# Toward Online Deployment of Neural-Networks to Search and Characterize Compact Binary Mergers

Deep Chatterjee
Research Scientist, LIGO MIT

ICERM SciMLGW 2025

# Outline

- Introduction
- LVK Online analyses and alert infrastructure
- Neural Network based algorithms
  - BBH search, Aframe [Marx+ (2024)]
  - Low-latency PE, AMPLFI [Chatterjee+ (2024)]
- Design decisions for online deployment

# Why Neural Networks?

- **Universal function approximators**
  [see Hornick+ (1989) for a proof]

$$\underbrace{\boldsymbol{y}}_{\text{Output}} = f(\underbrace{\boldsymbol{x}}_{\text{Input}}|\underbrace{\theta}_{\text{Trainable parameters}})$$

- Fundamental building blocks
  - Architecture
  - **Loss function**
  - Optimization process/ backpropagation

→ Quality of neural network performance

- Comparison of training data $\boldsymbol{y}$ and neural network prediction $f(\boldsymbol{x})$
- Dependent on the task

# Why Neural Networks?

- **Universal function approximators**
  [see Hornick+ (1989) for a proof]

$$\boldsymbol{y} = f(\boldsymbol{x}|\theta)$$

Output

Input

Trainable parameters

- Fundamental building blocks
  - Architecture
  - Loss function
  - **Optimization process/ backpropagation**

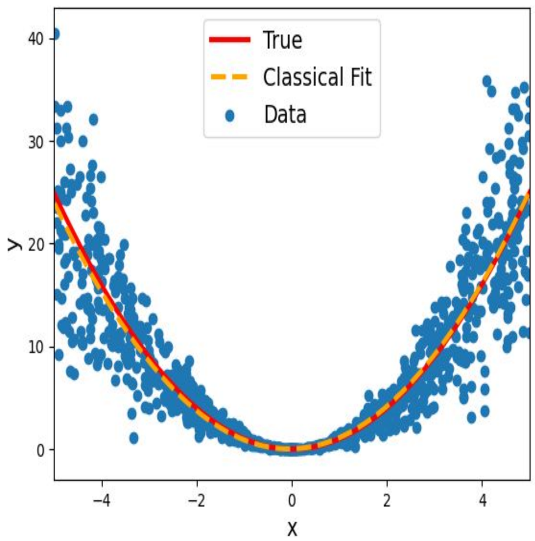
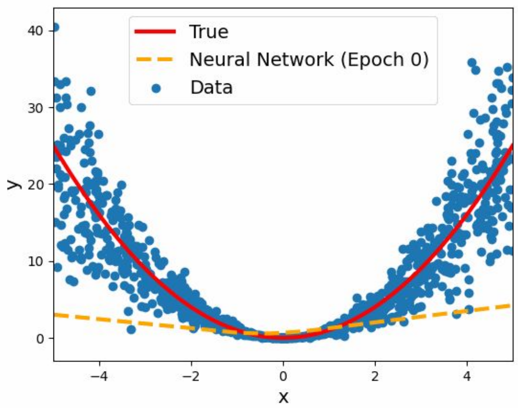
Learning rate

Loss function

$$W_{t+1} = W_t - \alpha \cdot \left[\frac{\partial \mathcal{L}}{\partial W}\right]$$

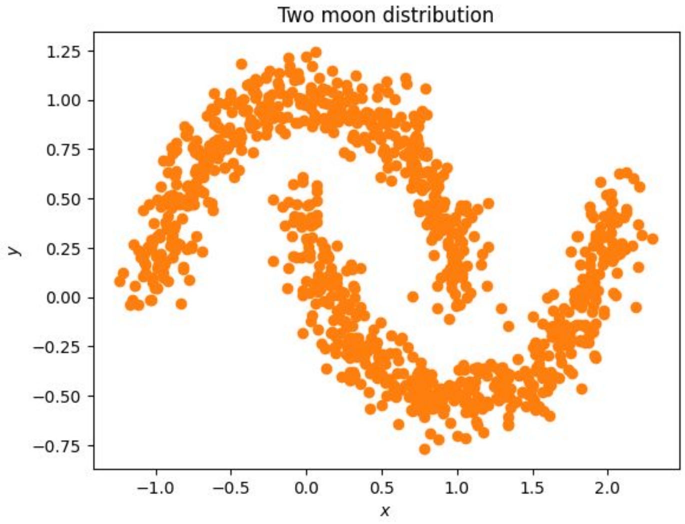
$$b_{t+1} = b_t - \alpha \cdot \left[\frac{\partial \mathcal{L}}{\partial b}\right].$$

# Simple Example

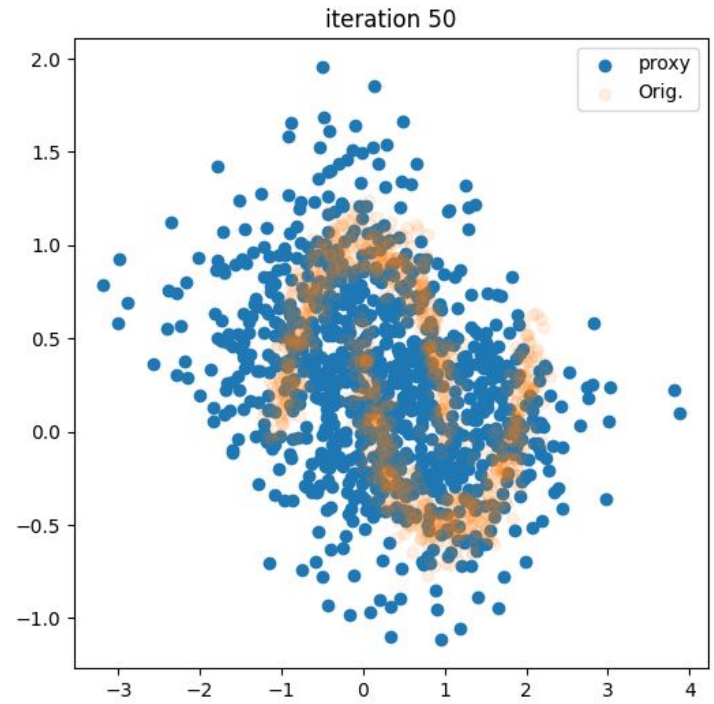
| | Classical Fit | Neural Network |
|---|---|---|
| **Model** | $a + b \cdot \mathbf{x} + c \cdot \mathbf{x}^2$ | $f(\mathbf{x}\|\theta)$ |
| **Optimization Criterium** | $\sum_i (y_i - f(x_i\|a,b,c))^2$ | $\sum_i (y_i - f(x_i\|\theta))^2$ |
| **Result** |   |   |

# Distribution Approximation

Samples from a true dist.



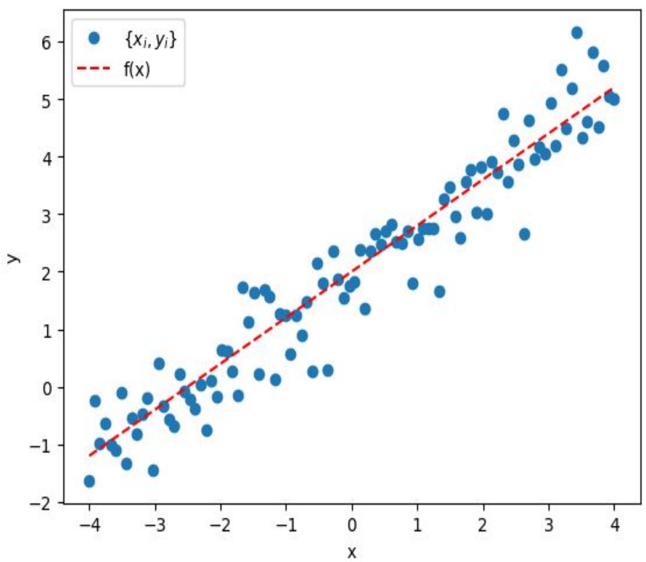

NN approximator

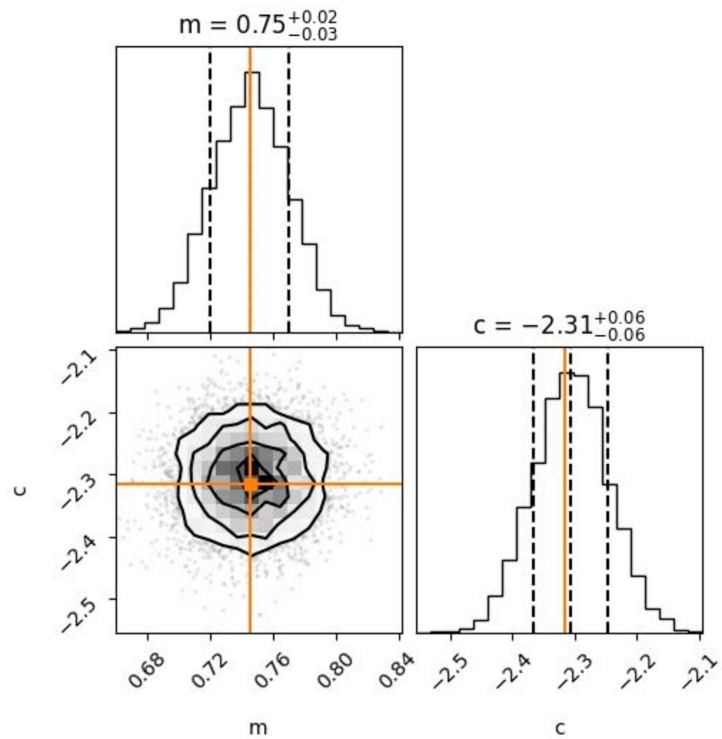



Code at https://shorturl.at/4lN1i

# Bayesian Inference

## Linear regression




Simple Example: Linear Reg.

$$\mathbf{\Theta} = \{m, c\}$$
$$\mathbf{d} = \{\text{Set of points}\}$$

Simulate $\{\mathbf{\Theta}_i, \mathbf{d}_i\}$ ; Approximate $p(\mathbf{\Theta}|\mathbf{d})$

Code at https://shorturl.at/ly5Fm

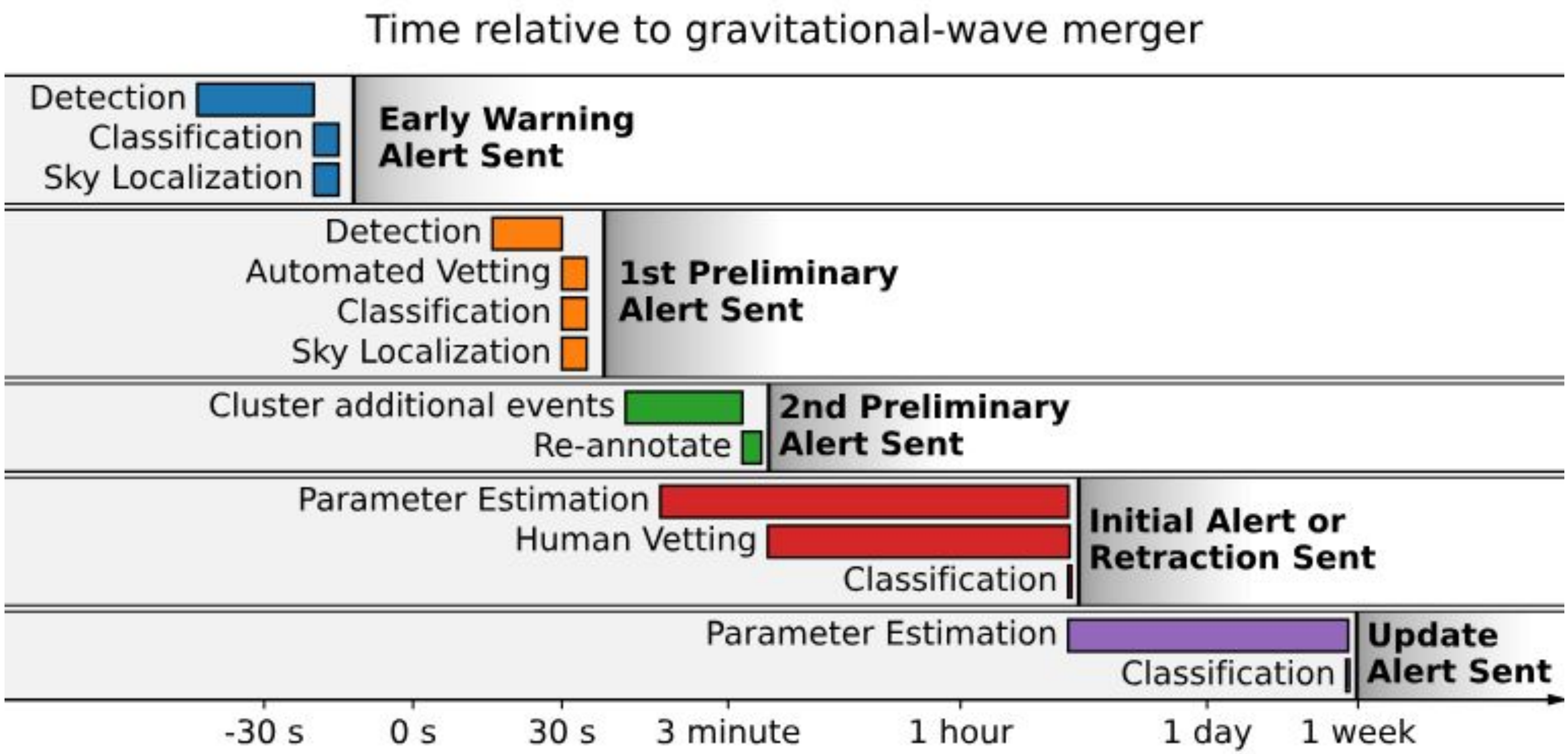
# Global network of Observatories



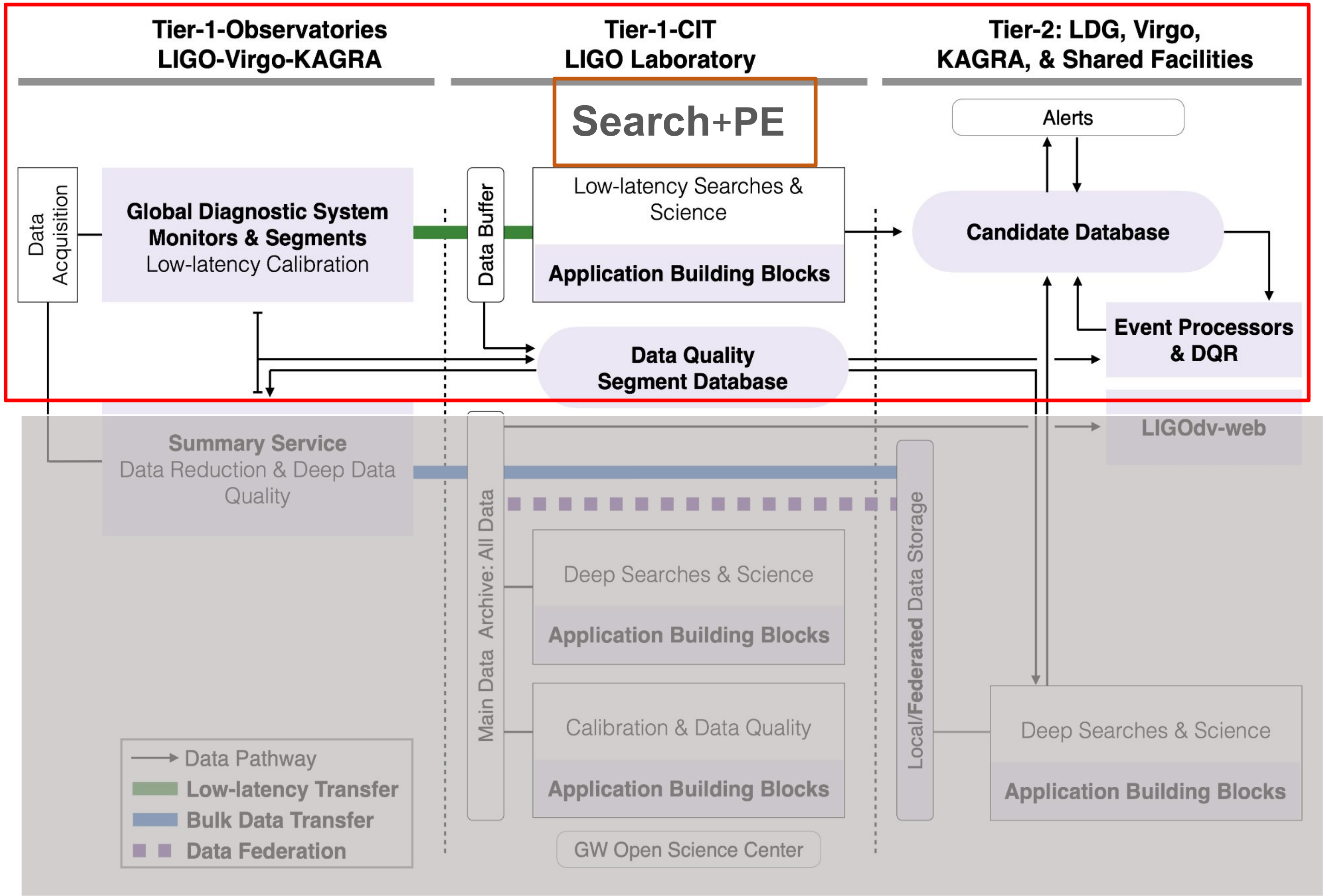


International Gravitational Wave Network (IGWN)

# Low-latency alerts



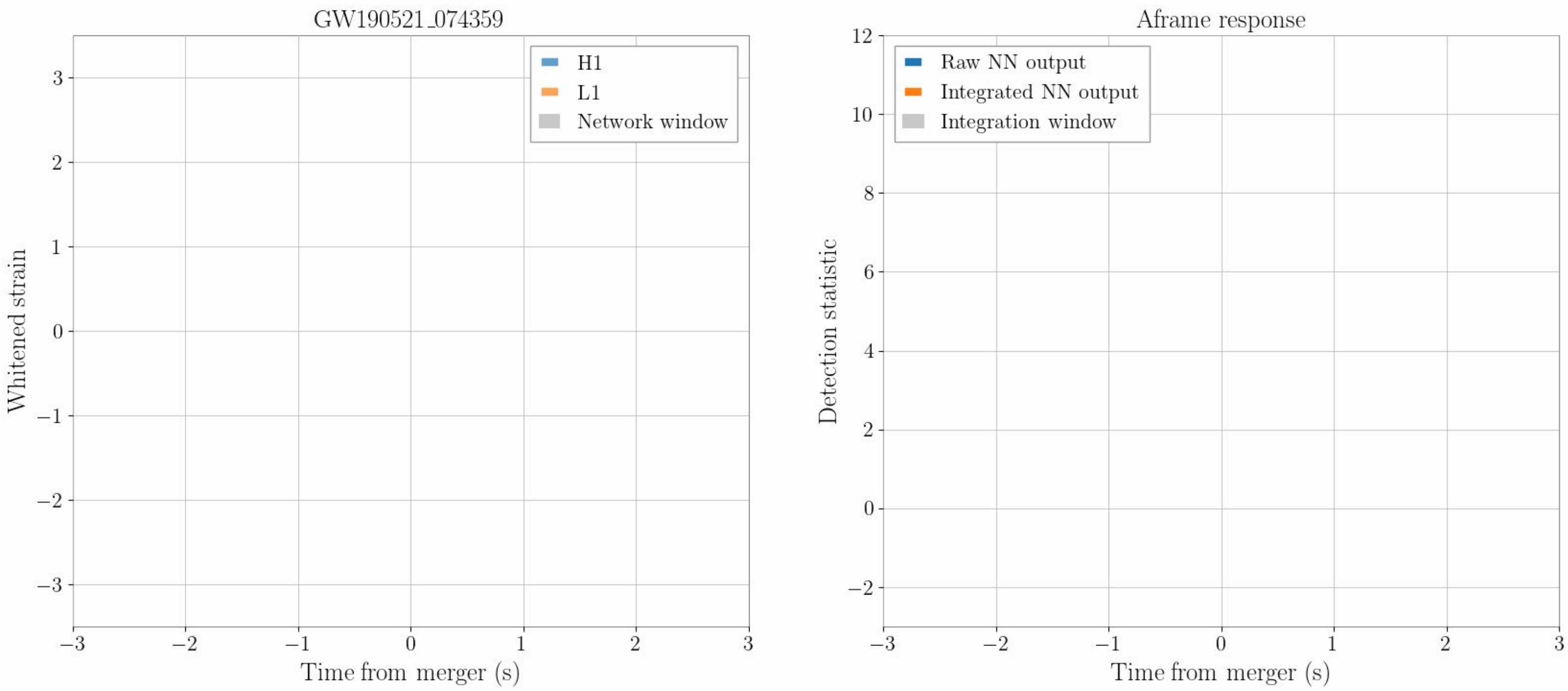
# Online Compute Infrastructure


Credits: DCC-G2100925

# Modeled CBC search: Aframe

Binary Black Hole (BBH) search using Convolutional Neural Networks



# Aframe

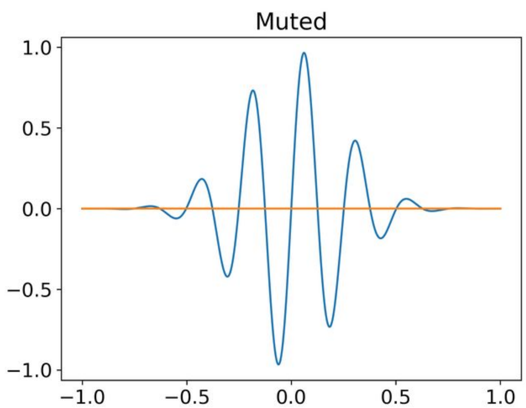
Streaming Binary classification

- ResNet architecture, maps 2-IFO strain $\longrightarrow$ a scalar neural-network output; integrated over 1s window to get detection stat.
- Training time ~40 hours on single NVIDIA V100
- Offline inference ~30 hours for 21 years of livetime on 8 V100 GPUs (~750 s processed / s / GPU).

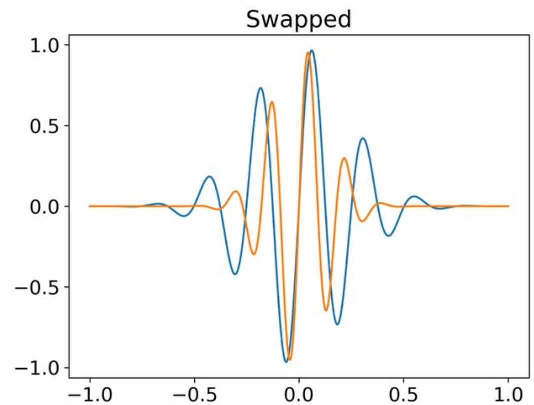
| Parameter | Prior | Limits | Units |
|---|---|---|---|
| $m_1$ | $m_1^{-2.35}$ | $(5, 100)$ | $M_\odot$ |
| $m_2$ | $m_2$ | $(5, m_1)$ | $M_\odot$ |
| $z$ | Comoving | $(0, 2)$ | - |
| $\psi$ | Uniform | $(0, \pi)$ | rad. |
| $a_{1,2}$ | Uniform | $(0, 0.998)$ | - |
| $\theta_{1,2}$ | Sine | $(0, \pi)$ | rad. |
| $\phi_{12}$ | Uniform | $(0, 2\pi)$ | rad. |
| $\phi_{JL}$ | Uniform | $(0, 2\pi)$ | rad. |
| $\phi$ | Uniform | $(0, 2\pi)$ | rad. |
| RA | Uniform | $(0, 2\pi)$ | rad. |
| Dec | Cosine | $(-\pi/2, \pi/2)$ | rad. |
| $\theta_{JN}$ | Sine | $(0, \pi)$ | rad. |

HL Analysis Ready segments
+
O3 R+P mass dist.
+
IMRPhenomPv2

# Aframe Augmentations

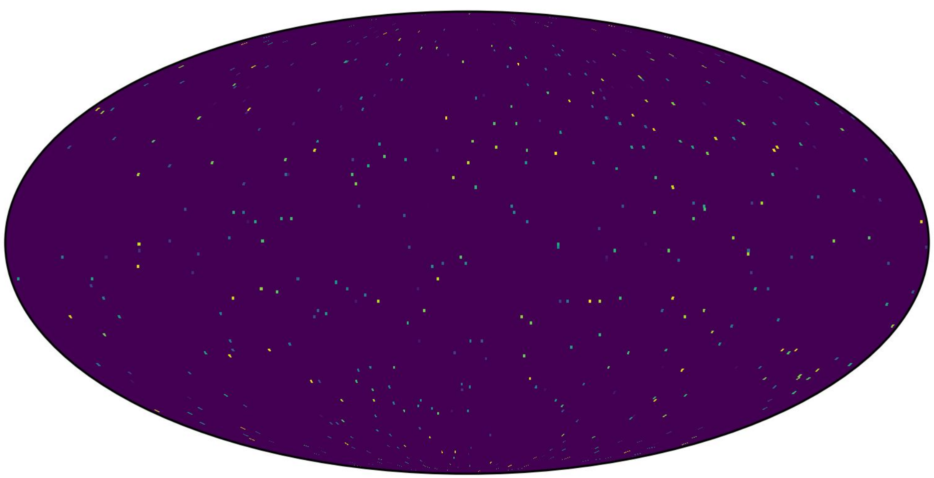



Coincidence




Coherence

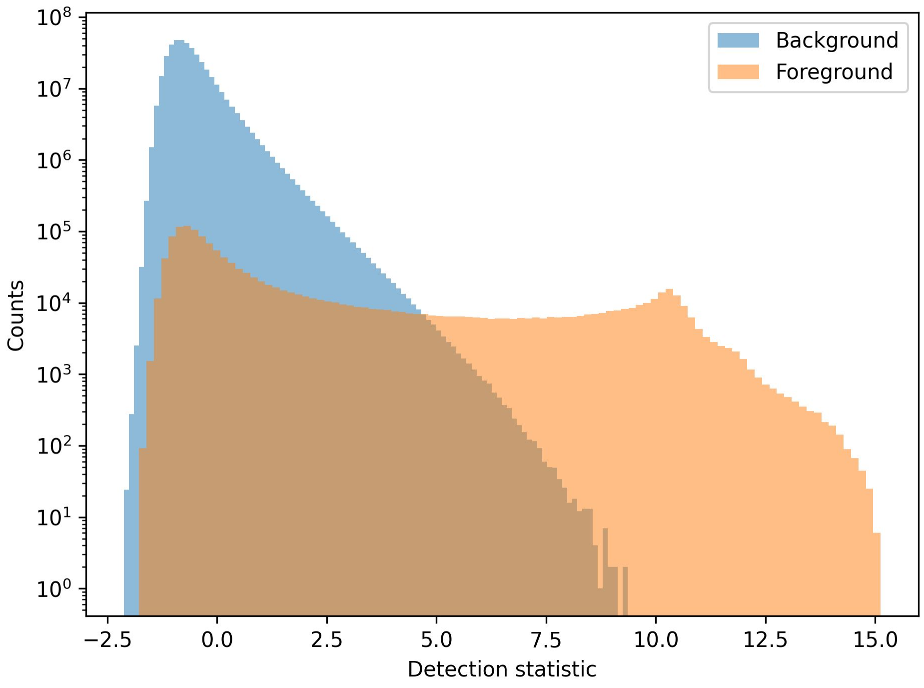
Identified as signal being absent



One waveform; many sky-locations and distances

# Inference Overview

- Background distributions estimated using several years of timeslides
- Foreground distribution estimated through injections
  - Injections drawn from training prior and rejection sampled to SNR > 4, for ~1.4M injections and ~43.8 M rejections
  - FAR assigned using det. stat. relative to the background distribution



# Aframe Performance in O3

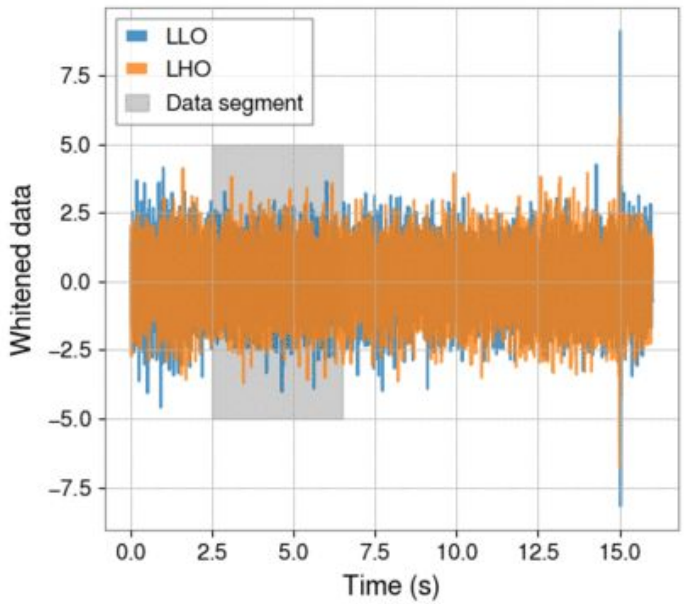
- May 9th, 2019 - June 8th, 2019 used for timeslides
- Systems with representative masses injected
- Sensitive spacetime volume computed (Gpc^3 yr).

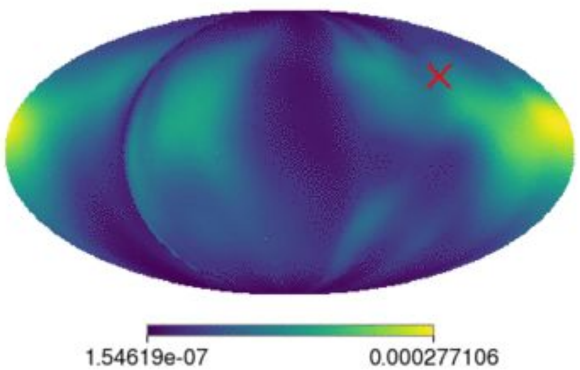
| *Mean masses* | **PyCBC-BBH** | **MBTA** | **GstLAL** | **cWB** | **Aframe @ FAR=1/hour** | **1/month** | **1/year** |
|---|---|---|---|---|---|---|---|
| 35/35 | 1445 | 1321 | 1360 | 1336.5 | 1803 ± 116 | 1461 ± 128 | 1355 ± 141 |
| 35/20 | 1206 | 1080 | 1122 | 1074 | 1568 ± 134 | 1229.5 ± 152 | 1050± 186 |
| 20/20 | 999 | 916.5 | 937 | 801 | 1036 ± 68 | 847.5 ± 78 | 719 ± 89 |
| 20/10 | 770 | 709.5 | 717 | 604 | 686 ± 82 | 584 ± 95 | 477 ± 111 |

*VT computed using p_astro > 0.5

Performance on > 30 sol. mass systems comparable with match filtering

# BBH Param. Est: AMPLFI




$$p(\Theta|\mathbf{d}) \propto p(\mathbf{d}|\Theta)\ p(\Theta)$$

Posterior

Likelihood

Prior

***In MCMC:*** Sample posterior

$$p(\mathbf{\Theta}, \mathbf{d}) \quad p(\mathbf{\Theta}|\mathbf{d}) \quad p(\mathbf{d}|\mathbf{\Theta})$$

NN approximator

$$\{\mathbf{\Theta}_i, \mathbf{d}_i\}$$

***In Likelihood-free Inference:***
Learn the distribution from simulations

# Posterior estimation using Normalizing flows

- Posterior estimation using Likelihood-free inference (LFI)

- Learn an approximator for the posterior from simulations

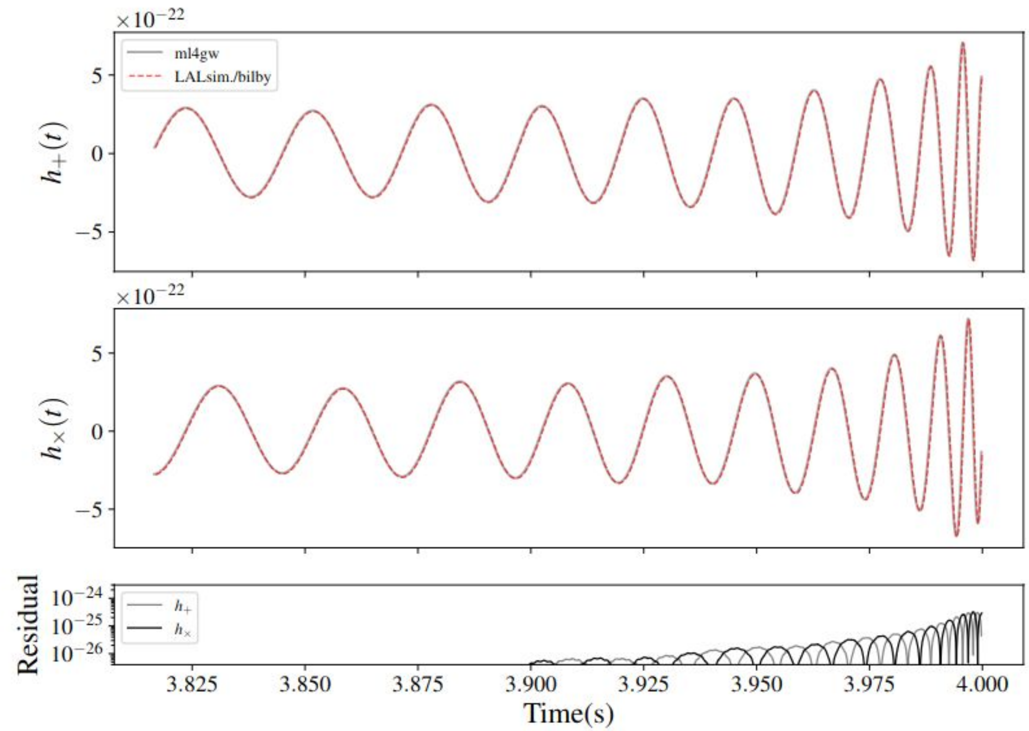
$$\mathbf{\Theta}_i \sim p(\mathbf{\Theta}) \xrightarrow[\text{Simulate}]{} h(\mathbf{\Theta_i}) + \underset{\substack{\uparrow \\ \text{Detector noise}}}{n} \rightarrow \mathbf{d}_i$$

- Use pairs $\{\mathbf{\Theta}_i, \mathbf{d}_i\}$ to learn the approximator.

- In our case the approximator is a normalizing flow.
  - » Flexible neural network transforms, that are learned during training
  - » Transforms the parameters conditioned on data(-summary) into a simple base distribution.

# Waveform generation

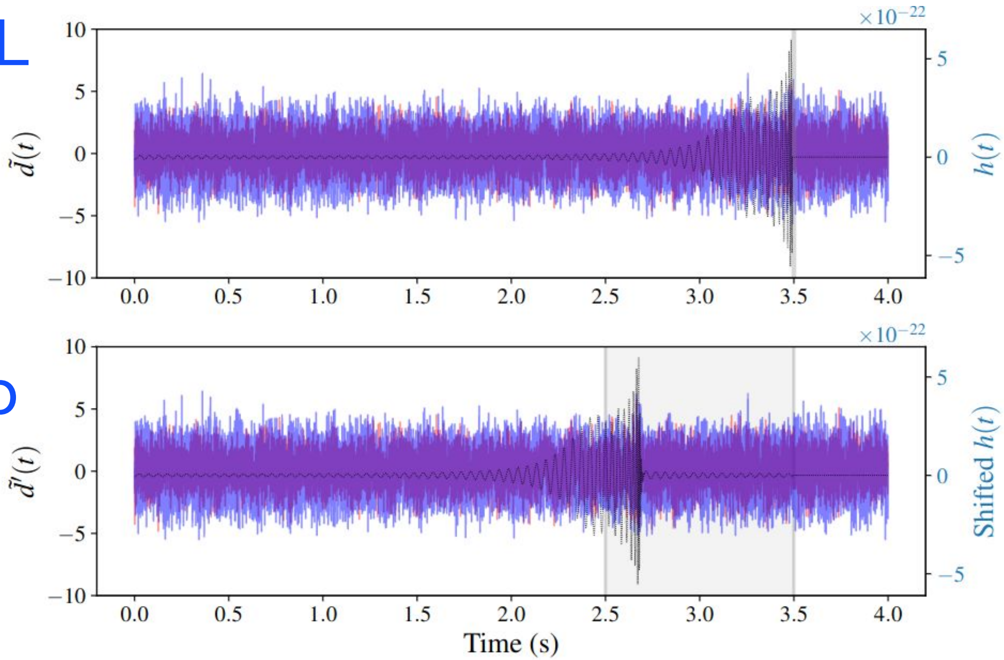
- Re-implement some CBC waveforms as a part of ml4gw.
  - » TaylorF2 (Analytic PN)
  - » IMRPhenomD ( + Merg. Ringdown)
  - » IMRPhenomPv2 ( + precession)
- Consistent with lalsimulation.
  - » Batch of 1000 ~ 0.15s on A40 GPU.




| Parameter | Prior |
|---|---|
| $\mathcal{M}$ | Uniform(10, 100)$M_\odot$ |
| $q$ | Uniform(0.125, 1) |
| $D_L$ | Uniform in Vol.(100, 3000) Mpc ($\sim D_L^2$) |
| $\theta_{\mathrm{JN}}$ | Sine(0, $\pi$) |
| $\alpha$ (RA) | Uniform(0, $2\pi$) |
| $\delta$ (Dec.) | Cosine($-\pi/2$, $\pi/2$) |
| $\phi_c$ (Coal. phase) | Uniform(0, $2\pi$) |
| $\psi$ (Pol. angle) | Uniform(0, $\pi$) |

# Data generation

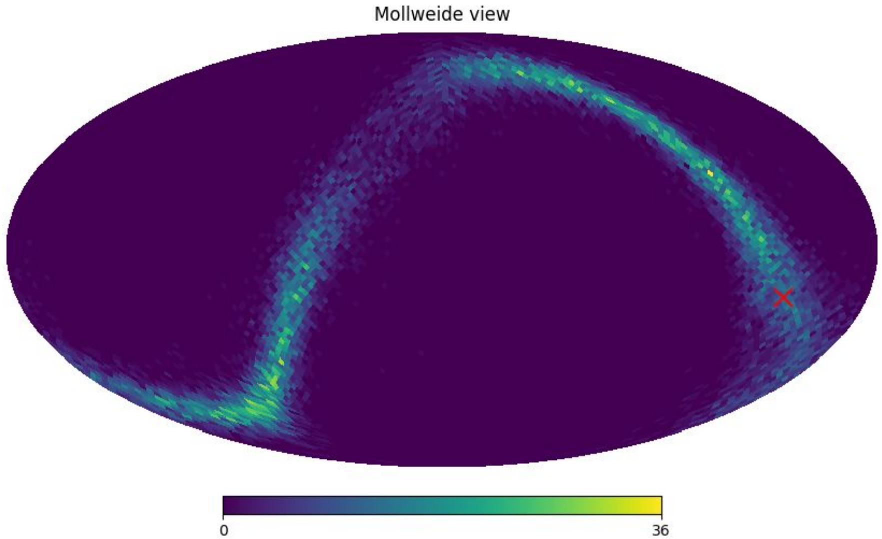
- Use real noise from the HL detectors
  - Stretches of ANALYSIS_READY segments from O3
- Background transferred to GPU
- Data loader
  - Lazily loads batch of 4s segments
  - Samples points from prior; generates, injects, and whiten the data.



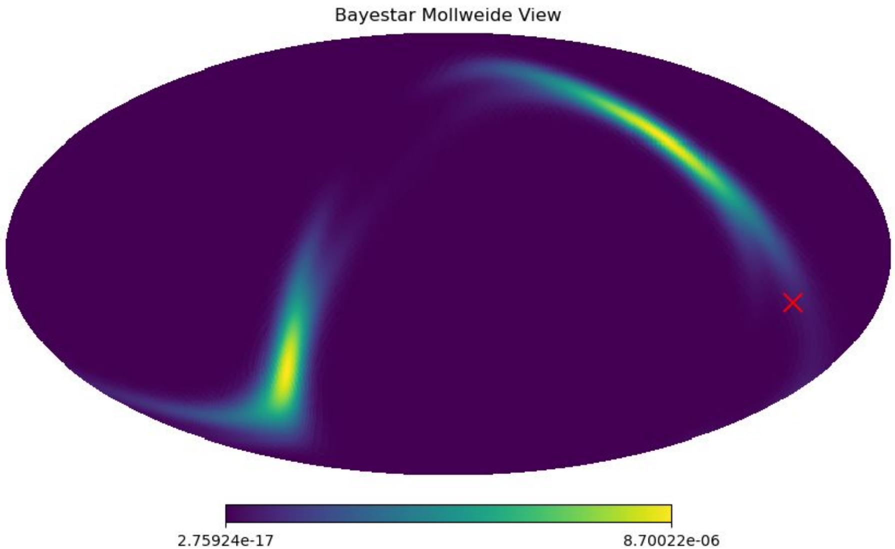

Summarized via Embedding network; Flow conditioned on data-summary

# AMPLFI vs. BAYESTAR

AMPLFI



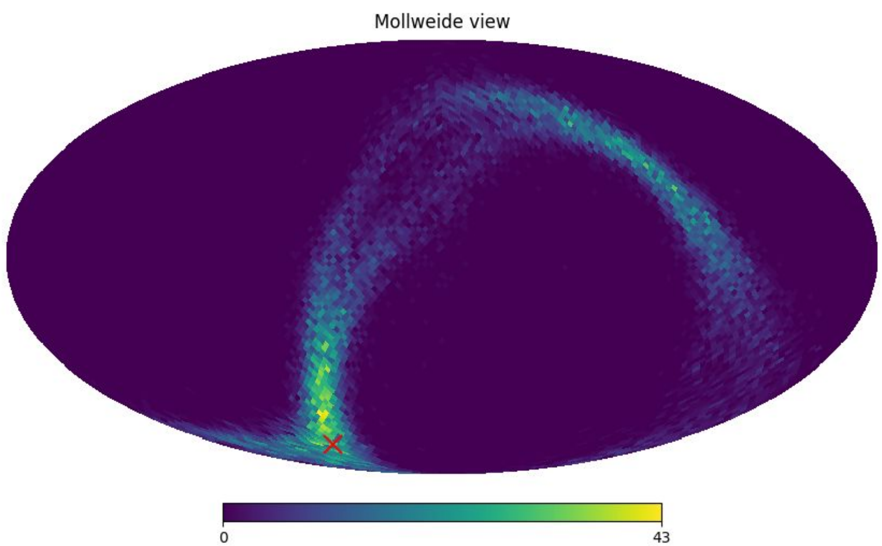

BAYESTAR



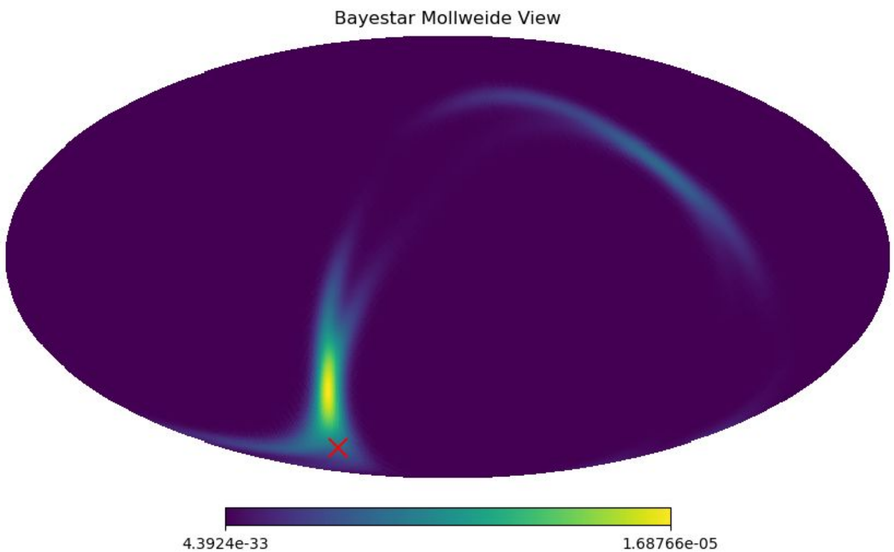

Comparison with BAYESTAR (MDC events)

# AMPLFI vs. BAYESTAR

AMPLFI



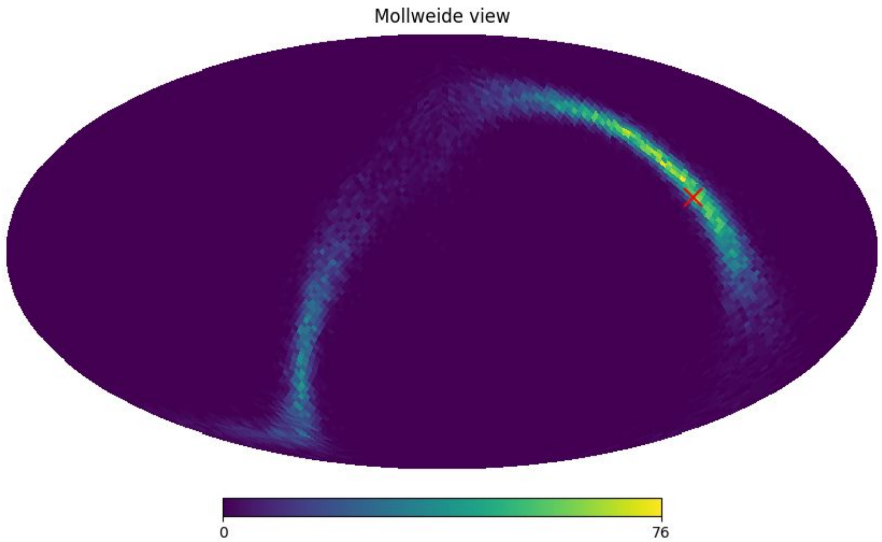

BAYESTAR



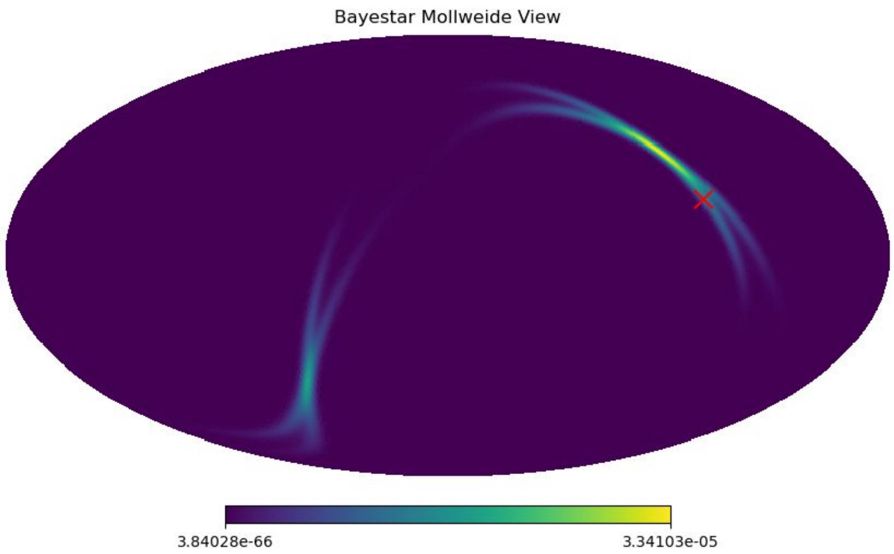

Comparison with BAYESTAR (MDC events)

# AMPLFI vs. BAYESTAR

AMPLFI




BAYESTAR

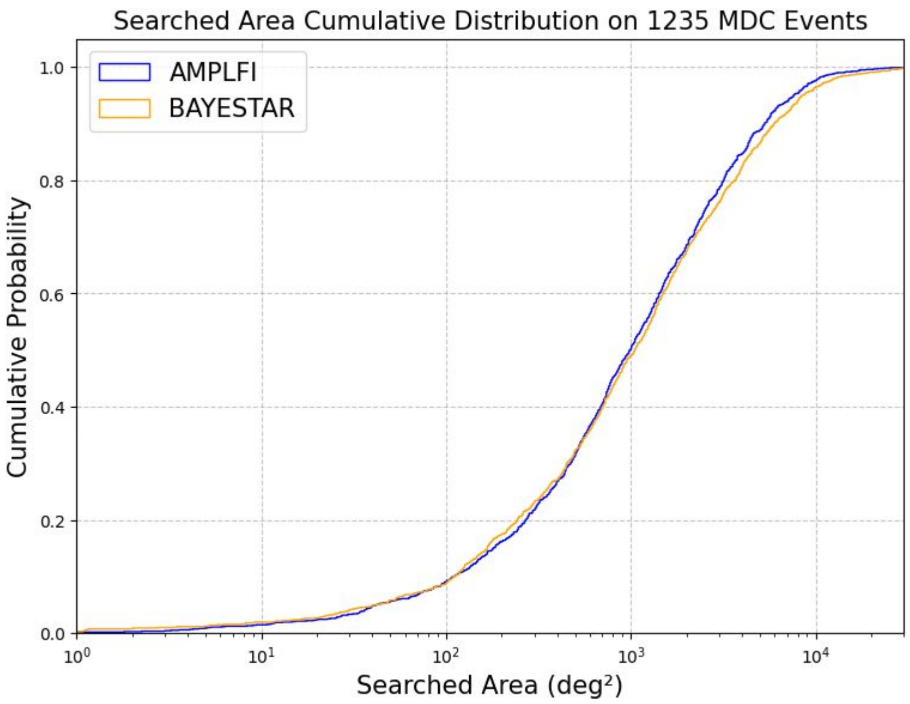



Comparison with BAYESTAR (MDC events)

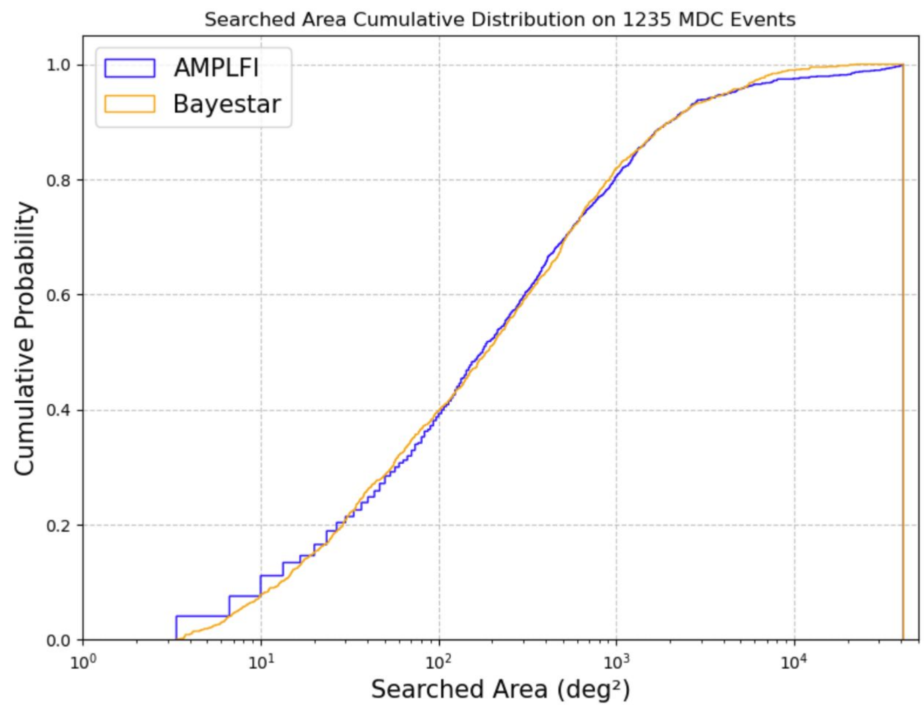
# O3 MDC Searched Area

Analyzed 1235 O3 MDC injections within our training prior and compared with Bayestar localizations

Hanford / Livingston

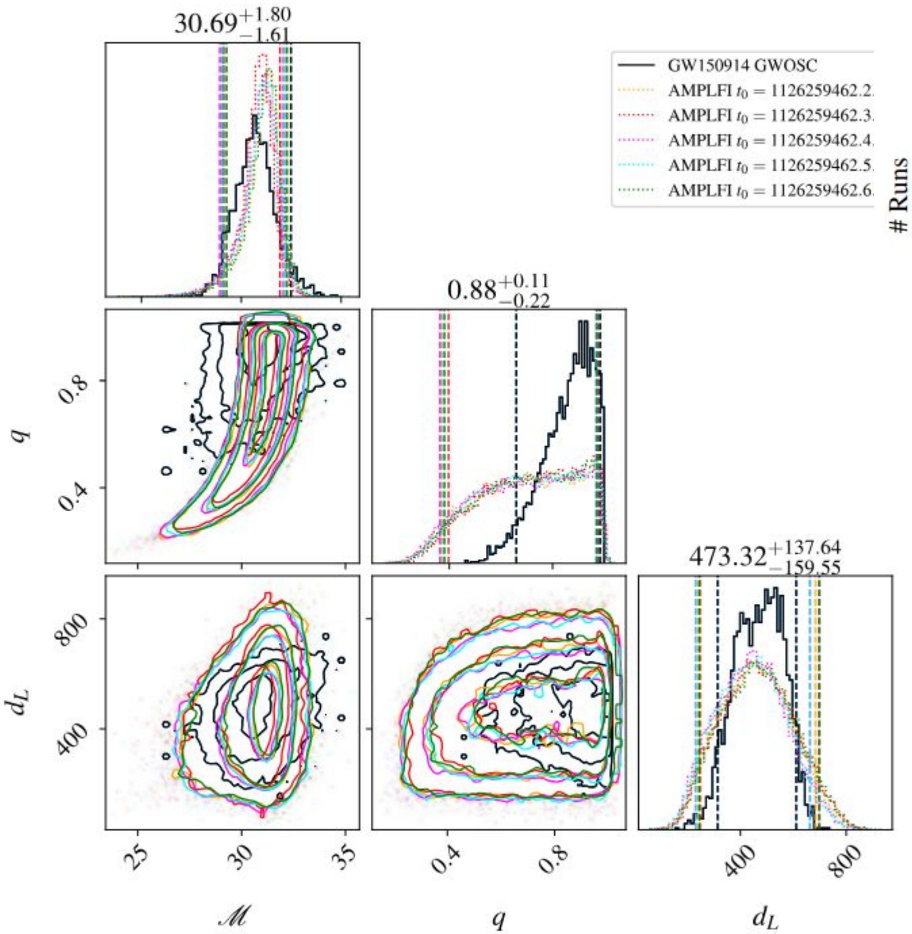


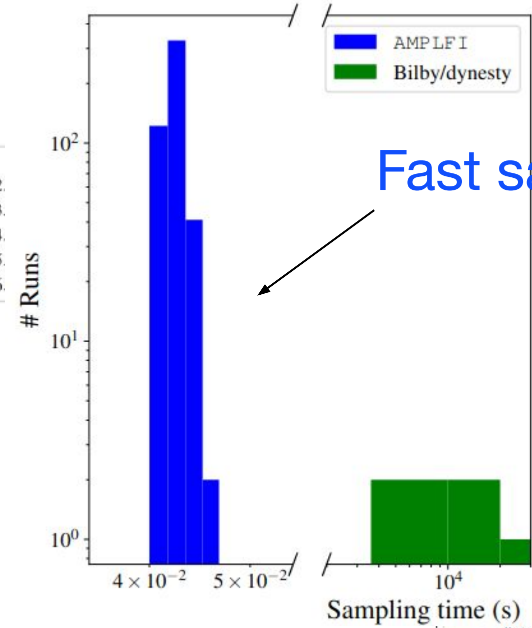

Hanford / Livingston / Virgo



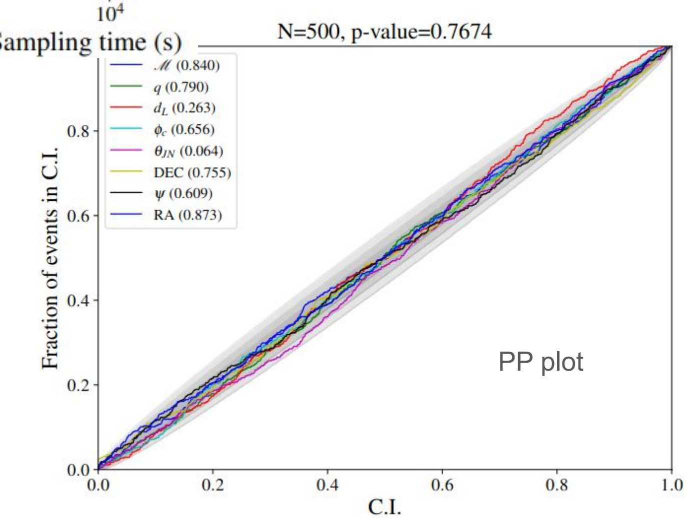
# CBC Param. Est: AMPLFI

GW150914



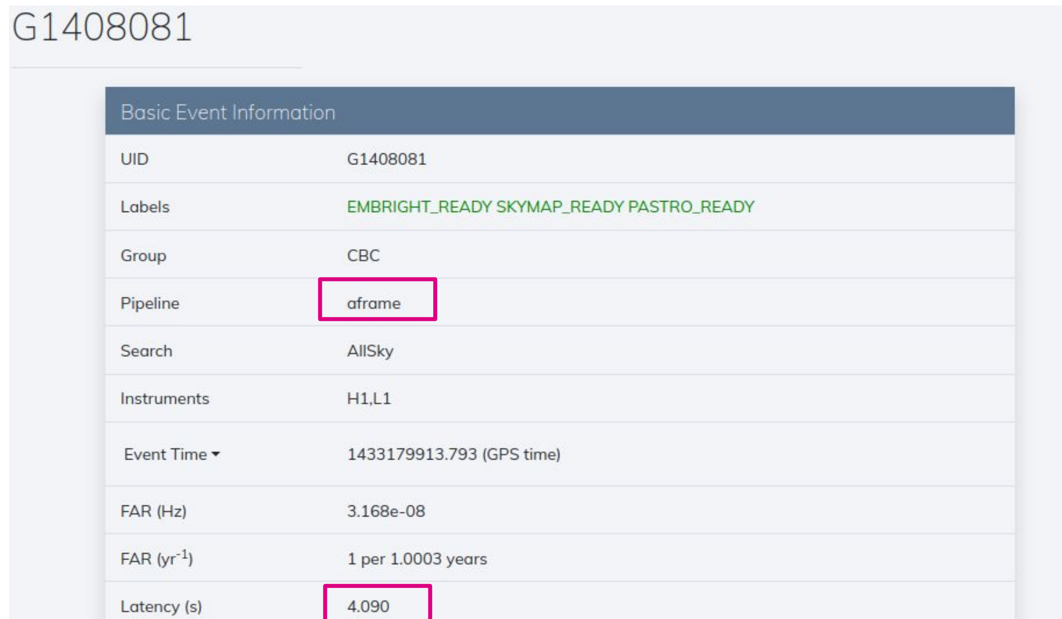

Fast sampling < 1 sec.



# Aframe + AMPLFI live tested!

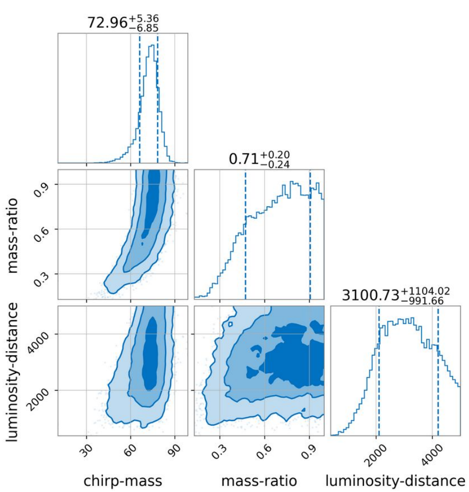
Prototype running on GraceDB-test

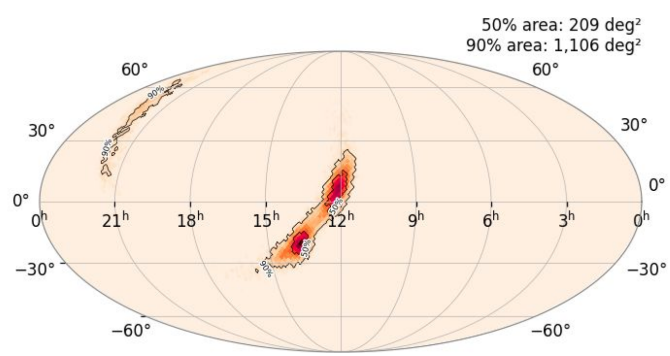
- Search performed by Aframe
- PE on “significant” stretches of data done by AMPLFI
- Net intrinsic latency <~ 6 sec.
- Code review ongoing

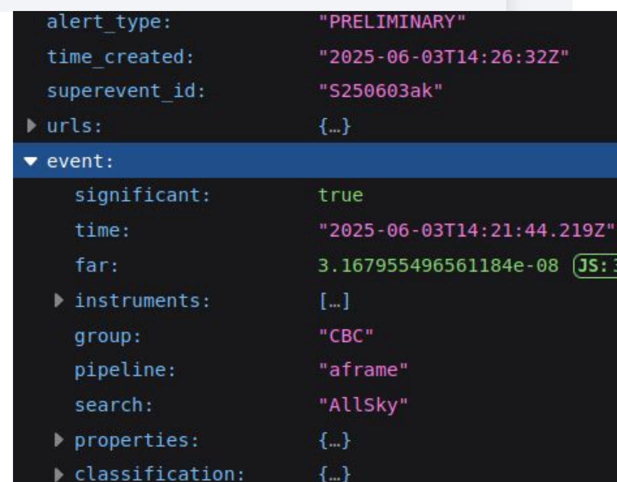


G1408081

| Basic Event Information | |
|---|---|
| UID | G1408081 |
| Labels | EMBRIGHT_READY SKYMAP_READY PASTRO_READY |
| Group | CBC |
| Pipeline | aframe |
| Search | AllSky |
| Instruments | H1,L1 |
| Event Time ▾ | 1433179913.793 (GPS time) |
| FAR (Hz) | 3.168e-08 |
| FAR ($yr^{-1}$) | 1 per 1.0003 years |
| Latency (s) | 4.090 |



# Design Decisions

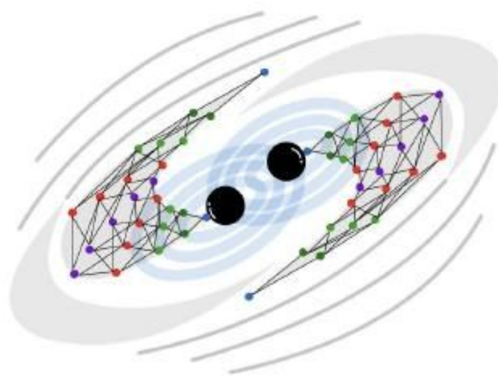
- “Small” models
  - O(10M) trainable parameters for Aframe and AMPLFI.
  - HL Aframe model + HL and HLV AMPLFI models run on a single NVIDIA A30.
  - Parallel testing infrastructure easily maintained on dedicated resources.
  - Model Fine tuning is easily done.
- “Big” data
  - “Compute is cheap; data transfer is expensive”
  - On-the-fly data generation and pre-processing; ensures maximum GPU utilization.
    - Waveform gen., windowing, PSD estimation, whitening, etc. carried out on GPU
  - Training times are 24-48 hours for our models from scratch.
- Search and PE in a single service
  - Online deployment holds ~8 sec of data in GPU mem.
  - Aframe identifies “interesting” segments; data is passed in mem. to AMPLFI reducing I/O
  - Search + Alert data products in < 6 seconds.

# Technology stack

- Models and datasets implemented in Pytorch-lightning.
  - Logging, Visualization, experiment tracking, distributed training come built in.
  - Easy to customize dataloaders to our needs.
  - Adopted over all our projects to use this framework.
- Inference as a service for Offline inference
  - hermes - Inference-as-a-Service deployment based on NVIDIA Triton.
  - Distributed inference, beneficial for estimating hundreds of years of background
- Hyperparameter optimization using Ray Tune.
  - Well integrated with Pytorch lightning.
  - Handy tool, LightRay can be used to control HPO parameters, integrates with Lightning CLI.

# ML4GW software stack

## Backbone for all projects



README.md

# ML4GW

Tools to make training and deploying neural networks in service of gravitational wave physics simple and accessible to all!

Includes applications under active research, such as denoising, signal detection, and parameter estimation algorithms.

With thanks to Vighnesh J R for the profile picture artwork.

NSF award #
PHY-2117997

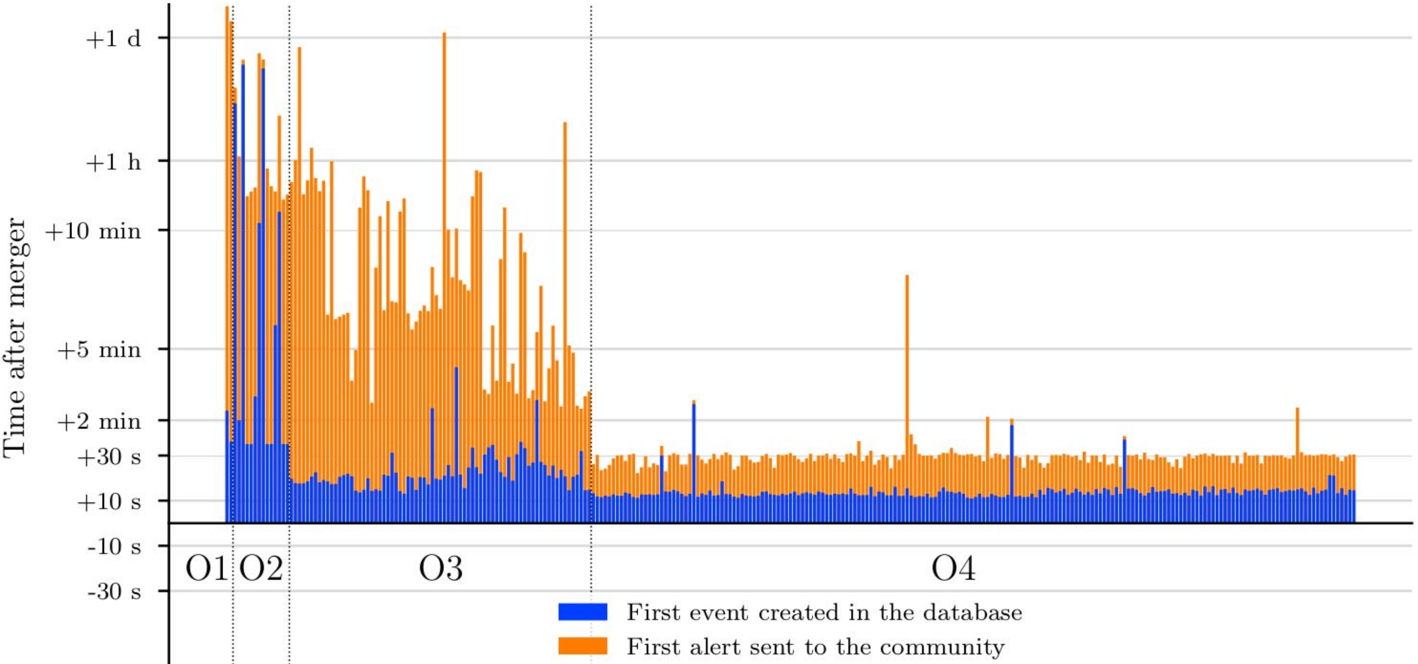
# Aim for wider adoption

- Core data-preprocessing tools of ML4GW are modular and can be used across other projects
  - Some elements adopted by the MLy team
- Several options for using the ML4GW codebase
  - Codebase public
  - PyPI releases
  - Containers available
- Documentation
  - Never enough! But we are building it.
  - Tutorials exists.
- Please connect if this sounds interesting.

Thank you for your attention!

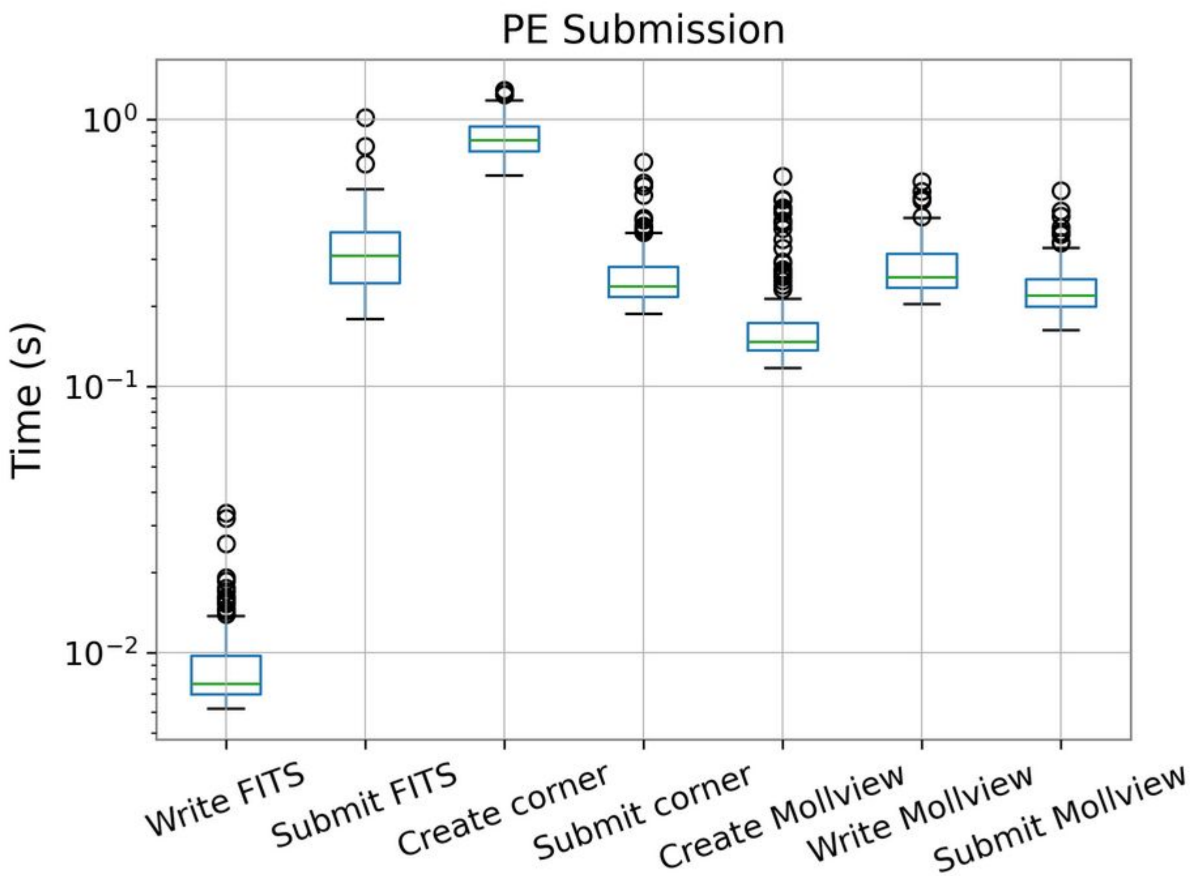
# Extra: O4 Discoveries thus far

- ~200 Significant (False Alarm rate < 1/month) events in O4
  - For reference GWTC-3 (O1+O2+O3) has 90 events.
  - ~3.5K low-significant events (FAR between 2/day and 1/month)
- *Summary*: Increased detection rate @ Better latency in O4!




Met our expectation based on ***Mock Data Challenge*** study [Chaudhary+Toivonen, Waratkar et. al. (2024)]

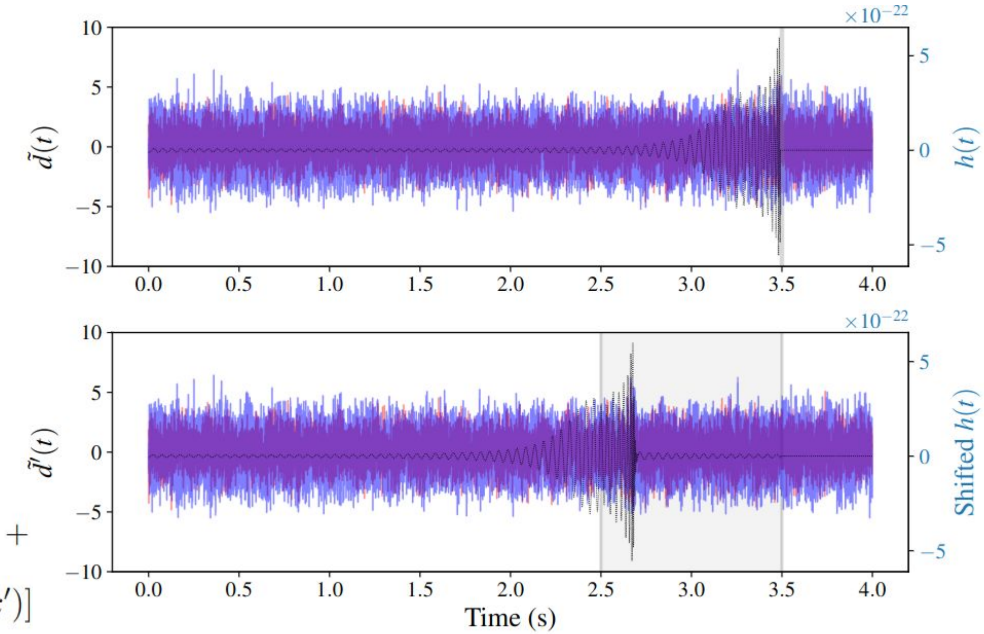
# Extra: AMPLFI latencies



# Extra: Embedding net pretraining

- Make data summary insensitive to time of arrival differences.
- Jointly embed two ''views'' of data.
  - Use a ResNet with 2-channel HL time domain data as input
  - Minimize VICReg.

$$\mathcal{L}_{\mathrm{VICReg}}(x, x') = \lambda_1 \ \mathrm{MSE}(x, x') + \lambda_2 \ \left[\sqrt{\mathrm{Var}(x) + \epsilon} + \sqrt{\mathrm{Var}(x') + \epsilon}\right] + \lambda_3 \ \left[C(x) + C(x')\right]$$

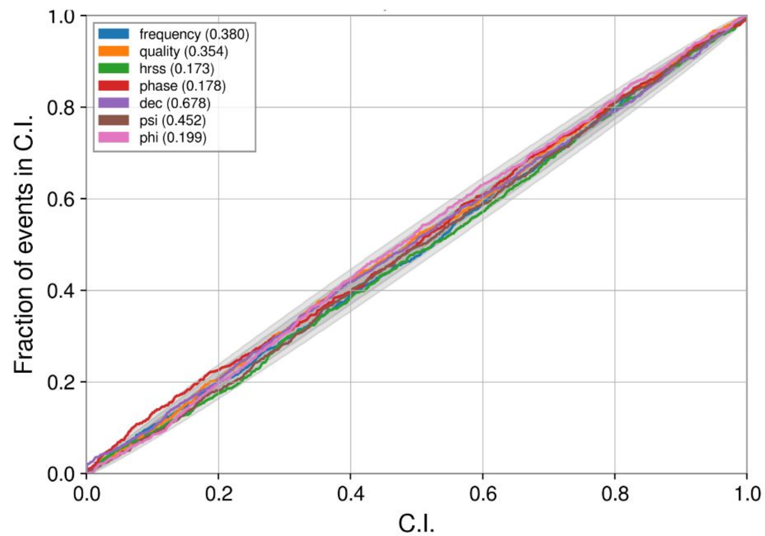


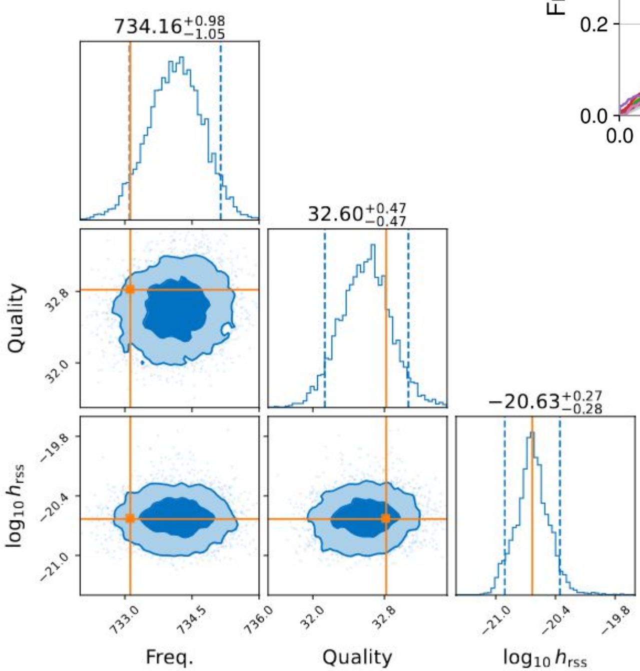

- Obtain data summary vector after hyper-parameter tuning.
- Condition our parameters on this data summary

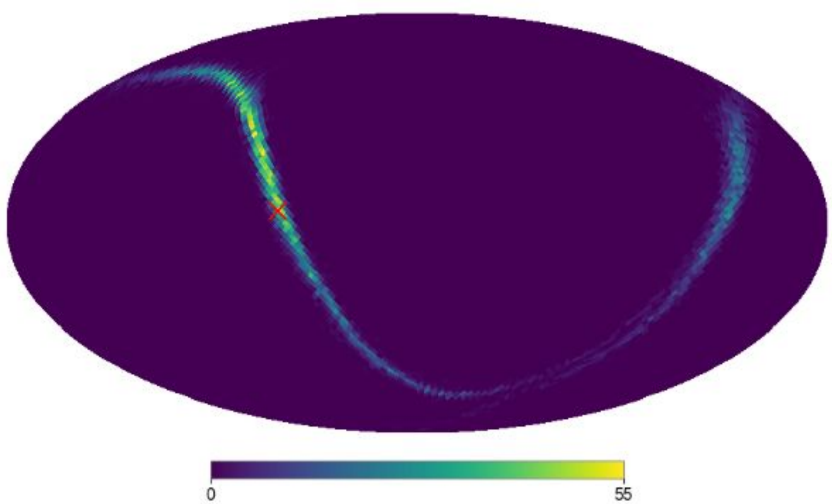
# Extra: Good/Not-so-good GPU utilization

# Extra: AMPLFI on Sine-Gaussians

Similar principle except simulations done using Sine-Gaussians



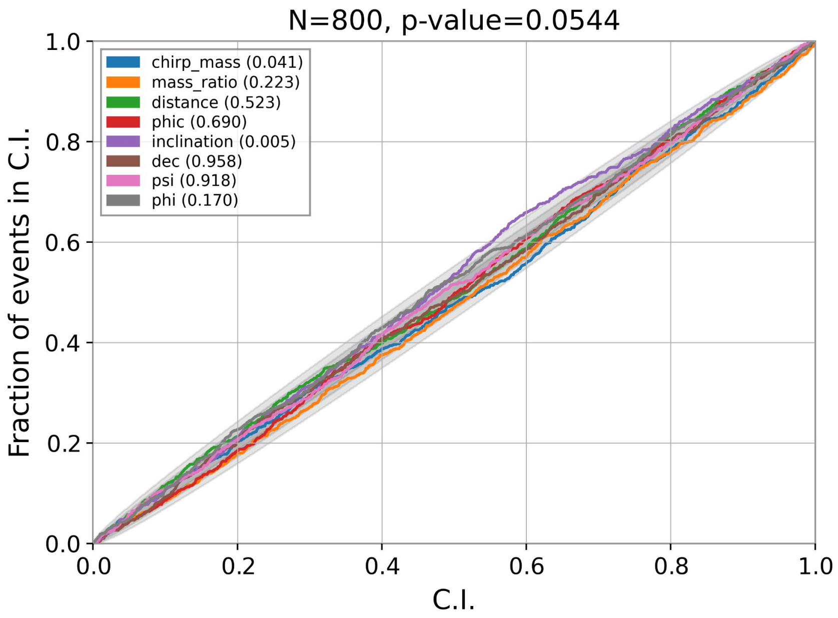



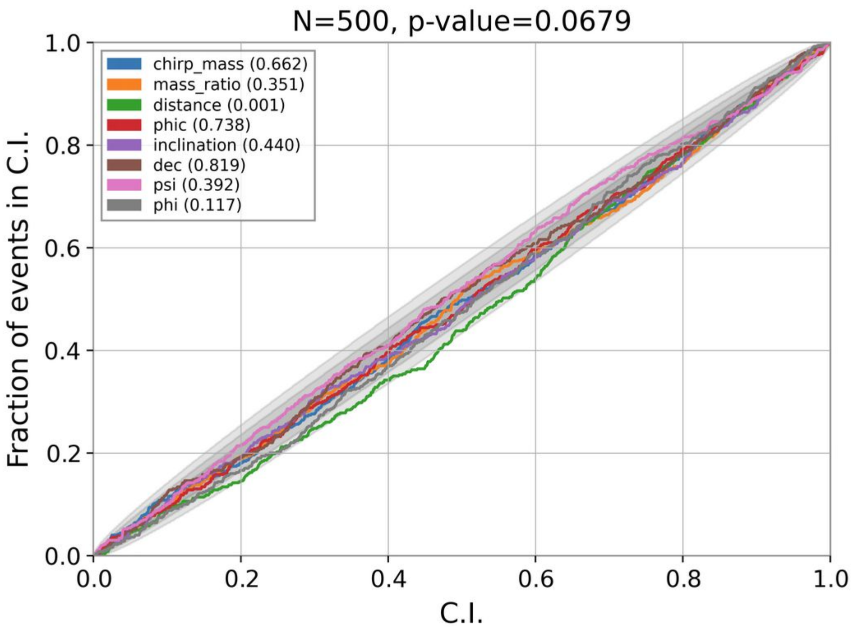
# Extra: Probability-Probability Plots

Parameter recovery is unbiased → diagonal PP plots for injections drawn across training prior

Hanford / Livingston




Hanford / Livingston / Virgo



# Extra - AMPLFI Dev. since paper

## Embedding

Compresses high dimensional data into lower dimensional “data summary”

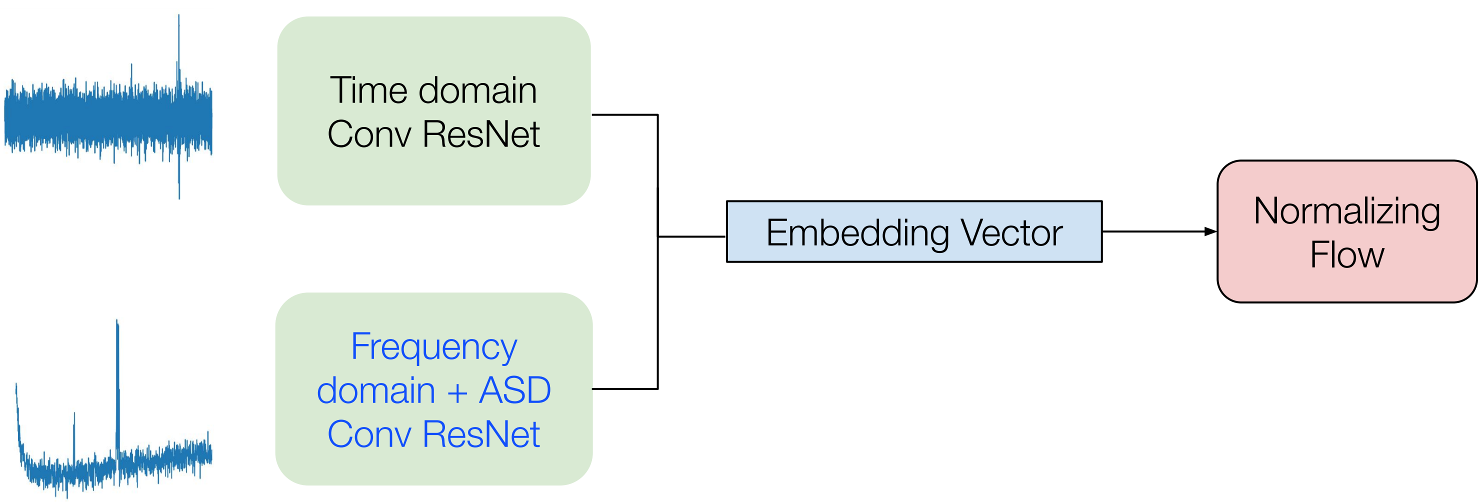
~ 4 million parameters

## Normalizing Flow

Probabilistic model conditioned on data summary

~ 7 million parameters

Provide the embedding network *multi-modal* data: time domain, frequency domain, and ASD’s



# Extra: ML4GW software stack

GW data analysis tools on accelerated hardware.

- Re-implemented data processing operations
  - Dataloading, Windowing, Whitening *real*-detector background using GPUs.
- Re-implemented waveforms on GPUs
  - CBC: TaylorF2, IMRPhenomD, IMRPhenomPv2.
  - Burst: Sine-gaussian, ringdown.
- On-the-fly waveform generation + Injection
  - Minimal disk I/O; Minimal CPU <-> GPU transfers.
  - Training data is generated lazily on device; dataset is infinite;
  - Larger datasets/Smaller models:
    - Consider AMPLFI: 6M parameter model.
    - Typical training run sees:
      250 ep. x 200 batch per ep. x 800 batch size ~ 40M unique parameter/data combinations passed to the model.
    - Not overspecified!

# Extra - Hyperparameter Opt.

```
# tune.Tune.param_space
param_space:
  model.learning_rate: tune.loguniform(1e-5, 1e-3)
  model.weight_decay: tune.loguniform(1e-5, 1e-3)
  model.arch.embedding_net.time_context_dim: tune.choice([12, 20, 25, 30, 35])
  model.arch.embedding_net.freq_context_dim: tune.choice([32, 48, 56, 64, 72, 80])
  model.arch.transform_type: tune.choice(["affine", "spline"])
```

24-GPU HPO trials ~ 12h

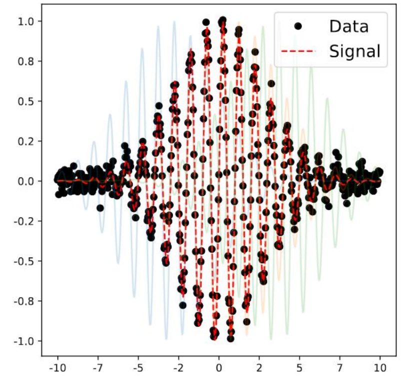
Early-stop bad performing trials

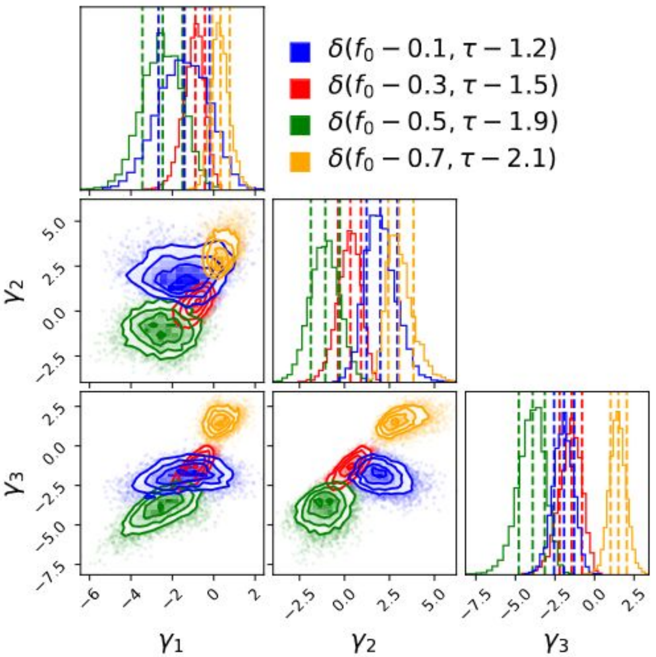

# Extra - Use of self-supervision

- Marginalize out parameters by joint embedding
  - A batch of data with a fixed reference time of arrival
  - A second batch where the time of arrival is varied
  - Embed the data and use a similarity loss between the batches. We use VICReg loss.

$$\mathcal{L}_{\mathrm{VICReg}}(x, x') = \lambda_1 \, \mathrm{MSE}(x, x') + \lambda_2 \, [\mathrm{Var}(x) + \mathrm{Var}(x')] + \lambda_3 \, [\mathrm{Cov}(x) + \mathrm{Cov}(x')] \, ,$$

  - Use the embedded space as a data summary.
  - Condition parameters on this summary.



# LIGO Extra - Aframe non-catalog events

## Non-GWTC-3 Catalog Candidates

`Aframe` finds **10 triggers** at FAR < ~12 per year not reported in GWTC-3

Most significant trigger not reported in GWTC-3 found at FAR of **3.6** per year

Top 2 most significant non-LVK catalog triggers also reported by Princeton IAS analysis which reported an additional 15 candidates (arxiv: 2201.02252, 2311.06061)

| gpstime | FAR (1 / yr) |
|---|---|
| 1262635012.75 | 3.6 |
| 1246523564.75 | 4.0 |
| 1264333383.00 | 4.2 |
| 1238351045.00 | 4.4 |
| 1251010355.50 | 4.7 |
| 1264246793.25 | 5.9 |
| 1262163593.25 | 7.8 |
| 1249032684.75 | 11.0 |
| 1253452013.50 | 11.7 |
| 1259411705.25 | 12.0 |