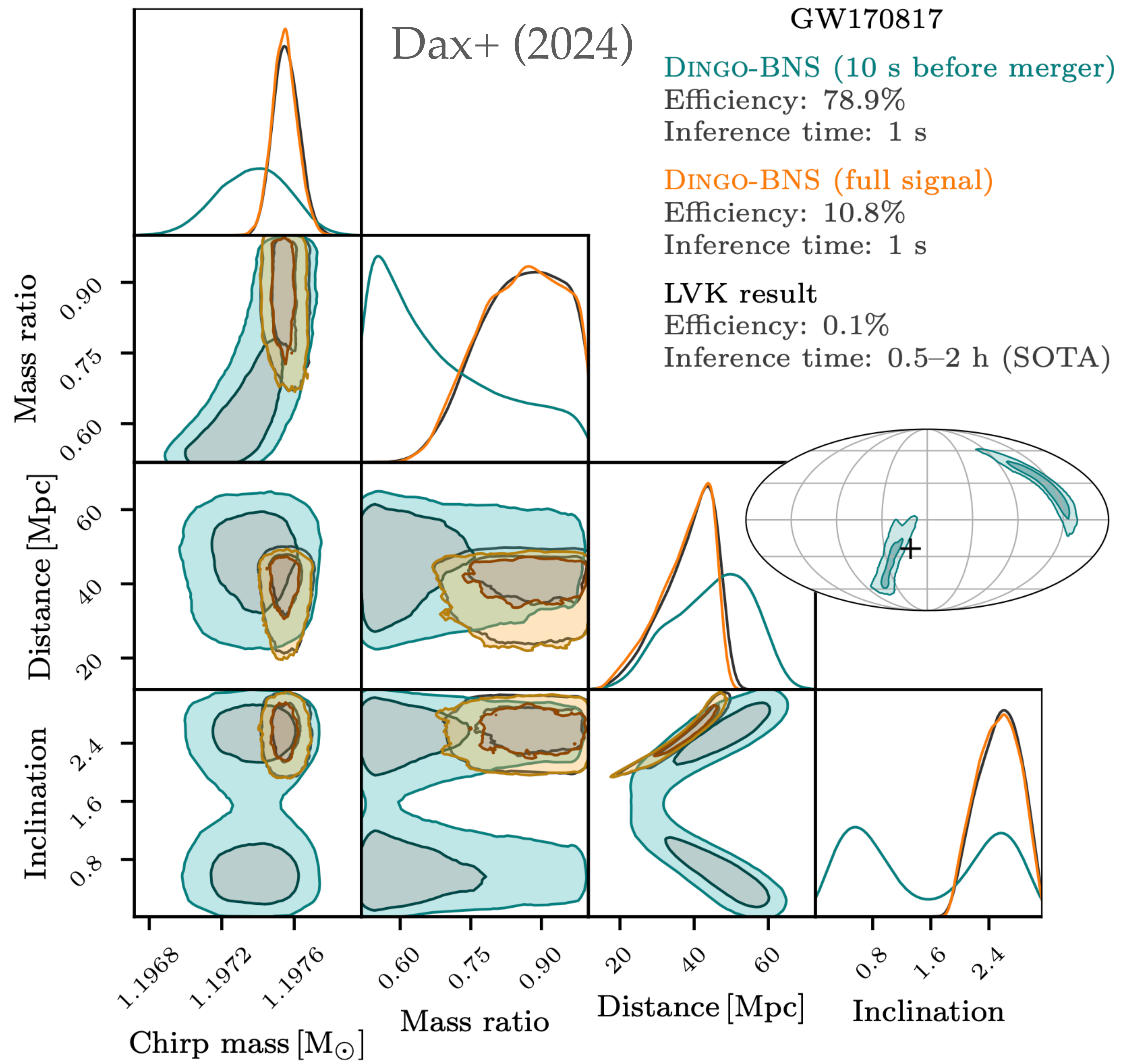
$q(\theta \,|\, d, M_c^{\text{est}})$




**⇒ Loss-free compression by 100x**

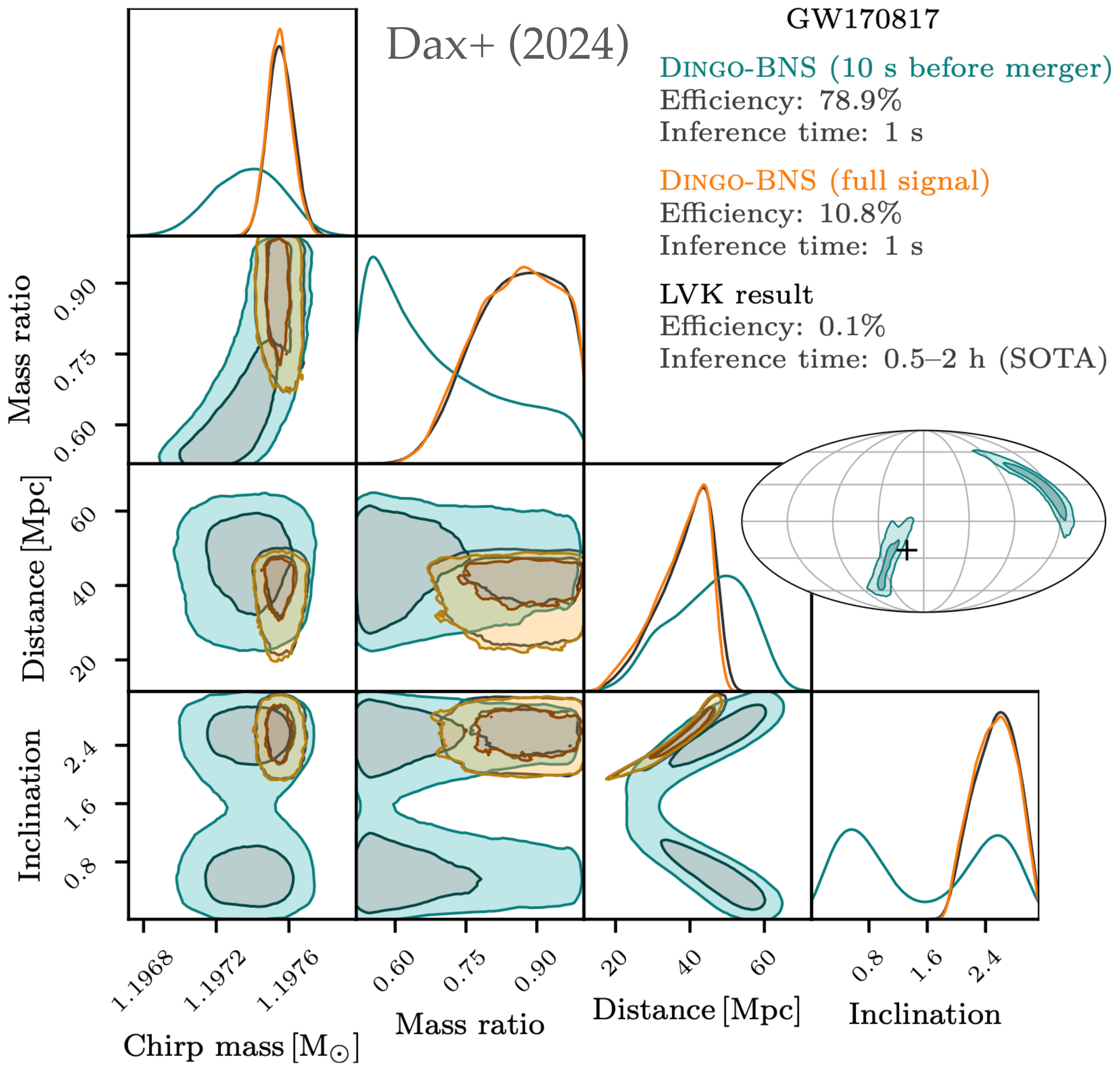
# BNS: Results

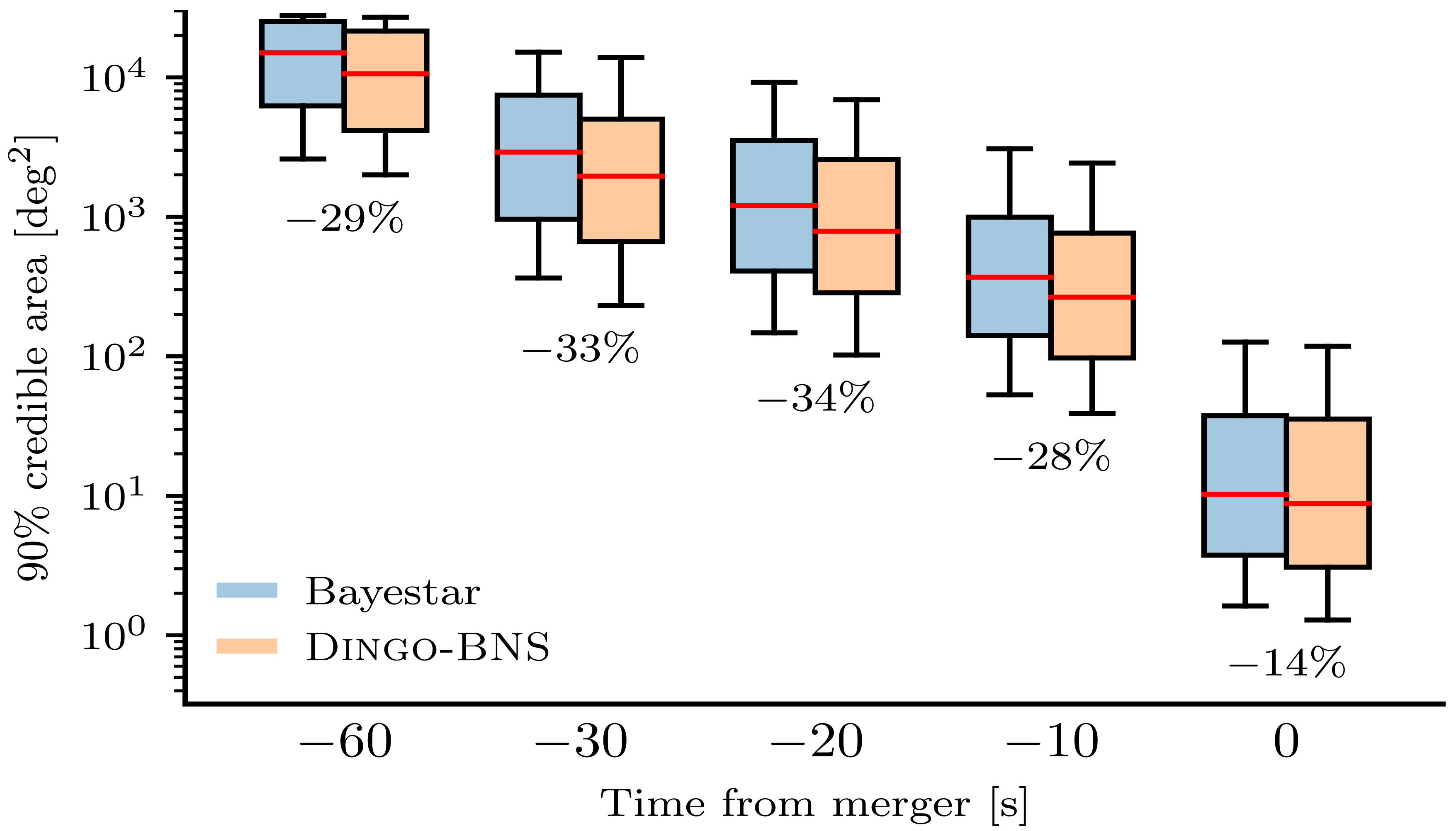
- DINGO-BNS reproduces public LVK results with only **1 second inference time**
- Inference at arbitrary times **before to the merger**



# BNS: Results

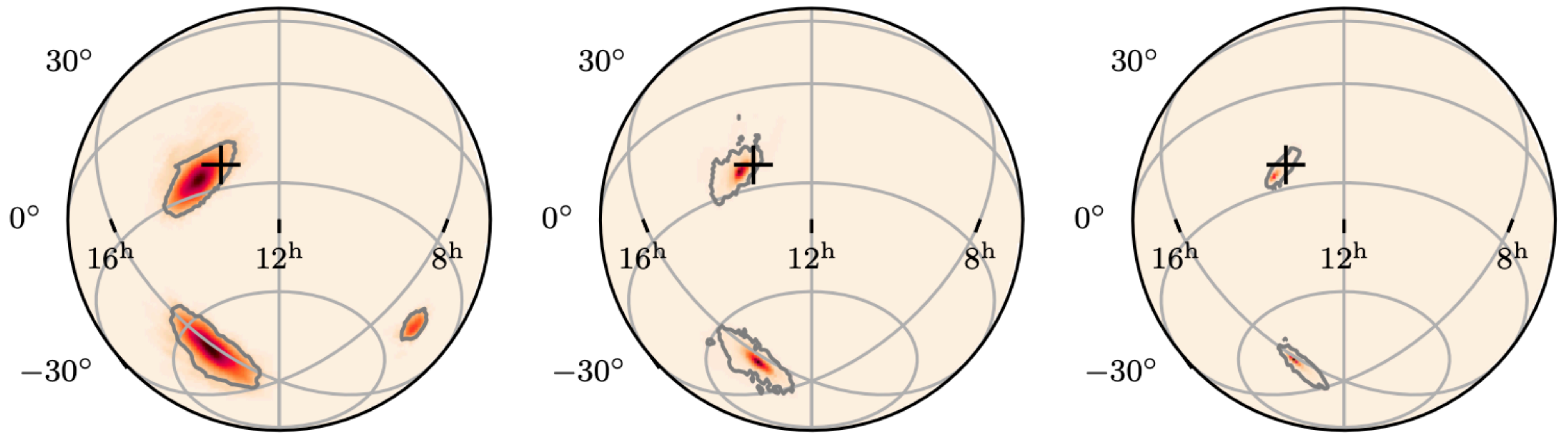
- DINGO-BNS reproduces public LVK results with only **1 second inference time**
- Inference at arbitrary times **before to the merger**
- Complete inference without approximations
  $\Rightarrow$ **30% improvement in low-latency localization**

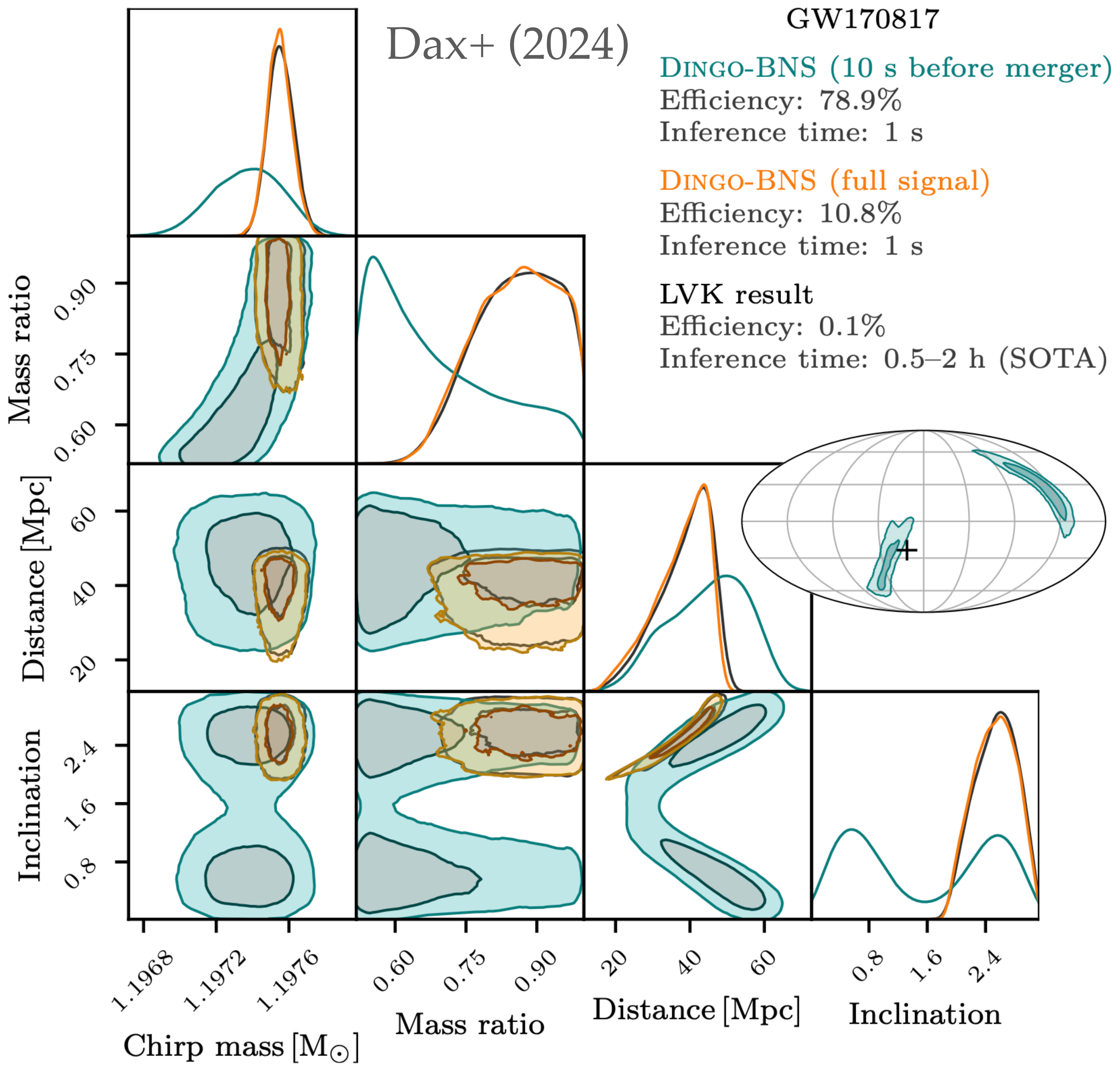


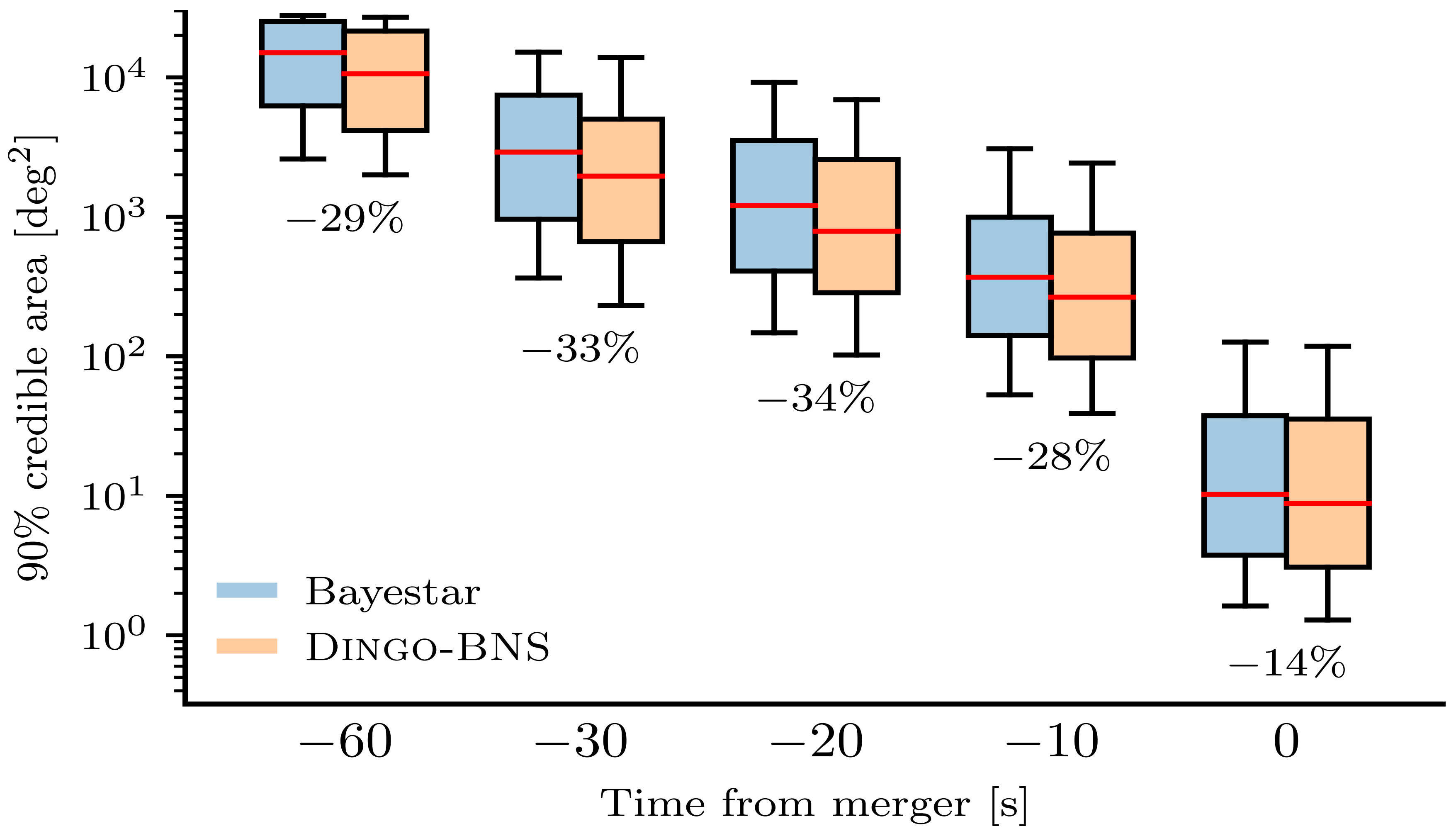



# BNS: Results

- DINGO-BNS reproduces public LVK results with only **1 second inference time**
- Inference at arbitrary times **before to the merger**
- Complete inference without approximations
  $\Rightarrow$ **30% improvement in low-latency localization**
- Scales to hour-long signals of **next-gen detectors**



# Summary

# Marginalizing vs. Conditioning

In some cases, there are additional non-inference parameters $\phi$ (related to likelihood or prior)
$\Rightarrow$ In these cases, need to impose some prior on $\phi$ either ***marginalize over*** or ***condition on*** $\phi$

# Marginalizing vs. Conditioning

In some cases, there are additional non-inference parameters $\phi$ (related to likelihood or prior)
$\Rightarrow$ In these cases, need to impose some prior on $\phi$ either ***marginalize over*** or ***condition on*** $\phi$

**Marginalization**

- $\phi$ dependence implicit at inference
- Inference result depends on $p(\phi)$ via correlations between $\theta$ and $\phi$

$$q(\theta \mid d)$$

# Marginalizing vs. Conditioning

In some cases, there are additional non-inference parameters $\phi$ (related to likelihood or prior)
$\Rightarrow$ In these cases, need to impose some prior on $\phi$ either ***marginalize over*** or ***condition on*** $\phi$

**Marginalization**

- $\phi$ dependence implicit at inference
- Inference result depends on $p(\phi)$ via correlations between $\theta$ and $\phi$

$q(\theta\,|\,d)$

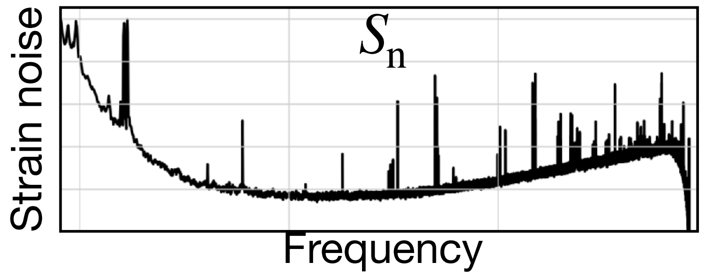
**Conditioning**

- $\phi$ dependence explicit: need to fix / sample $\phi_{\text{obs}} \in p(\phi)$ at inference
- Inference result (asymptotically) independent of $p(\phi)$
- Can apply loss-free (e.g., invertible) transformation $f_\phi$ to $d$
- Sometimes it makes sense to introduce artificial control parameters $\phi$

$q(\theta\,|\,d, \phi)$

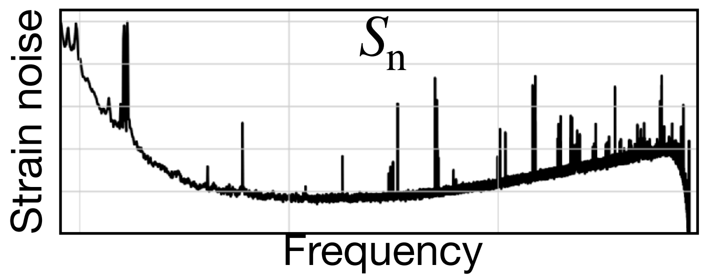
$q(\theta\,|\,f_\phi(d), \phi)$

# Conditioning for GW inference

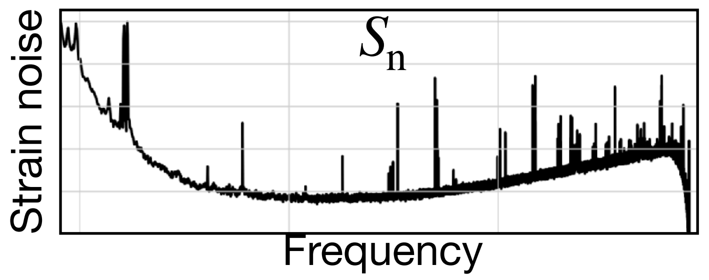


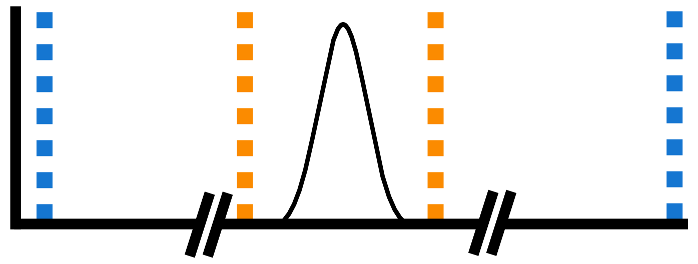

| Technique | Conditioning parameter(s) | Conditioning transformation | At inference, determined via | Purpose |
|---|---|---|---|---|
| Noise conditioning | $q(\theta \mid d, S_\mathrm{n})$ | $d \to d/S_\mathrm{n}$ | Signal free data | Tuning to variable detector noise level |

# Conditioning for GW inference

| | Technique | Conditioning parameter(s) | Conditioning transformation | At inference, determined via | Purpose |
|---|---|---|---|---|---|
|   | Noise conditioning | $q(\theta \mid d, S_n)$ | $d \rightarrow d / S_n$ | Signal free data | Tuning to variable detector noise level |
| $p(\theta \mid d) = p(g\theta \mid T_g d)$ | GNPE | $q(\theta \mid d, \hat{t})$ | $d \rightarrow d \cdot \exp(2\pi i f \hat{t})$ | Gibbs sampling | Data simplification |

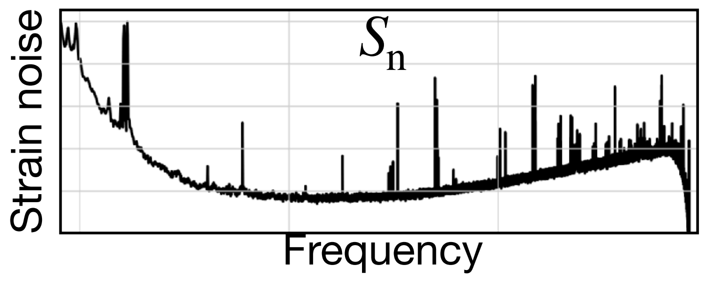
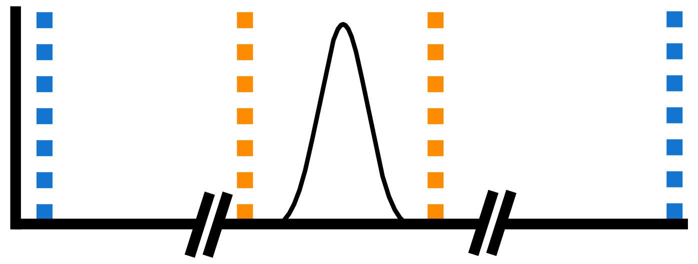
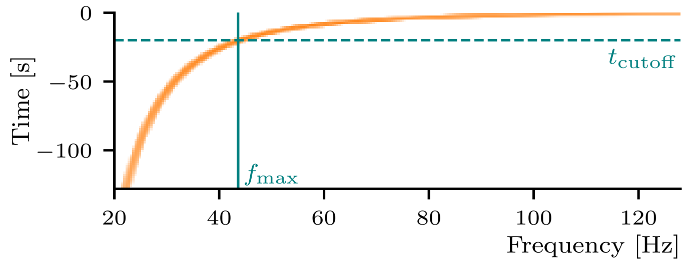
# Conditioning for GW inference



| | Technique | Conditioning parameter(s) | Conditioning transformation | At inference, determined via | Purpose |
|---|---|---|---|---|---|
| | Noise conditioning | $q(\theta \mid d, S_n)$ | $d \to d/S_n$ | Signal free data | Tuning to variable detector noise level |
| $p(\theta \mid d) = p(g\theta \mid T_g d)$ | GNPE | $q(\theta \mid d, \hat{t})$ | $d \to d \cdot \exp(2\pi i f \hat{t})$ | Gibbs sampling | Data simplification |
| | Prior-conditioning | $q(\theta \mid d, M_c^{est})$ | $d \to \overline{d \cdot \exp(i\varphi(M_c^{\text{est}}))}$ | GW search triggers | Data compression |

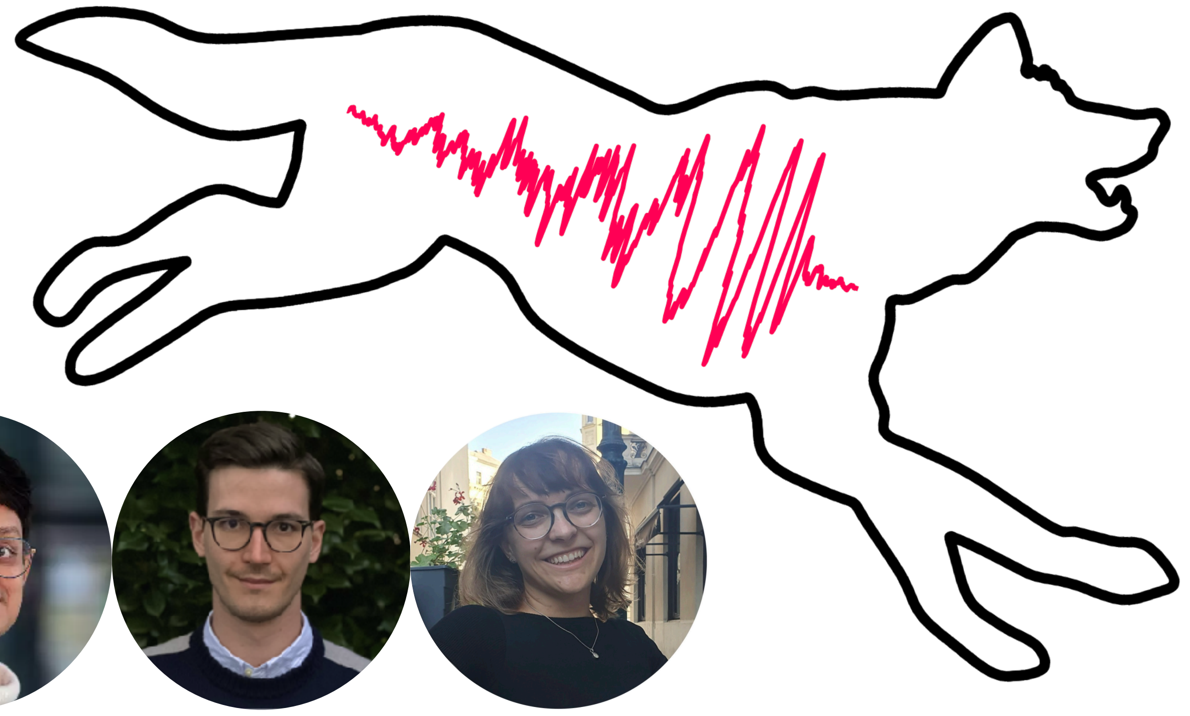
# Conditioning for GW inference

| | Technique | Conditioning parameter(s) | Conditioning transformation | At inference, determined via | Purpose |
|---|---|---|---|---|---|
|   | Noise conditioning | $q(\theta \mid d, S_\mathrm{n})$ | $d \to d / S_\mathrm{n}$ | Signal free data | Tuning to variable detector noise level |
| $p(\theta \mid d) = p(g\theta \mid T_g d)$ | GNPE | $q(\theta \mid d, \hat{t})$ | $d \to d \cdot \exp(2\pi i f \hat{t})$ | Gibbs sampling | Data simplification |
|  | Prior-conditioning | $q(\theta \mid d, M_c^{est})$ | $d \to \overline{d \cdot \exp(i\varphi(M_c^\mathrm{est}))}$ | GW search triggers | Data compression |
|   | Frequency masking | $q(\theta \mid d, f_\mathrm{min}, f_\mathrm{max})$ | $d \to d[i_\mathrm{min} : i_\mathrm{max}]$ | Pre-merger time | Inference with partial data |

# Conclusion

- **Lots of other great work** on SBI/ML for GWs!
- Simulation-based inference powerful paradigm for **fast and accurate GW inference**; after training, only the trained network is required for inference ("amortization")
- Reviewed for GW parameter estimation at LVK; github.com/dingo-gw/dingo
- **NPE-IS** provides a **generic framework to verify SBI** results (for tractable likelihoods)
- Science cases
  - *Binary black holes*: **enables** new, traditionally **expensive analyses** (e.g., 2404.14286)
  - *Binary neutron stars*: fast inference **enhances follow-up searches**
  - *Next-gen detectors*: Many open problems, ML most likely part of the solution
- GW science is a **great playground** to develop more general ML methods (NPE-IS, GNPE, prior-conditioning, Flow matching for SBI 2305.17161)

# Thanks for your attention!



S. Green J. Gair N. Gupte J. Wildberger A. Kofler

S. Buchholz M. Pürrer J. Macke A. Buonanno B. Schölkopf

## References


- **NPE for binary black holes**
  Dax+, *Real-Time Gravitational Wave Science with Neural Posterior Estimation*, Phys. Rev. Lett. 127, 241103 (2021)
- **Symmetries with NPE**
  Dax+, *Group equivariant neural posterior estimation*, ICLR 2022
- **Importance-sampled NPE**
  Dax+, *Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference*, Phys.Rev.Lett. 130, 171403 (2023)
- **NPE for binary neutron stars**
  Dax+, *Real-time gravitational-wave inference for binary neutron stars using machine learning*, 2024