
Lecture Recording

- ❖ **Note: These lectures will be recorded and posted onto the IMPRS website**
- ❖ Dear participants,
- ❖ We will record all lectures on “*Making sense of data: introduction to statistics for gravitational wave astronomy*”, including possible Q&A after the presentation, and we will make the recordings publicly available on the IMPRS lecture website at:
 - <https://imprs-gw-lectures.aei.mpg.de/2021-making-sense-of-data/>
- ❖ By participating in this Zoom meeting, you are giving your explicit consent to the recording of the lecture and the publication of the recording on the course website.

Software for practical session

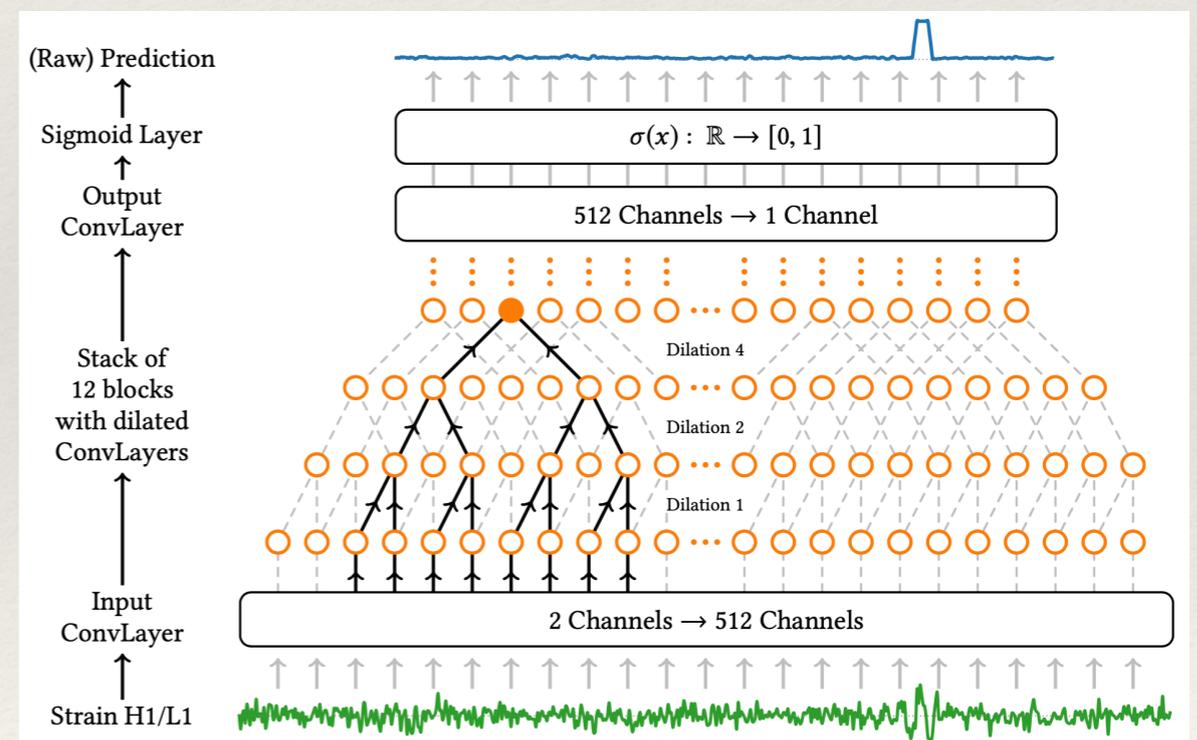
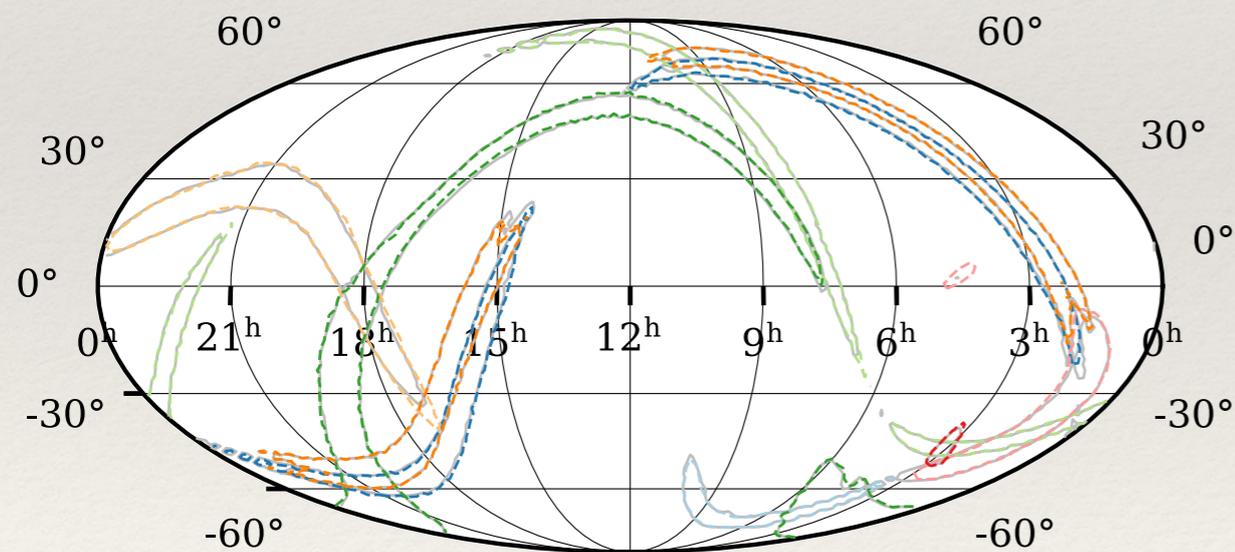
- ❖ Tomorrow's class will be a practical session. Please update the conda installation for the previous practicals to include:
 - pytorch
 - lalsimulation
 - corner (for plotting)

Making sense of data: introduction to statistics for gravitational-wave astronomy

Lecture 11: Applications of Machine Learning to Gravitational Waves

AEI IMPRS Lecture Course

Stephen Green stephen.green@aei.mpg.de



Introduction

❖ Many **applications** so far of machine learning to gravitational waves:

- searches, parameter estimation, waveform modeling, glitch classification, population inference, noise reduction, ...

(somewhat biased presentation)

❖ **Objectives** of these applications:

- Usually **speed of calculations**
- Increasingly (and I believe in the future) **previously-intractable analyses**

Searches

- ❖ George and Huerta (2017), Gabbard et al (2018), Gebhard et al (2019), ...
- ❖ Approach is to train a **classifier** to distinguish **signal + noise** vs **noise only**.
- ❖ **Dataset:**
 - **time domain** data, $T = 1$ s, $f_s = 8192$ Hz
 - 5×10^5 samples; half with signal, half without; **whitened**
 - IMRPhenomD, $5 M_\odot \leq m_{1,2} \leq 95 M_\odot$, zero spin, 0.65 s $\leq t_c \leq 0.85$ s
 - **Probability model:** 2 softmax outputs representing $p(y = i | \mathbf{x})$
 - **Loss:** Binary cross-entropy $f(\theta) = -\sum_{i \in S} \log(\theta_i^S) - \sum_{i \in N} \log(\theta_i^N)$

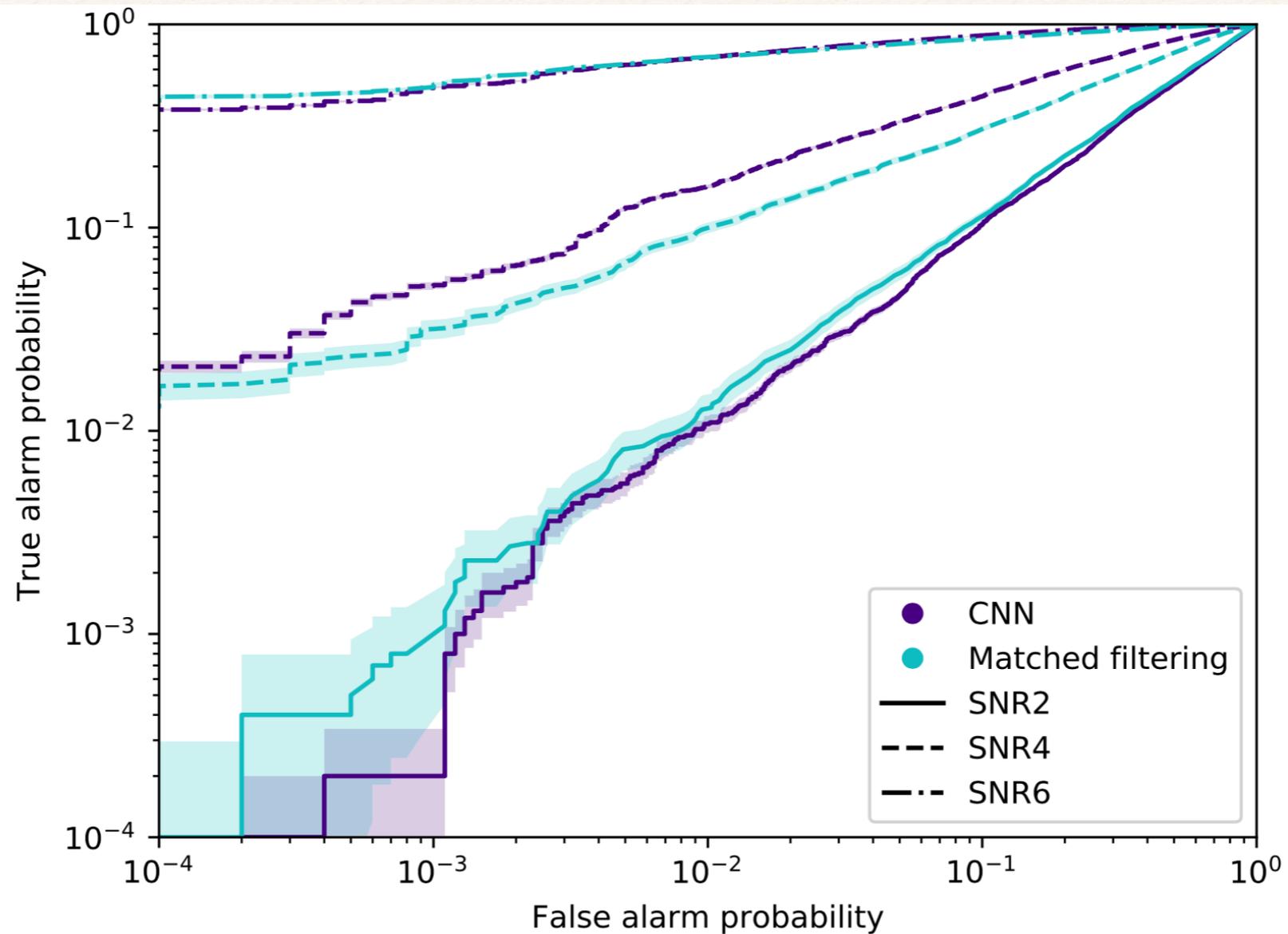
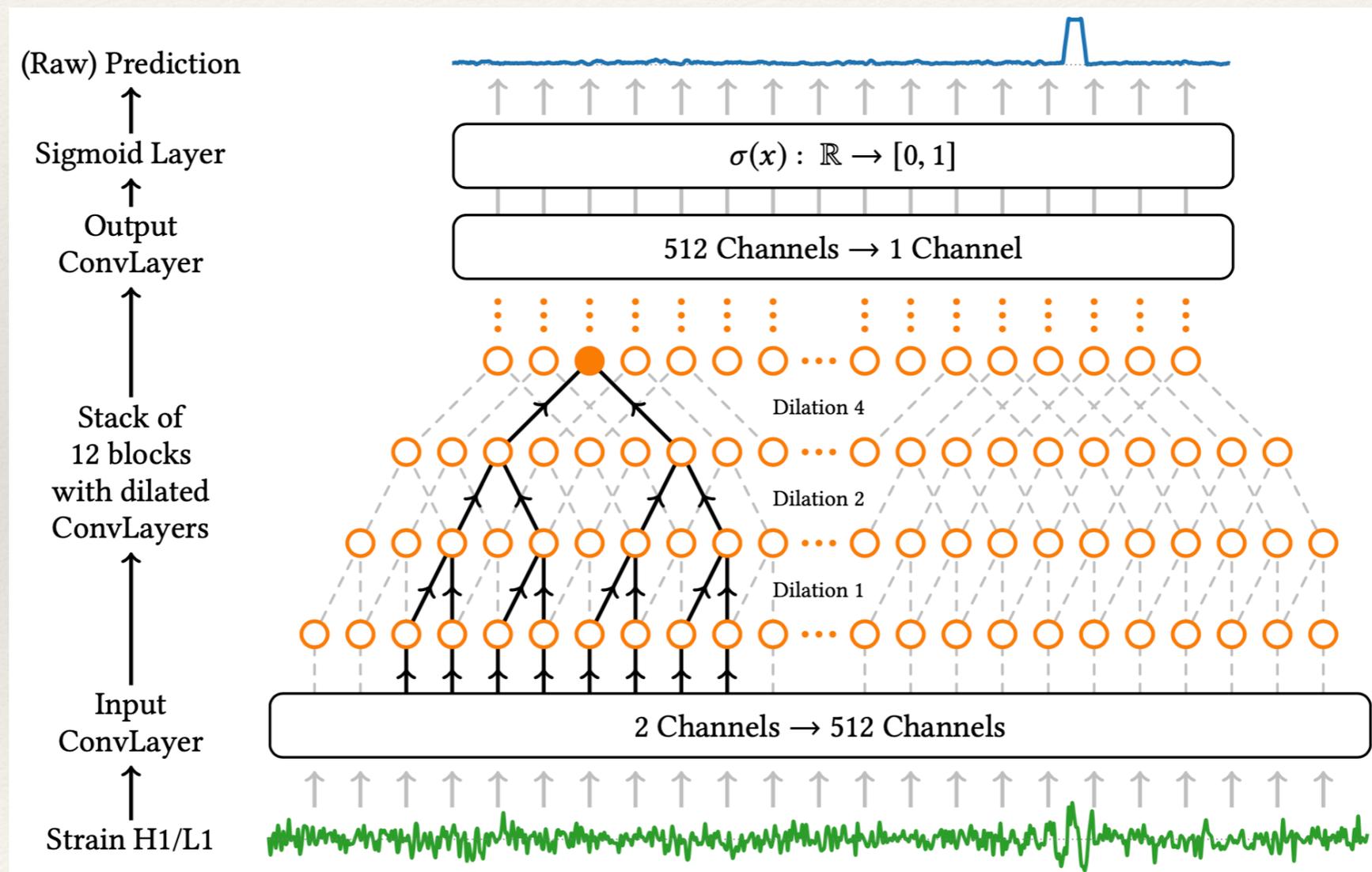


FIG. 2. The ROC curves for test data sets containing signals with optimal SNR, $\rho_{\text{opt}} = 2, 4, 6$. We plot the true alarm probability versus the false alarm probability estimated from the output of the CNN (purple) and matched-filtering (cyan) approaches. Uncertainties in the true alarm probability correspond to $1-\sigma$ bounds assuming a binomial distribution.

Other search approaches

- ❖ Gebhard et al (2019) used **dilated convolutions** for longer data sets.



Other search approaches

- ❖ Jadhav et al (2021) used **transfer learning** with InceptionV3 network

AC	93.2	0.0	0.0	0.0	0.0	0.0	2.3	0.0	0.0	0.0	4.5	0.0	0.0	0.0	0.0	0.0	0.0
BL	0.0	87.3	0.0	0.0	0.5	0.0	1.6	0.0	0.0	0.0	0.0	7.9	0.0	0.0	0.0	0.0	2.6
EL	0.0	0.0	98.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.0	0.0	0.0	0.0
GN	0.0	0.0	0.0	95.3	0.0	4.6	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HX	0.0	1.2	0.0	0.0	98.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CBC	0.0	0.0	0.0	1.9	0.0	98.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LM	0.0	0.7	0.0	0.0	0.0	0.0	97.8	0.7	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0
LFB	0.0	0.0	0.0	0.0	1.6	0.0	1.6	91.9	3.2	0.0	0.0	0.0	0.0	0.0	1.6	0.0	0.0
LFL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.9	91.4	0.0	1.4	0.0	1.4	0.0	0.0	0.0	2.9
PL1	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.1	0.9	0.0	0.0	0.0	0.0	0.0	0.0
PL2	5.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.7	91.9	0.0	0.0	0.0	0.0	0.0	0.0
RBL	0.0	2.8	0.0	0.0	0.0	0.0	1.4	0.0	0.0	0.0	0.0	94.3	0.0	0.0	0.0	0.0	1.4
SL	1.8	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.3	0.0	0.0	0.0	0.0
SC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.6	0.0	98.8	0.0	0.0	0.0
TM	0.6	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	1.3	0.0	0.0	97.4	0.0	0.0
WL	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	80.0	0.0
WH	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.9	0.0	1.0	0.0	0.0	95.2
	AC	BL	EL	GN	HX	CBC	LM	LFB	LFL	PL1	PL2	RBL	SL	SC	TM	WL	WH

Predicted Class

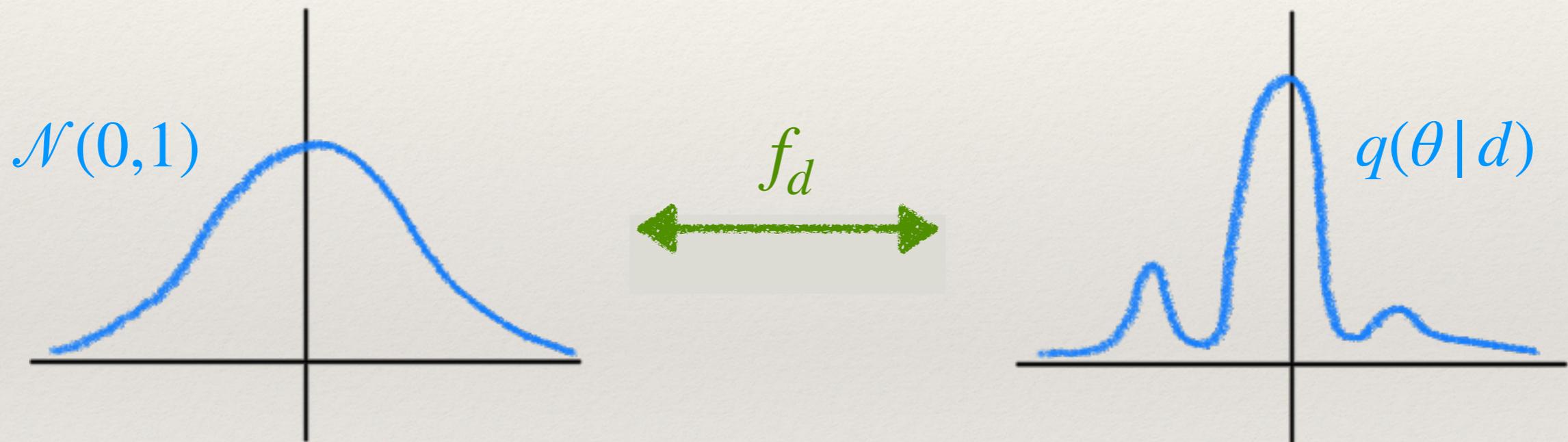
Parameter estimation

- ❖ George and Huerta (2018): point estimate
- ❖ Gabbard et al (2019): variational autoencoder
- ❖ Chua and Vallisneri (2020): Gaussian with learned covariance, histogram
- ❖ **Dax et al (2021): normalizing flow**

 method to build very
complicated distributions

Normalizing flow

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



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

Normalizing flow

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

❖ Requirements:

1. Invertible
2. Simple Jacobian determinant

❖ Parametrize f_d using **neural network**.

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

needed for computing
cross-entropy loss

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)

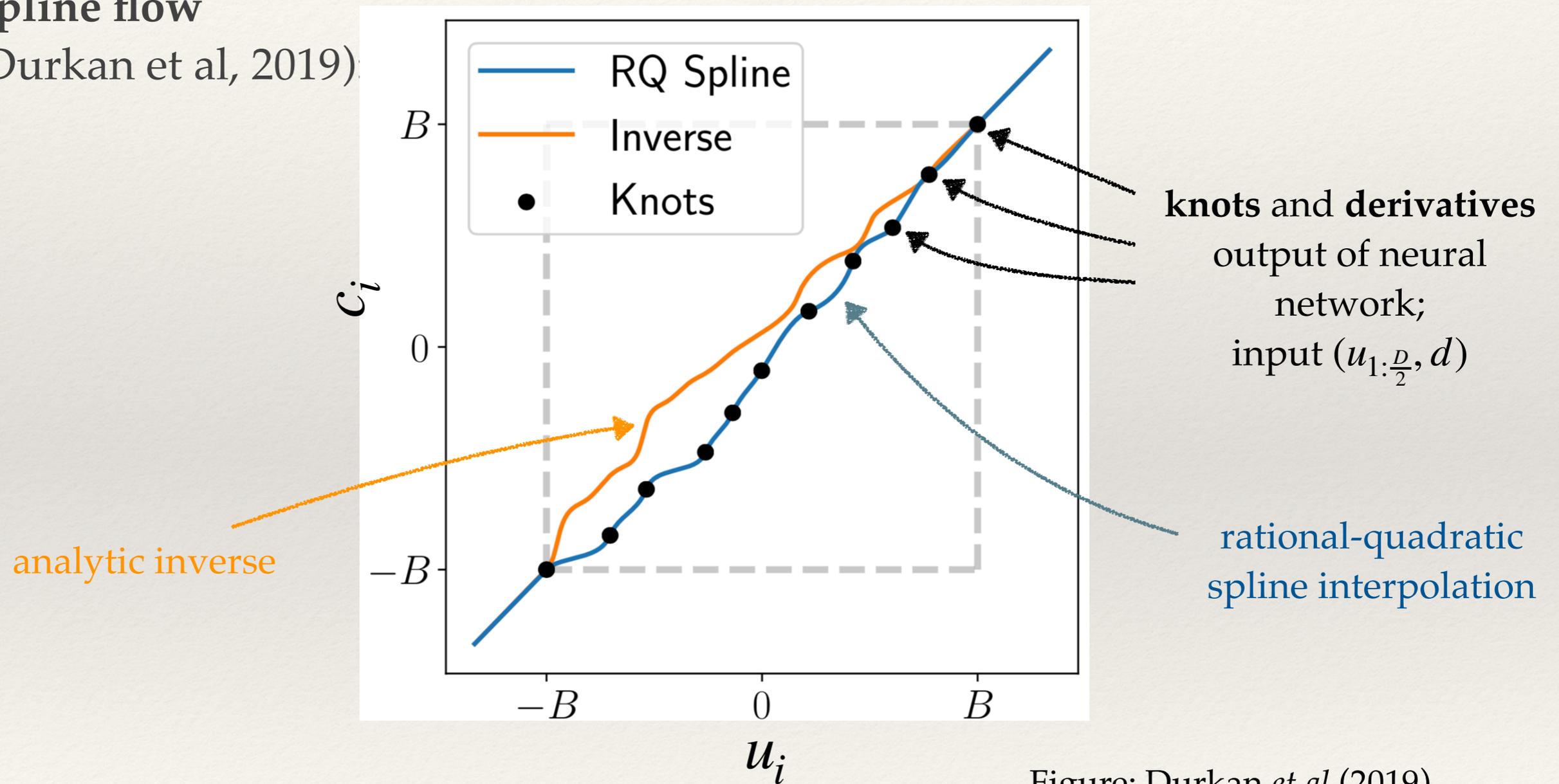


Figure: Durkan *et al* (2019)

Normalizing flow

- ❖ Sequence of flows can give very complicated distribution

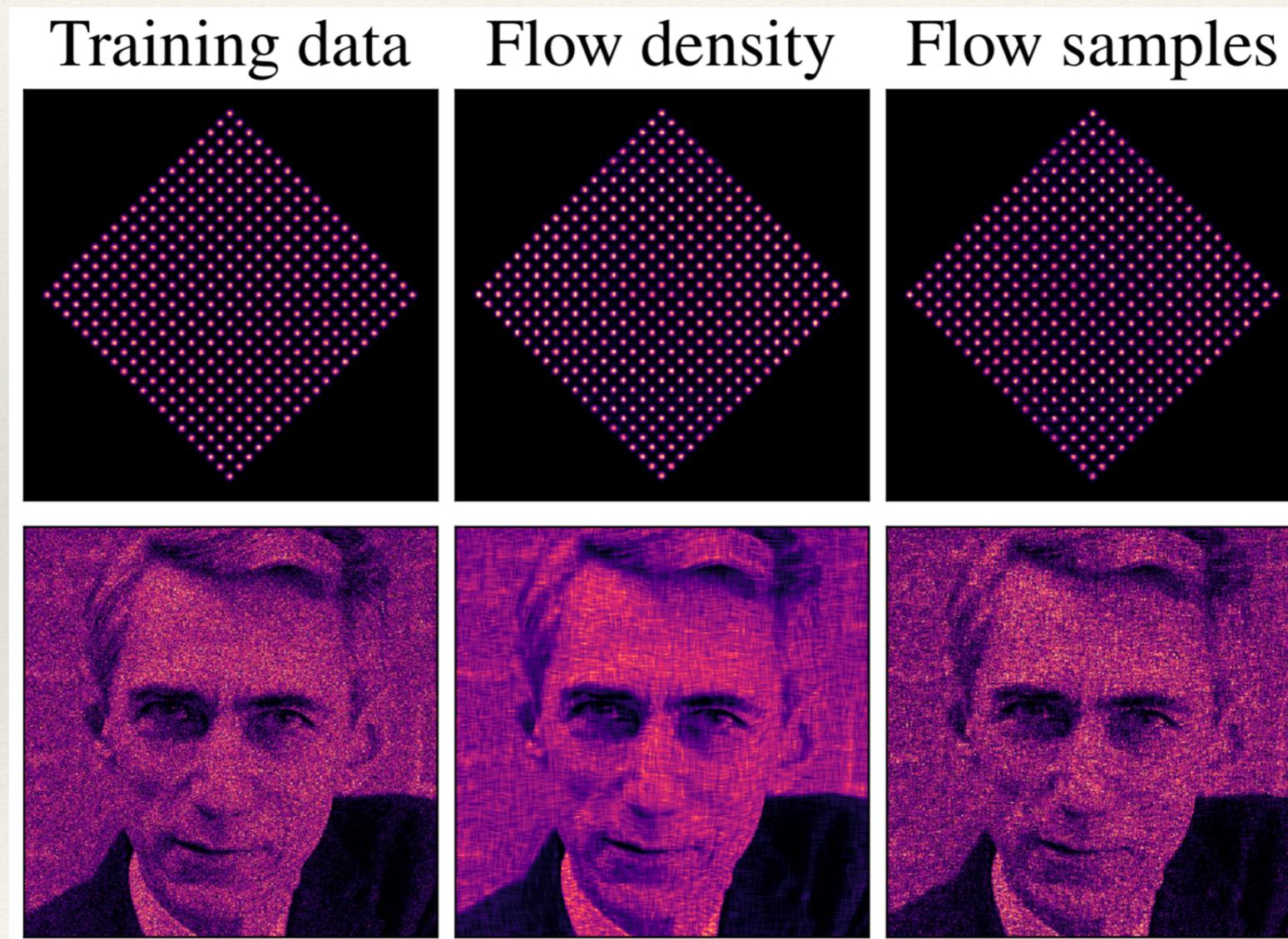
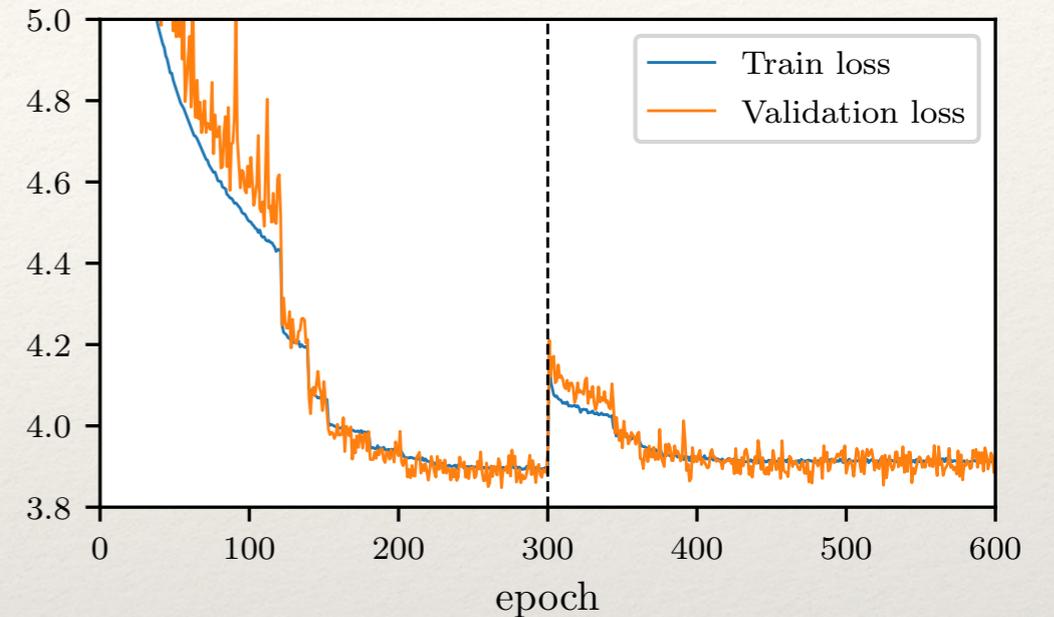


Image: Durkan *et al* (2019)

Training

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

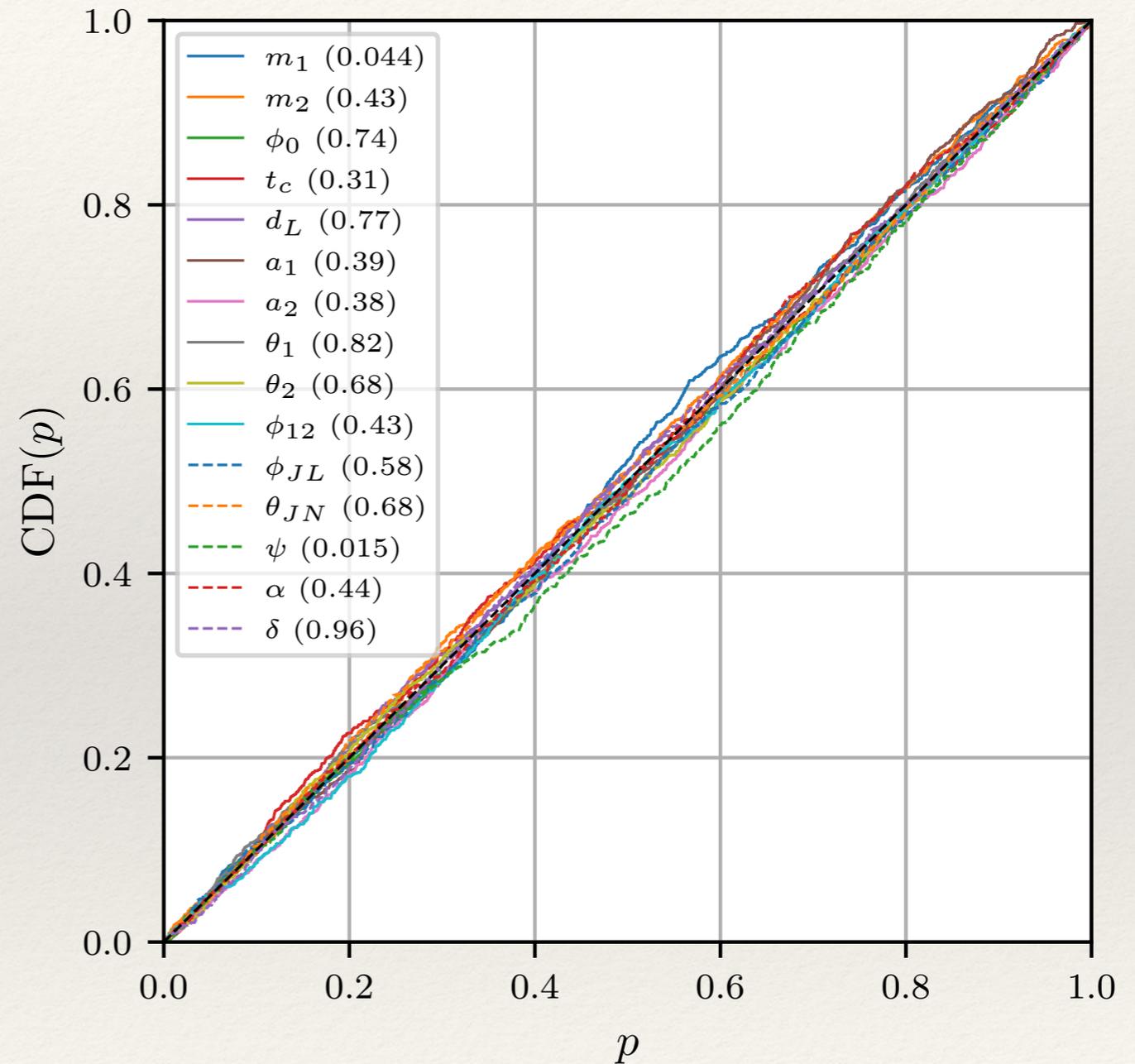


- ❖ + stationary Gaussian noise realizations
- ❖ Train several neural networks based on different noise level / number of detectors / distance range:

Observing run	Detectors	Distance range [Mpc]
O1	HL	[100, 2000]
O2	HL	[100, 2000]
	HLV	[100, 6000]
		[100, 1000]

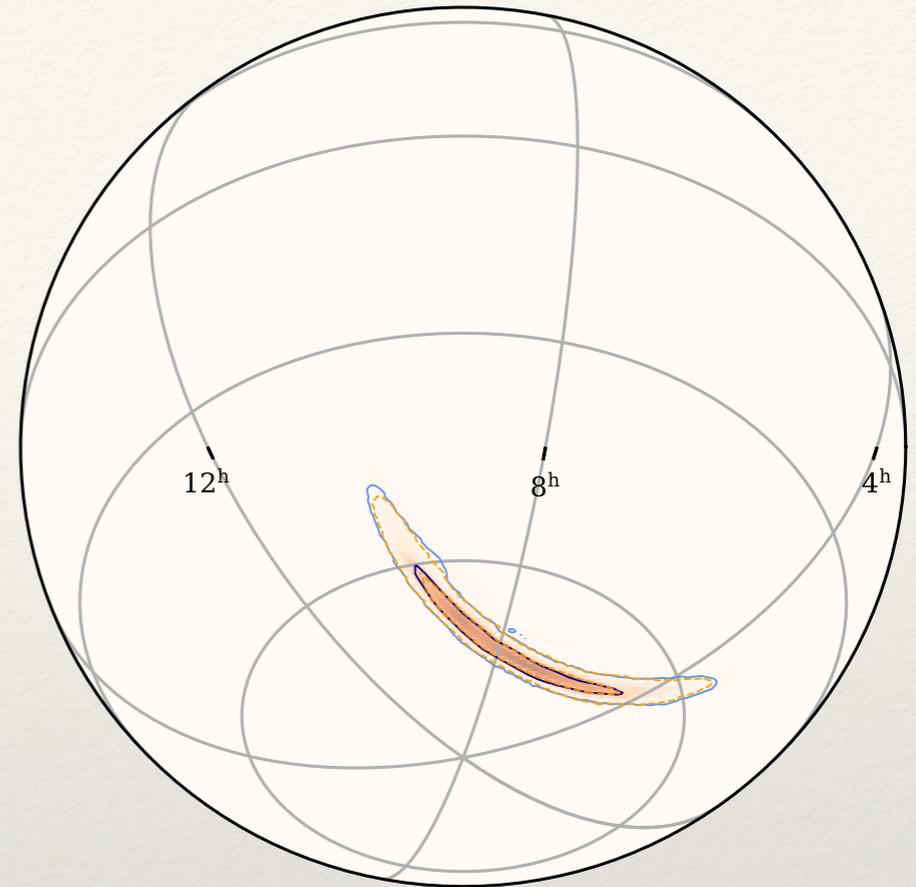
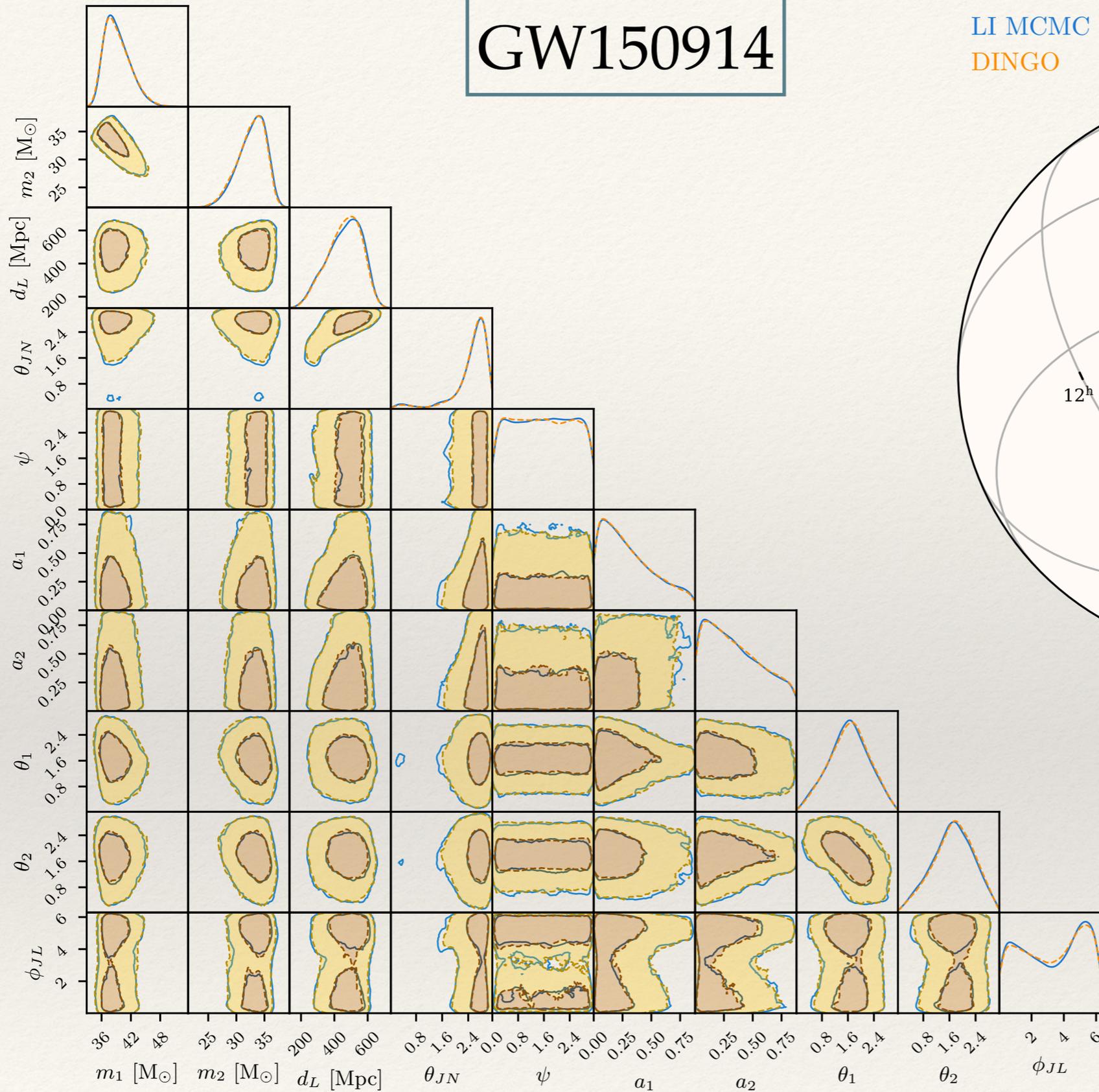
Method validation

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



GW150914

LI MCMC
DINGO

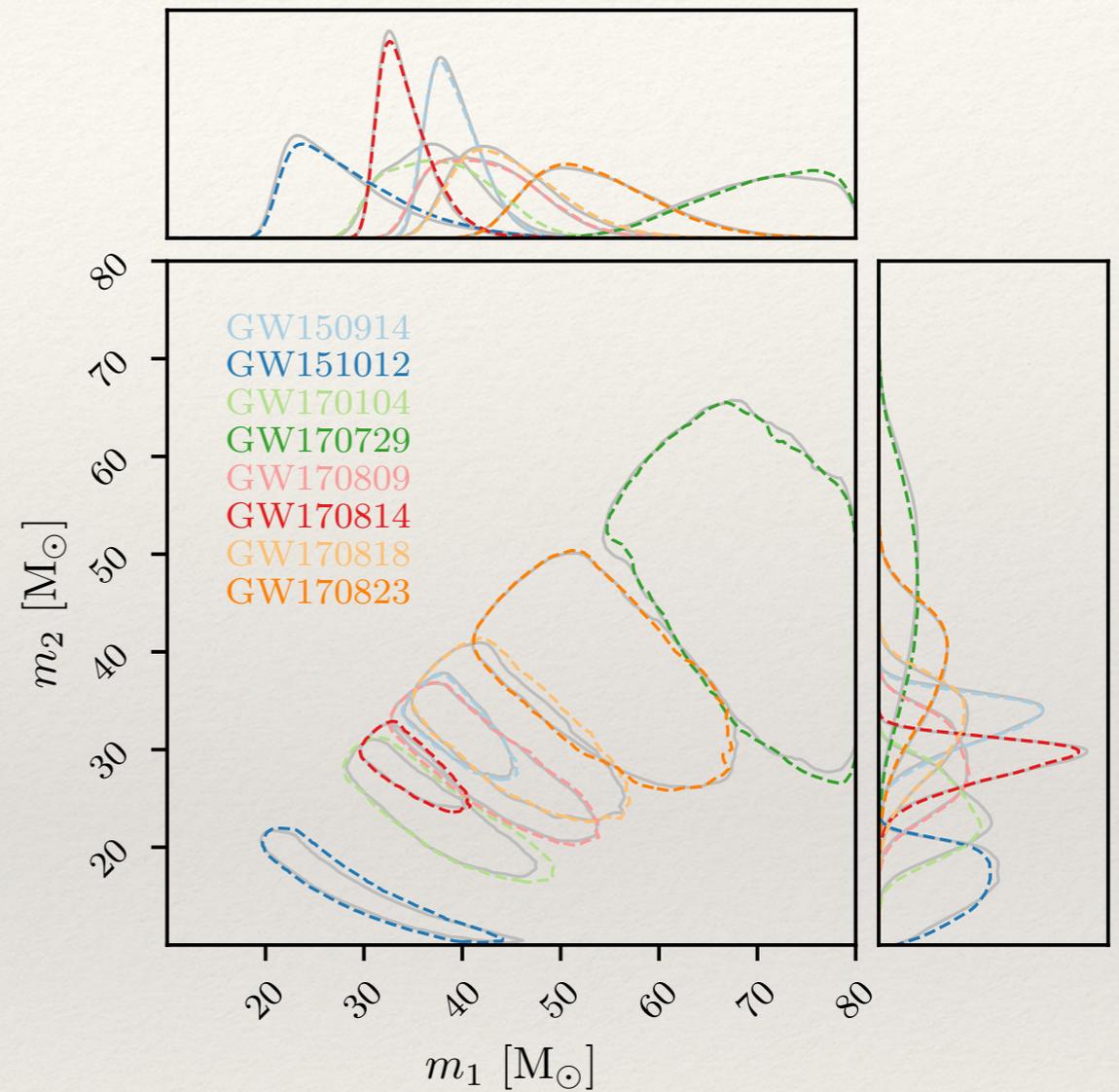
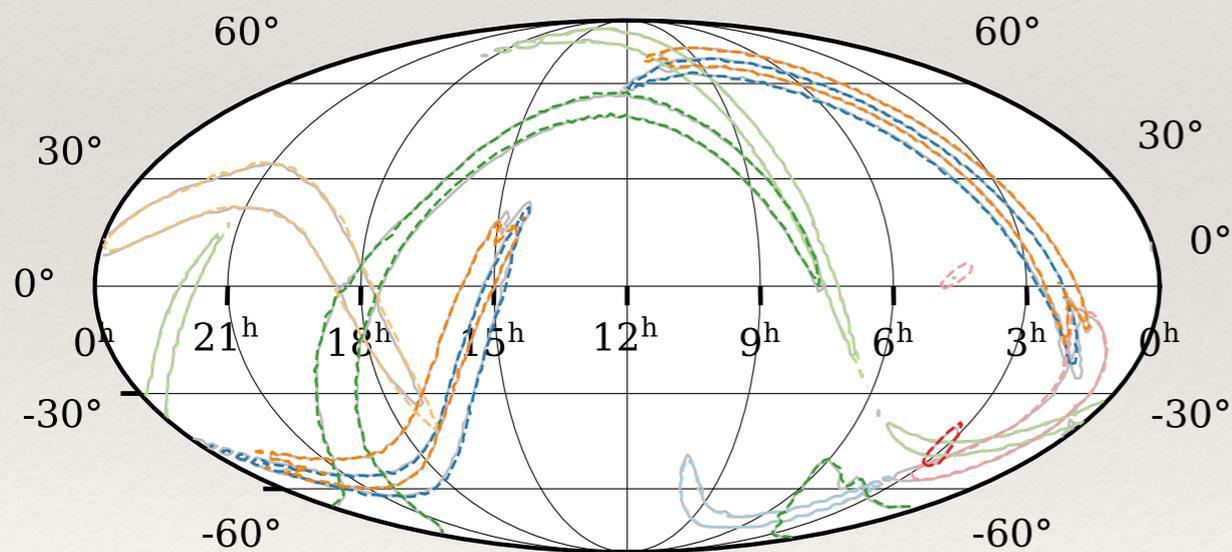


- ❖ This **generates the posterior**, matching MCMC
- ❖ 50,000 samples in ~ 20 s

Amortized inference

- Account for detector nonstationarity from event to event by **conditioning on noise PSD**

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



GWTC-1

Simulation-based inference

- ❖ These machine-learning methods are all examples of **simulation-based inference**, which simply means that training uses simulated data (noisy waveforms).
 - Standard inference methods (e.g., MCMC) are **likelihood-based**.
 - Simulation-based inference is applicable in situations where likelihoods are unavailable or too expensive.
 - Because of this, machine learning can carry out analyses that are not possible using standard tools. E.g., non-Gaussian detector noise.

Waveform modeling

- ❖ **Chua, Galley, Vallisneri (2019):** Reduced Order Modeling with Artificial Neurons (ROMAN)
- ❖ Expand waveforms in reduced basis ($\sim 10^2$ elements) and interpolate with neural network

$$h(\theta) = \sum_i \langle h(\theta) | e_i \rangle e_i := \sum_i \alpha_i(\theta) e_i \equiv \alpha(\theta)$$

- ❖ **Training data:** 6×10^5 pairs $\{\theta_n, \alpha(\theta_n)\}$

- ❖ **Loss function:**

$$L := \frac{\langle |\alpha - \hat{\alpha}|^2 \rangle}{\sqrt{\langle |\hat{\alpha}|^2 \rangle}}$$

Engineered to give more weight
to later basis elements

- ❖ **Fully-connected network:** 25 hidden layers x 256 units

Waveform modeling

- ❖ **Initial work:** TaylorF2; four parameters $\theta = (m_1, m_2, \chi_1, \chi_2)$

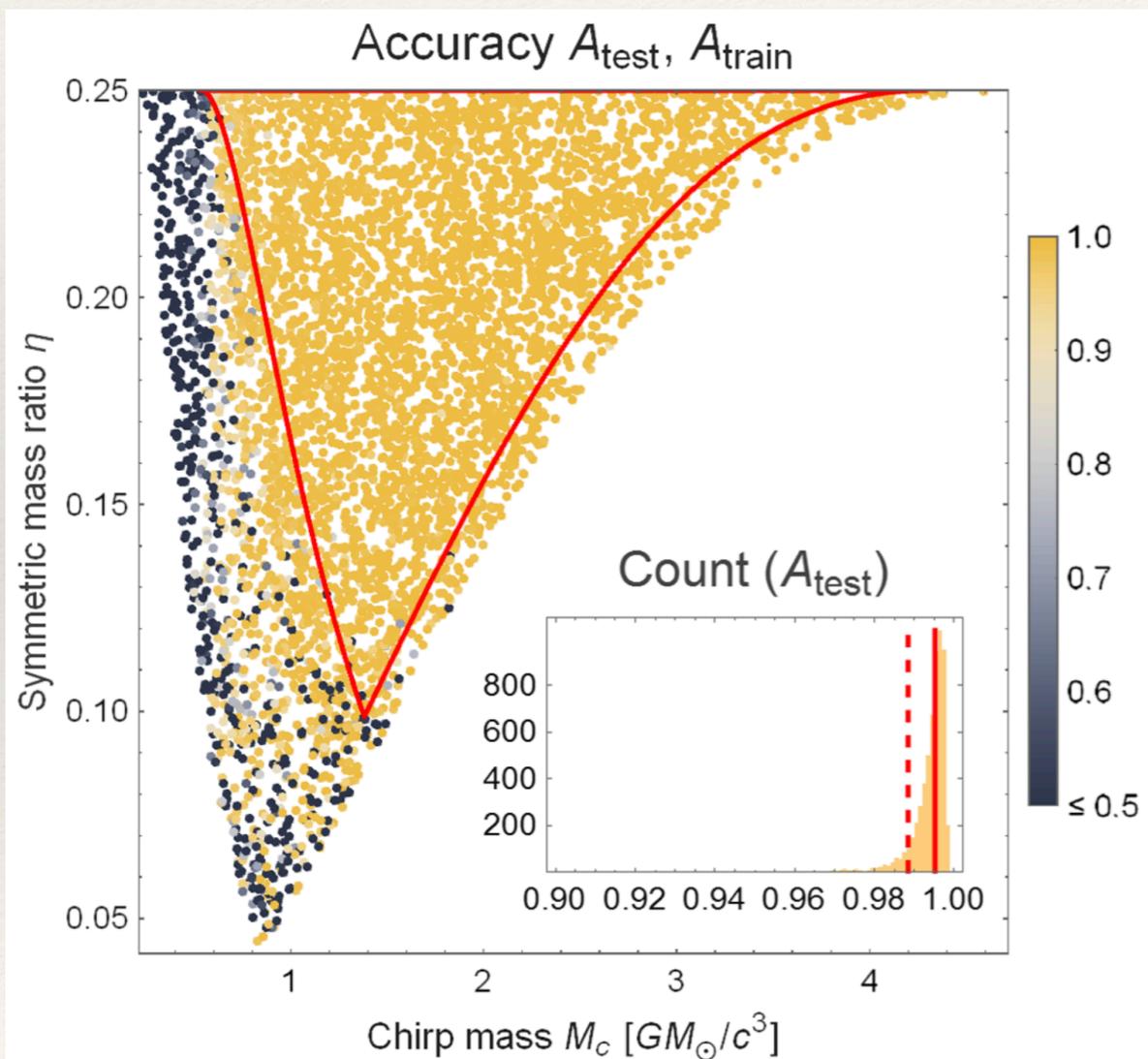
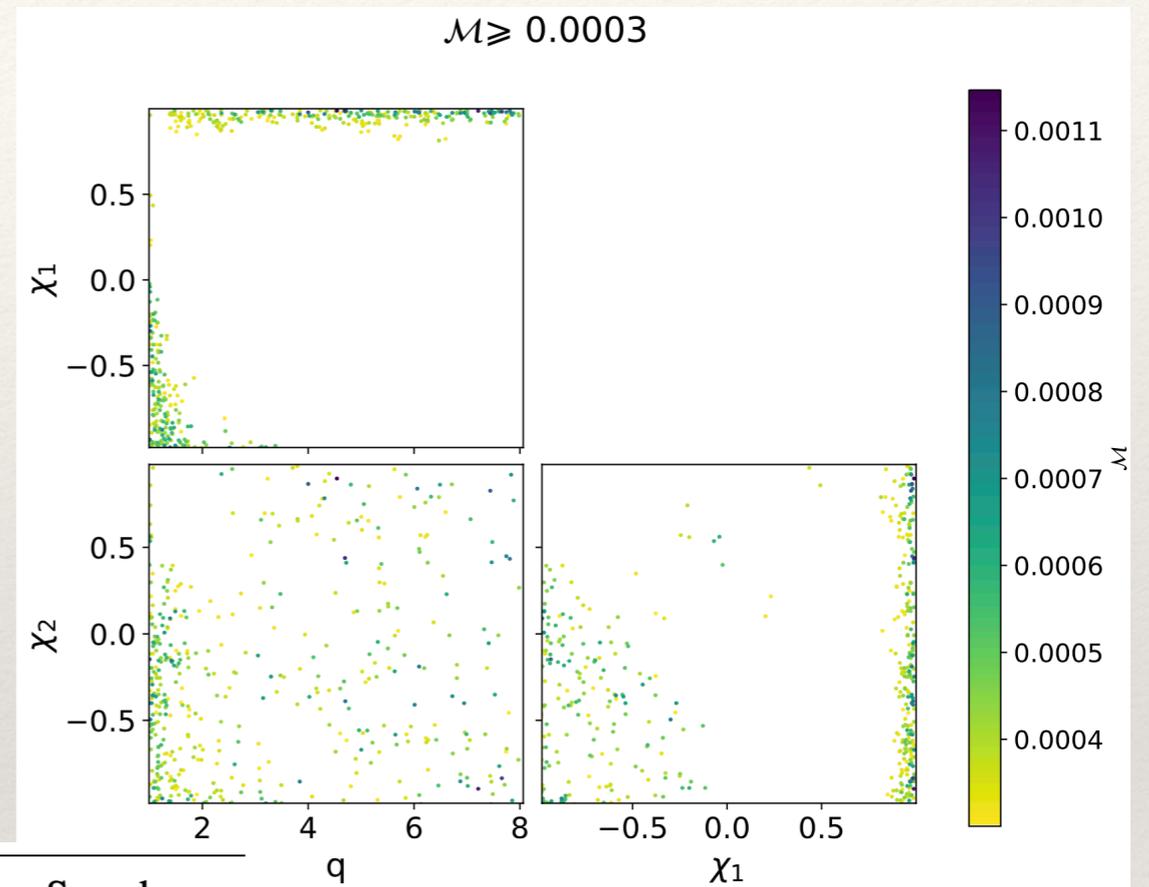


FIG. 2. Top: Plot of accuracy A as a function of (M_c, η) for test set (inside red border) and for 3000 training examples with (M_c, η) outside the domain of interest. Inset: Histogram of test-set accuracy values with tenth percentile (dashed line) and median (solid line) indicated. Bottom: Visualization of typical

- ❖ Automatic differentiation enables gradient-based sampling methods (e.g., Hamiltonian Monte Carlo)

Waveform modeling

- ❖ More recent work (S. Khan and R. Green, 2021) uses similar techniques applied to SEOBNRv4.
 - ❖ Fit to amplitude and phase



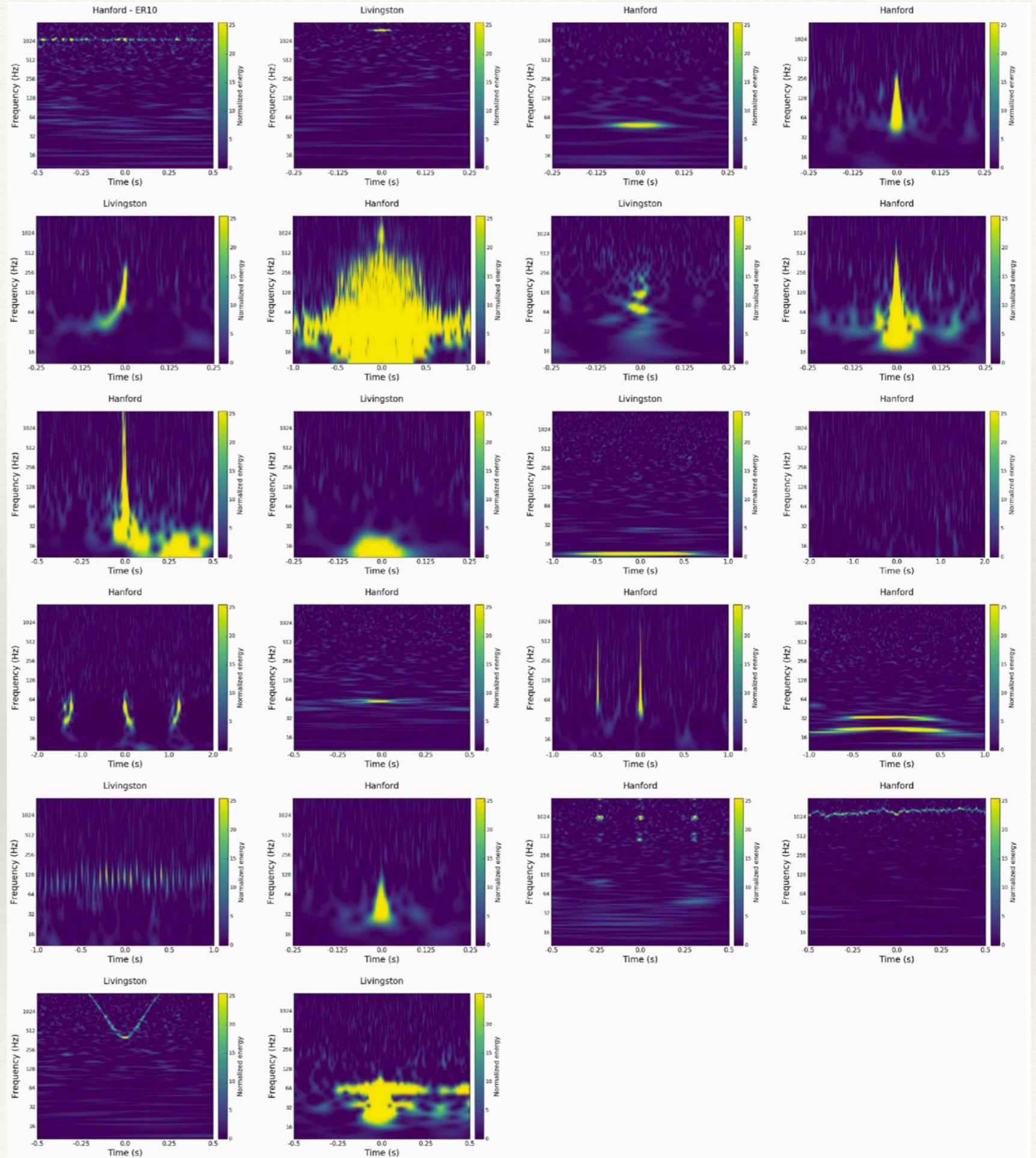
	CPU		GPU		Speed-up (CPU/GPU)
	Total Time (ms)	Time Per Waveform (ms)	Total Time (ms)	Time Per Waveform (ms)	
Single	2.7	2.7	0.4	0.4	7
Batched (10)	13	1.3	0.5	0.05	26
Batched (10^2)	73.3	0.73	2.1	0.021	35
Batched (10^3)	575.4	0.58	16.98	0.017	34
Batched (10^4)	5010	0.50	163.4	0.016	31

Machine learning for Gravity Spy: Glitch classification and dataset

S. Bahaadini^{a,*}, V. Noroozi^b, N. Rohani^a, S. Coughlin^{c,d}, M. Zevin^{c,d}, J.R. Smith^e, V. Kalogera^{c,d}, A. Katsaggelos^a

Class	Total	# train set	# valid set	# test set	Duration	Frequency	Evolving
1080Lines	328	230	49	49	Long	High	No
1400Ripples	232	162	35	35	Short	High	No
Air Compressor	58	41	8	9	Short	Low	No
Blip	1869	1308	281	280	Short	Mid	Yes
Chirp	66	46	10	10	Short	Mid, Low	Yes
Extremely Loud	454	318	68	68	Long	High, Mid, Low	Yes
Helix	279	195	42	42	Short	Mid	Yes
Koi Fish	830	581	125	124	Short	Mid, Low	Yes
Light Modulation	573	401	86	86	Long	Mid, Low	Yes
Low Frequency Burst	657	460	99	98	Short	Low	Yes
Low Frequency Lines	453	317	68	68	Long	Low	No
No Glitch	181	127	27	27	Long	-	No
None of the Above	88	62	13	13	Short	High, Mid, Low	Yes
Paired Doves	27	19	4	4	Short	Mid, Low	Yes
Power Line	453	317	68	68	Short	Low	No
Repeating Blips	285	200	69	42	Short	Mid	No
Scattered Light	459	321	69	69	Long	Low	Yes
Scratchy	354	248	53	53	Long	High, Mid	Yes
Tomte	116	81	17	18	Short	Low	Yes
Violin Mode	472	330	71	71	Short	High	No
Wandering Line	44	31	6	7	Long	High	Yes
Whistle	305	213	46	46	Short	High	Yes

Types of glitches in database



Summary

- ❖ Presented a **sample** of applications of machine learning for gravitational waves:
 - Search, parameter estimation, waveform modeling glitch classification
- ❖ Frequent new papers on arxiv
- ❖ **Next class:** practical session!