Likelihood-free inference for gravitational-wave data analysis and public alerts

Ethan Marx,[1, 2] Deep Chatterjee,[1, 2] Malina Desai,[1, 2] Ravi Kumar,[3, 4] William Benoit,[4]
Argyro Sasli,[4] Leo Singer,[5] Michael W. Coughlin,[4] Philip Harris,[1] and Erik Katsavounidis[1, 2]

[1]*Department of Physics, MIT, Cambridge, MA 02139, USA*
[2]*LIGO Laboratory, 185 Albany St, MIT, Cambridge, MA 02139, USA*
[3]*Department of Aerospace Engineering, IIT Bombay, Powai, Mumbai, 400076, India*
[4]*School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA*
[5]*Astroparticle Physics Laboratory, NASA Goddard Space Flight Center, Code 661, Greenbelt, MD 20771, USA*
(Dated: September 29, 2025)

Rapid and reliable detection and dissemination of source parameter estimation data products from gravitational-wave events, especially sky localization, is critical for maximizing the potential of multi-messenger astronomy. Machine learning based detection and parameter estimation algorithms are emerging as production ready alternatives to traditional approaches. Here, we report validation studies of AMPLFI, a likelihood-free inference solution to low-latency parameter estimation of binary black holes. We use simulated signals added into data from the LIGO-Virgo-KAGRA's (LVK's) third observing run (O3) to compare sky localization performance with BAYESTAR, the algorithm currently in production for rapid sky localization of candidates from matched-filter pipelines. We demonstrate sky localization performance, measured by searched area and volume, to be equivalent with BAYESTAR. We show accurate reconstruction of source parameters with uncertainties for use distributing low-latency coarse-grained chirp mass information. In addition, we analyze several candidate events reported by the LVK in the third gravitational-wave transient catalog (GWTC-3) and show consistency with the LVK's analysis. Altogether, we demonstrate AMPLFI's ability to produce data products for low-latency public alerts.

I. INTRODUCTION

A decade has passed since the first direct detection of gravitational waves (GWs) [1]. Now, observing GWs from compact binary coalescences (CBCs) is commonplace. The number of GW candidates has increased by two orders of magnitude in this decade, rising from three candidates in the first observing run, to over three-hundred cumulative after the fourth observing run (O4) [2–5]. The discovery rate is expected to continue to rise as the Advanced LIGO [6], Advanced Virgo [7] and KAGRA [8] interferometers approach the design sensitivity era of the fifth observing run [9]. Accompanying this rapid growth has been excitement to transmit real-time public alerts to the broader astrophysical community, with the goal of finding coincident electromagnetic (EM) signals.

Machine learning (ML) methods for detecting and characterizing CBCs are emerging as production ready solutions to real-time GW data analysis. For detection, many methods are being developed, showing promise in achieving the sensitivity of traditional matched-filter pipelines at a reduced latency and computational cost [10–12]. However, these methods do not provide a solution for producing the suite of low-latency data products necessary for informing effective EM followup after an initial detection. Currently, the LVK's low-latency alert infrastructure relies on data products expected from matched-filter pipelines. For example, the production sky localization algorithm for events from matched-filter pipelines, BAYESTAR, requires signal to noise ratio (SNR) time series [13]. In addition, coarse-grained chirp

mass information (a recent addition to LVK alerts[1]) and other source property classifiers utilize point estimates from the best fit matched-filtering template [14]. ML-based search algorithms do not naturally provide SNR time series or template parameters required by these downstream tasks.

In Ref. [15], we introduced AMPLFI[2], a parameter estimation (PE) algorithm based on likelihood-free inference (LFI), as a tool for providing low-latency PE data products. We motivated AMPLFI as a real-time followup to ML-based detection algorithms, specifically, Aframe [10]. In this paper, we describe recent improvements to AMPLFI, and demonstrate its ability to produce robust low-latency source PE data products.

This paper is organized as follows. In Sec. II we provide a summary of LFI in GW astronomy. In Sec. III we discuss the AMPLFI algorithm previously presented in Ref. [15] including improvements made since. In Sec. IV we present the dataset used to evaluate AMPLFI's performance. In Sec. V, we discuss and benchmark our method for producing sky localizations from AMPLFI posterior samples. In Sec. VI A we compare AMPLFI's sky localization performance with BAYESTAR [13]. In Sec. VI A 2 we demonstrate the self-consistency of AMPLFI sky localizations. In Sec. VI C, we illustrate AMPLFI's robustness to changing detector background over a time scale of several months. Finally, in Sec. VI D we analyze real GW candidates from GWTC-3 and show

---

[1] https://emfollow.docs.ligo.org/userguide/
[2] **A**ccelerated **M**ultimessenger **P**arameter estimation using **LFI**; pronounced 'amp-li-fy'.

consistency with published GWTC-3 results.

II. LIKELIHOOD FREE INFERENCE

In GW astronomy, estimating the parameters θ of a source given strain data d is typically done using Bayesian inference. Traditional Bayesian PE algorithms rely on repeated evaluations of an explicit likelihood function, utilizing stochastic sampling techniques such as Markov Chain Monte Carlo (MCMC) or nested sampling to explore the signal parameter space [16, 17]. For GW PE, this process can be arduous. Each likelihood evaluation requires simulating a waveform, which can be expensive. In addition, the full 15 dimensional signal parameter space must be fully explored. Low-latency PE methods typically make approximations, to either the likelihood function or waveform physics, to simplify the exploration of the posterior [18–21].

Posterior estimation using LFI involves training a probabilistic neural network with tuneable parameters ϕ, $q_\phi(\theta|d)$, to approximate the true posterior $p(\theta|d)$. A type of probabilistic neural network known as a conditional normalizing flow [22] is a popular choice for constructing the approximation $q_\phi(\theta|d)$. Conditional normalizing flows consist of flexible, parameterized variable transforms which map between a simple base distribution, $\pi(u)$, typically taken to be a standard normal, and a more complex distribution which is being approximated (e.g. $q_\phi(\theta|d)$). The normalizing flow is conditional in the sense that the parameters that define the transformations from the base distribution to the more complex distribution are a function of the data d. Normalizing flows have been utilized in GW astronomy for a variety of use cases [23–26].

Training normalizing flow models requires large quantities of data realizations from the likelihood. This means a diverse set of both noise instances, n, and signal simulations $h(\theta)$ such that one can simulate $d = n + h(\theta) \sim p(d|\theta)$. The utility of LFI is that once the approximator is trained, generating samples from the posterior $\theta \sim q_\phi(\theta|d)$ can be done rapidly by drawing samples from the base distribution and applying the learned transformations to obtain the posterior sample. Drawing thousands of samples takes seconds when done on accelerated hardware like GPUs. In addition, LFI also provides density estimation, or scoring, of samples i.e. $p(\theta|d) \approx q_\phi(\theta|d)$.

III. AMPLFI

AMPLFI is a PE algorithm based on LFI and was presented in detail in Ref. [15]. In Sec. III A we summarize the algorithm and in Sec. III B we discuss in more detail improvements that have been made regarding training data usage and neural network architecture.

A. Summary

Applications of LFI with normalizing flows to GW PE of CBC sources have been pioneered in Ref. [23]. While the core principle of AMPLFI is similar, there are significant implementation differences, namely

- An empirical noise model that makes use of real detector data to sample noise instances for generating simulations from the likelihood (see Sec. VI C).

- Implementation of GW signal approximants with PyTorch [27] such that signals can be generated using GPUs on-the-fly during the training process. This implies that the training dataset is effectively infinite, and does not need to be pre-generated or saved to disk.

- Real-time data processing steps like filtering, power spectral density (PSD) estimation, and whitening employed on the GPU.

These algorithmic and data processing choices allow a maximally diverse, high entropy training dataset while still maintaining effective GPU utilization. In more detail, the training loop for a batch size of N training examples includes

1. Sampling N parameter instances $\theta^{(i)}$, $i \in \{1, ..., N\}$, from a prior distribution $p(\theta)$ of waveform parameters. We use the `torch.distributions` library for the priors which allows direct sampling on the GPU.

2. Generating intrinsic signal polarizations $h_\times(\theta^{(i)})$ and $h_+(\theta^{(i)})$ in batch. This is done using signal generators implemented in `torch` and publicly available in the ml4gw library[3].

3. Sampling right ascension, declination and polarization angle uniformly and projecting waveforms onto the interferometers, producing the observed signal $h_k(\theta^{(i)})$ in the kth interferometer.

4. Sampling N noise instances $n_k^{(i)}$ from data stored on disk, independently in time for each interferometer k. This includes data used for estimating the PSD (see Sec. VI C for details).

5. Injecting the waveforms into detector noise i.e., $d_k^{(i)} = h_k(\theta^{(i)}) + n_k^{(i)}$. The injection is performed in the time domain into 3 seconds of noise. The coalescence time is randomly placed with uniform probability between 0.4 and 0.6 seconds from the right edge of the window.

---

[3] https://github.com/ML4GW/ml4gw

6. Creating whitened data, $\tilde{d}_k^{(i)}$, using a PSD estimate $S_k^{(i)}$ local to each training sample.

7. Jointly mapping the whitened data and PSD estimates for all interferometers, $\{\tilde{d}_k^{(i)}, S_k^{(i)}\}$, to a lower dimensional data summary $\gamma^{(i)} = \Gamma_\psi(\{\tilde{d}_k^{(i)}, S_k^{(i)}\})$ using a neural network Γ with weights ψ that are optimized during training.

8. Providing the N combinations of parameters and data summaries, $\{\theta^{(i)}, \gamma^{(i)}\}$ to the neural network, with the training objective of minimizing the negative log-probability with respect to weights ψ and ϕ.

$$\min_{\phi,\psi} \frac{1}{N} \sum_{i=1}^{N} -\log q_\phi\left(\theta^{(i)}|\gamma^{(i)}\right), \qquad (3.1)$$

In this work, strain data is sampled at 2048 Hz.

B. Improvements

1. Training Data

In Ref. [15], AMPLFI was trained and evaluated using a fraction of a day of strain data. As such, accounting for variability in detector noise was not critical to performance. So, a single, global estimate of the PSD was used to whiten the data for each training batch. When deployed to a production environment, any deviations in noise properties could lead to model under performance. Practically, using a global PSD estimate encoded the assumption of noise-stationarity into the model.

In this work, the quantity of strain data used for training is increased to ~ 2 months. A global PSD estimate is no longer sufficient at modeling the varying noise profiles of the detectors across this long of a period. Instead, we now utilize data local to each training sample to estimate the PSD. For each training noise instance, 64 seconds of strain data immediately preceding the sample is used to estimate the PSD using Welch's method with median averaging. Increasing the quantity of strain data and using local PSD estimates allows a richer sampling of detector noise states, leading to improved generalization and model longevity. This amount of data cannot fit into memory at once. We take advantage of an efficient out-of-memory data loader implemented in ml4gw so that data loading does not bottleneck the training process. This is same data loader implementation is used to train the Aframe detection algorithm [10].

Lastly, in Ref. [15], AMPLFI was trained with a GPU-accelerated IMRPhenomD approximant [28]. Since then, we have implemented the IMRPhenomPv2 waveform approximant, allowing AMPLFI to be trained with a waveform that includes spin precession physics.

2. Neural Network

In Ref. [15], the AMPLFI embedding network Γ_ψ consisted of a 1 dimensional convolutional ResNet [29], which processed time-domain representations of the data to produce a lower-dimensional data summary. Here, we introduce a parallel embedding network that processes a frequency-domain representation of the signal and a local PSD estimate, S. The data summaries produced by the time and frequency embedding networks, denoted γ_t and γ_f, respectively, are concatenated into a final data summary, γ, which is used to condition the normalizing flow.

$$\gamma_t = \Gamma_{\psi_t}^t(\tilde{d}) \qquad (3.2)$$
$$\gamma_f = \Gamma_{\psi_f}^f(FFT(\tilde{d}), S) \qquad (3.3)$$
$$\gamma = \text{Concat}(\gamma_t, \gamma_f) \qquad (3.4)$$

The real and imaginary parts of the frequency domain data are processed as separate channels by the frequency domain embedding network. In equations 3.3 and 3.4, ψ_t and ψ_f correspond to the trainable parameters of the time domain and frequency domain embedding networks, $\Gamma_{\psi_t}^t$ and $\Gamma_{\psi_f}^f$, respectively. The dimensions of γ_t and γ_f are hyperparameters that can be optimized, and control how much information from each domain is passed to the normalizing flow.

Providing redundant data in the form of the Fourier transform can be interpreted as an inductive bias. Intuitively, different data representations provide easier pathways for the neural network to learn certain parameters. As an example, the coalescence time might be easier to identify in the time domain, whereas the chirp mass, a quantity determined by a power law in frequency, might be easier in the frequency-domain. Multi-modal learning approaches have been explored in other GW data analysis contexts [30, 31].

In addition, we also increase the number of trainable parameters of the normalizing flow. In, Ref. [15], affine flow transforms were used [32]. Here, we use the more expressive spline-based transforms [33]. We reduce the number of auto-regressive transforms, and compensate by increasing the hidden dimensions and number of hidden layers of the hyper-networks used to learn the parameters of the spline transformations. Normalizing flows are implemented with the zuko library [34].

For the studies that follow, we train two AMPLFI models for Hanford-Livingston (HL) and Hanford-Livingston-Virgo (HLV) detector configurations. Both HL and HLV neural networks utilize the same fundamental architecture described above. However, the HLV network utilizes a larger normalizing flow component. See Table I for hyperparameters used in this work. For each detector configuration, we use coincident science mode segments from the beginning of the second half of the third observing run (O3b) until the start of the testing dataset (see Sec. IV) for training. We train the models using

the low-latency calibrated strain data. The embedding and normalizing flow for the HLV (HL) model total 70 (30) million parameters, a factor of ~ 10 increase from Ref. [15]. Training this larger model takes ~ 10 days using 2 NVIDIA A100 GPUs.

Hyperparameter	HL	HLV
# Transforms	15	20
# Hyper-network layers	3	3
# Hidden units	1024	1024
size of γ_f	32	48
size of γ_t	12	20

TABLE I: Neural network hyperparameters for the HL and HLV models used in this work.

IV. TESTING DATASET

In preparation for O4, Ref. [35] performed a study which added simulated CBC signals (known colloquially as *injections*) into a real-time data replay. The injections were added across a one month period during O3b, between 2020-01-05T:23:59:42 and 2020-02-14T23:59:42. Search pipelines analyzed the data replay using their online configurations. GWCelery[45] processed pipeline triggers, mimicking the end-to-end real-time alert infrastructure. This study provided matched-filter pipeline SNR time series for injections across a broad parameter space.

During the real-time analysis, a given injection may be detected by multiple different matched-filter pipelines, creating several associated SNR time series to use for localization. For comparison, we choose the localization corresponding to the preferred event in GraceDB. This corresponds to the matched-filter event with the highest SNR. In some instances, the preferred event is associated with the un-modeled cWB pipeline which does not produce sky localizations with BAYESTAR. For these instances, we select the matched-filter event with the highest SNR.

Using these SNR time series, we create a dataset of BAYESTAR localizations. We first filter the injections to those consistent with AMPLFI's training prior (see Table II). The injections are binary black hole mergers with chirp masses between 10 and 100 $M_\odot$. We ensure the injections were performed using science quality data from each interferometer. Using the SNR time series, we create Hanford-Livingston (HL), and if available, Hanford-Livingston-Virgo (HLV) BAYESTAR localizations. BAYESTAR is configured to use the same distance prior used to train AMPLFI. After these cuts, this leaves
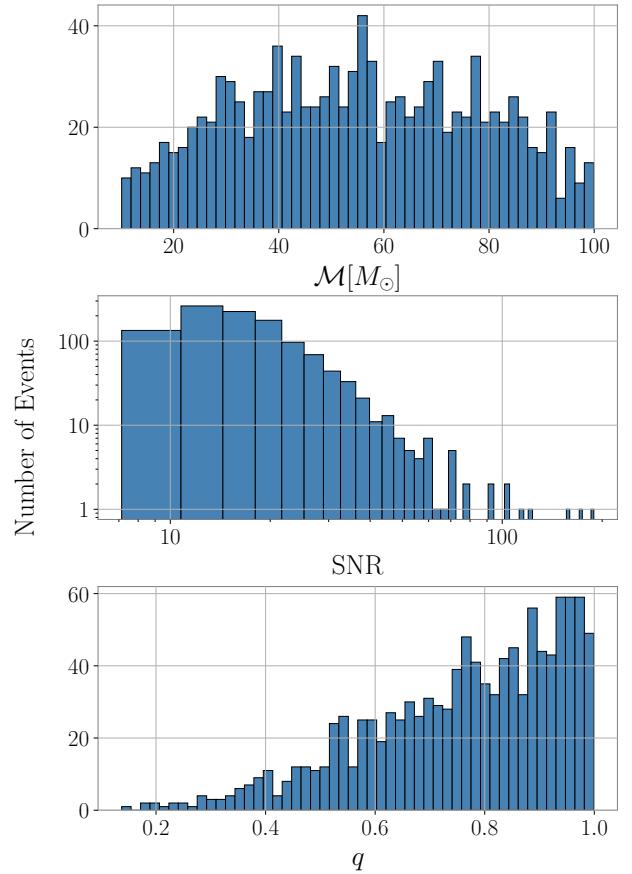

FIG. 1: Histograms of chirp mass (top), SNR (middle) and mass ratio (bottom) for the 1233 HL injections used in this study to compare with BAYESTAR. The HLV injections are a strict subset of the HL injections.

1233 HL and 903 HLV injections. Figure 1 visualizes the distribution of signal parameters. The dataset consists of a broad parameter space of signals, nearly uniform in chirp mass between 10 and 100 $M_\odot$ and covering a wide range of SNRs.

V. SKY MAP PRODUCTION

GW sky maps are probability density distributions on the sky constructed using the HEALPix[6] format [36]. For CBC events with distance calibrated waveform models, three-dimensional volume information can also be estimated. BAYESTAR estimates the parameters of a conditional distance distribution ansatz for each sky pixel which is distributed as part of the sky map in low-latency GW discovery alerts [37].

For stochastic sampling algorithms which draw samples from the posterior, a method for estimating each sky

---

Parameter		Prior
$a_{1,2}$	(Spin magnitude)	Uniform(0, 0.999)
ϕ_{12}	(Spin azimuthal angle)	Uniform(0, 2π)
ϕ_{jl}	(Spin phase angle)	Uniform(0, 2π)
$\mathrm{tilt}_{1,2}$	(Spin tilt angle)	Uniform(0, π)
Inference		
$\mathcal{M}_c$	(Chirp mass)	Uniform(10, 100) $M_\odot$
q	(Mass ratio)	Uniform(0.125, 1)
d_L	(Luminosity distance)	Uniform (100, 3100) Mpc
θ_{jn}	(Inclination)	Sine(0, π)
α	(Right ascension)	Uniform(0, 2π)
δ	(Declination)	Cosine($-\pi/2$, $\pi/2$)
ϕ_c	(Coalescence phase)	Uniform(0, 2π)
ψ	(Polarization angle)	Uniform(0, π)

TABLE II: Prior distributions for source parameters used to generate waveforms during training. Parameters which AMPLFI is trained to infer are specified. We use a spin parameterization consistent with Bilby. Note that the distance prior is uniform in distance. So, cosmological effects are not included, but can be incorporated in post-processing through importance sampling.

pixels probability density and distance ansatz parameters from the samples is required. Here, we compare two techniques for estimating the densities and distance ansatz parameters: a kernel density estimate (KDE) using the `ligo-skymap-from-samples` tool, and an adaptive histogram estimator which adaptively grids the sky, creating higher resolutions in regions with higher numbers of posterior samples. Both of these tools are available via the `ligo.skymap`[7] library. We show that the searched area performance for the two methods is equivalent. For searched volume, the performance is comparable at mid to high SNRs. However, the KDE begins to outperform the adaptive histogram estimator at low SNRs ($\lesssim 10$).

The adaptive histogram density is estimated as follows

1. Draw approximately 10^4 samples from the normalizing flow. This step takes $\sim 1 - 2$ seconds on an NVIDIA A30 GPU.

2. Bin the samples using an adaptive HEALPix grid. Resample the grid to a flat resolution with NSIDE=64. This corresponds to pixel area of ~ 1 deg.^{2} and provides sufficient resolution for all-sky survey instruments which have fields of view $1 - 50$ deg^2 [27, 38].

3. Using the posterior samples in each sky pixel, calculate the first two distance moments, and solve

for the distance distribution ansatz parameters as described in Ref. [39].

4. De-rasterize the sky map into multi-order scheme [40]. This reduces the data size of the sky map with no information loss, allowing for low-latency distribution via alert brokers (e.g. GCN and SCiMMA[8]).

We calculate the conditional distance ansatz parameters for pixels with ≥ 5 samples. For pixels which have a smaller number or no samples, we use placeholder values as done in Ref. [39]. The adaptive histogram estimator including distance estimation can be run in less than a second.

The KDE is constructed by modeling $p(\alpha, \delta, d_L)$ as the product of the 2D marginal distribution $p(\alpha, \delta)$ and the conditional distance distribution $p(d|\alpha, \delta)$. For both, posterior samples are clustered using a k-means algorithm into k clusters. A brute-force search over k is performed in order to maximize the Bayesian information criterion. For each cluster, a separate KDE is constructed, and the total distribution is modeled by summing the cluster KDEs with appropriate weights. The conditional distance distribution $p(d|\alpha, \delta)$ is evaluated as $p(\alpha, \delta, d_L)/p(\alpha, \delta)$ by analytically marginalizing the 3D KDEs. See Sec. 5 in [39] for more details. Due to the brute-force search over k, the runtime is ~ 11 seconds parallelized on 64 CPUs, which is not suitable for extremely rapid alerts.

Using the dataset of injections described in Sec. IV, we compare the accuracy of the AMPLFI sky map produced using the histogram and KDE estimators. For each injection, we calculate searched area and searched volume metrics for both estimators. Searched area (volume) is measured by sorting sky area (volume) pixels by descending probability, and accumulating area (volume) until the pixel containing the true location of the simulated signal is reached. For the adaptive histogram estimator, we draw 20,000 samples from the model. The top row of Figure 2 shows that searched areas from the two methods are in statistical agreement. However, the histogram estimator has a minimum searched area of ~ 1 deg^2 due to the pixel resolution of NSIDE=64, and is therefore unable to achieve searched areas smaller than this. The bottom row compares the searched volumes between the estimators. The methods agree for searched volumes $\lesssim 10^9$ Mpc^3. For larger searched volumes, the KDE method performs better. This behavior is due to the large number of samples required for the histogram estimator to properly converge for low SNR events, where the sky localization probability is spread across a larger number of pixels. In practice, these low SNR events are not as likely to be pursued for EM followup. For example, the Rubin target of opportunity observing strategy plans to pursue EM followup for events with sky localizations less than
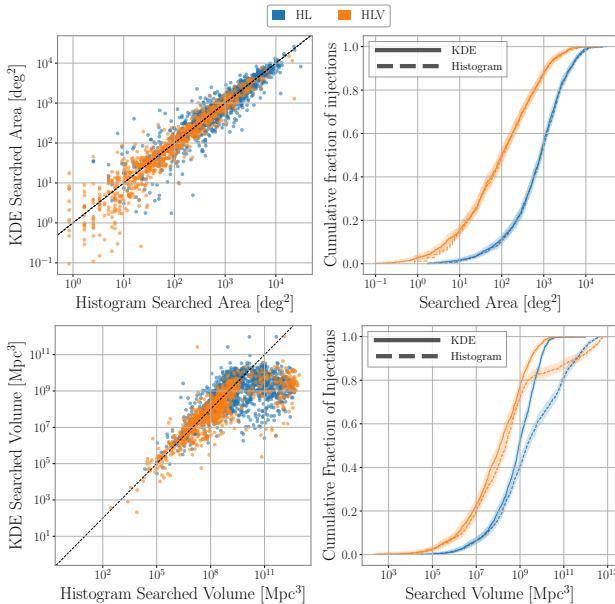

FIG. 2: Top Left: Scatter plot of searched areas comparing the histogram and KDE estimators for the HL (blue) and HLV (orange) injections. Top Right: Cumulative histogram of searched areas for the same events. Error bars correspond to 1σ confidence intervals calculated by bootstrapping are shown. Searched areas from the two density estimation methods are in statistical agreement. Bottom: Same as top, but for searched volumes. Searched volumes for the two methods broadly agree up until $\sim 10^9$ Mpc^3 for HLV and $\sim 10^8$ Mpc^3 for HL network , where the KDE method begins to outperform the histogram estimator.

~ 100 deg^2 [38]. In this regime, the histogram methods searched volume is comparable with the KDE.

VI. PERFORMANCE STUDIES

A. Sky Maps

1. Accuracy

We compare AMPLFI's sky localization performance with BAYESTAR using the dataset set described in Sec. IV. In Figure 3 we compare searched areas (top) and searched volumes (bottom) between AMPLFI and BAYESTAR sky maps for both HLV and HL datasets. Bootstrapped confidence intervals show the searched area cumulative histograms agree within 2σ. The searched volume cumulative histogram shows that AMPLFI provides smaller searched volumes than BAYESTAR. One drawback of searched area and volume as performance metrics is they do not capture sky map multi-modality (see Ref. [41] for a discussion). In Figure 4 we compare the angular offset between the sky maps maximum a pos-

terior pixel and the true pixel of the injection. We plot $\cos\delta\theta$ which highlights groupings at the true and antipodal location. We see agreement between the BAYESTAR and AMPLFI distributions.

2. Consistency

Searched area and volume measure accuracy, but not calibration (i.e. probabilistic consistency). Sky map calibration can be evaluated by performing probability-probability (P-P) tests on a dataset of simulated signals. For well calibrated sky maps, reported confidence intervals should correspond to the probability of finding the true location of the GW source in that area or volume region. This probability can be estimated empirically by analyzing injections.

For BAYESTAR, this calibration is achieved by rescaling the SNR timeseries by a correction factor which tuned to injections [13]. Notably, this correction factor is dependent on algorithmic details of the matched-filter pipeline that provides the SNR timeseries. The default value of 0.83 is tuned to injections analyzed by the GstLAL pipeline. Using this default correction factor, BAYESTAR sky maps produced from the SNR timeseries provided by the PyCBC live analysis have been shown to be biased (see Figure 5 in Ref. [42]). The source of this correction factor was further studied in Ref. [43] using PyCBC. The correction factor was shown to depend on variables like the construction of the matched-filtering template bank, and detector configuration. The variability of this correction factor upon pipeline specifics is undesirable, since it requires tuning to algorithmic choices.

Figure 7 shows self-consistency tests for the HLV AMPLFI model using injections with SNR ≥ 12. Consistency tests for area and volume are shown to be unbiased. A small bias in distance is present. Since AMPLFI is trained directly on strain data it does not require additional pipeline dependent tuning factors to achieve self consistency.

B. Intrinsic Parameter Recovery

In low-latency, matched-filter pipelines provide only point estimates of source parameters. These point estimates can suffer from systematic biases due to algorithmic choices such as template bank construction [44]. These biases have been shown to affect the performance of downstream source property classifiers which are used to publicly distribute probabilities of the source containing a neutron star, remnant, or mass gap component [14]. Because AMPLFI provides a full posterior, estimates of uncertainty can be used in low latency. These estimates can be used to restrict the prior space for downstream PE tasks which aim to explore the full signal parameter space including spins. Restricting the prior space leads to faster PE using fewer computational resources. In Fig. 5,
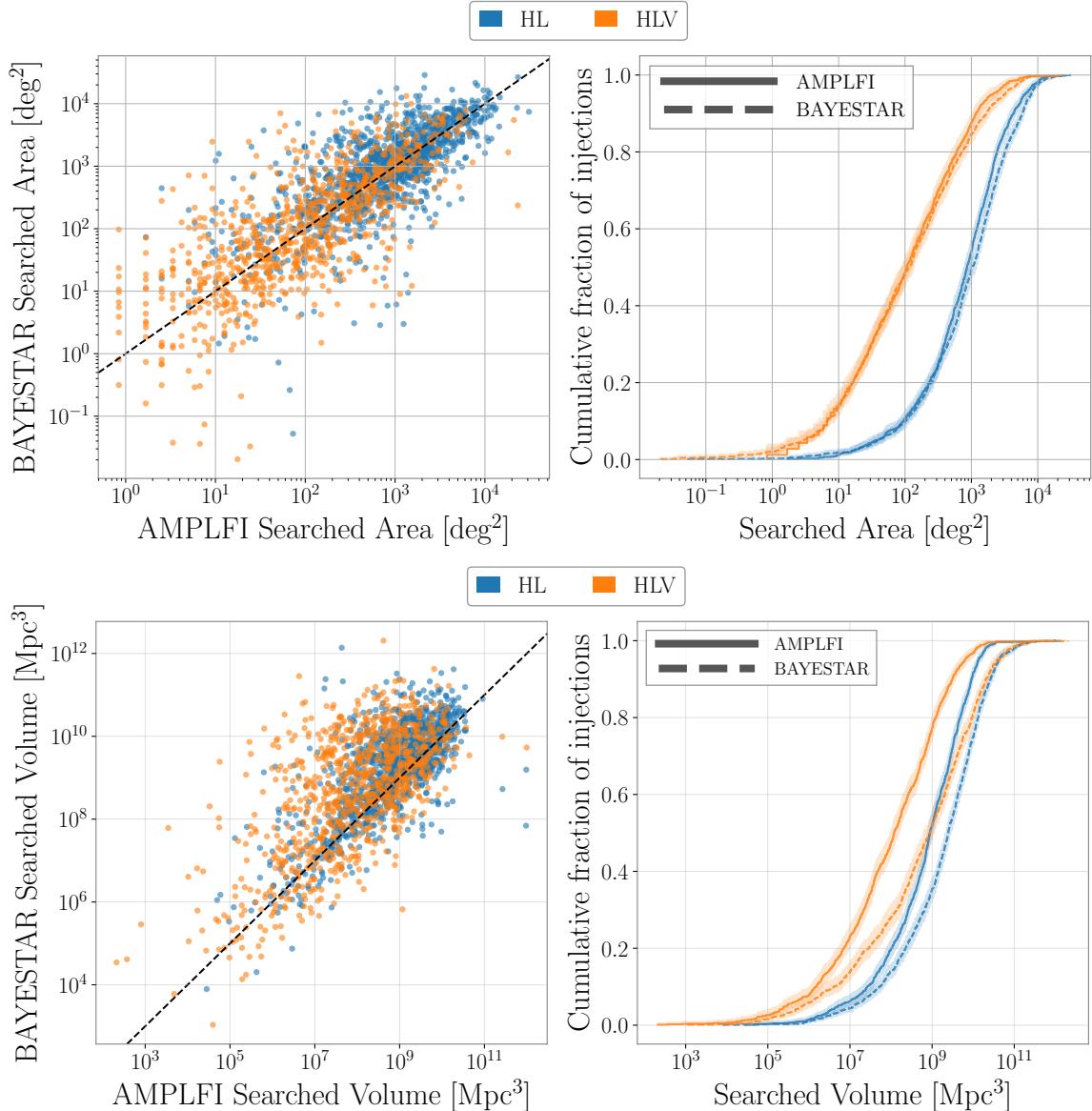
FIG. 3: Top Left: Scatter plot comparing searched areas of AMPLFI and BAYESTAR. HLV injections are show in orange, and HL injections are shown in blue. Top Right: Cumulative histogram of searched areas for the same events. Error bars correspond to 1σ confidence intervals calculated by bootstrapping are shown. The cumulative distribution of searched areas from the two methods are statistically equivalent. Bottom: Same as top, but for searched volumes. AMPLFI provides smaller searched volumes than BAYESTAR

we illustrate AMPLFI's ability to recover the chirp mass of the source. We see the expected increase in uncertainty as the injected chirp mass increases. In addition, we perform a P-P test using 500 injections drawn from AMPLFI's training prior. Fig. 6 shows a P-P test demonstrating AMPLFI's unbiased recovery of parameters.

C. Model Robustness

GW interferometers have non-stationary noise distributions [45, 46]. Changing noise distributions pose problems to machine learning algorithms, which can suffer from overfitting to the noise statistics of the data period used for training. For production grade deployment, it is critical to validate the robustness of model performance beyond the initial training period, and the length of time over which satisfactory performance is maintained.

Different strategies can be employed to construct a ro-

FIG. 4: Normalized histogram of $\cos \delta\theta$ for the HLV injection set, where $\delta\theta$ is the angular offset between the true pixel of the injection and the maximum a posterior pixel

FIG. 5: Scatter plot comparing true chirp mass and AMPLFI's median chirp mass for simulated signals from the injection dataset described in Sec. IV. Error bars corresponding to the 5$^{\text{th}}$ and 95$^{\text{th}}$ percentiles of AMPLFI's posterior samples are shown. Orange and blue points correspond to the HL and HLV networks respectively.

bust noise model. For example, in Ref. [47], data augmentations applied to a fiducial PSD are utilized to construct a noise model that captures the variation in noise properties across an observing run. In that work, the augmented PSDs are used to generate synthetic Gaussian noise for training. As described in Sec III, AMPLFI utilizes real detector strain data for training, and so our

noise model is constructed empirically. Specifically, our training data consists of T disjoint, coincident, science quality segments $\{[a_i, b_i]\}_{i=1}^T$. Let l denote the length of each training sample provided to the neural network, and s denote the length of data used for PSD estimation. The algorithm for sampling strain data instances during training is as follows. For each detector k,

1. Sample segment index i_k, with probability proportional to the segments duration:

$$p(i_k) = \frac{b_{i_k} - a_{i_k}}{\sum_{j=1}^T (b_j - a_j)}$$

2. Sample time $\tau_k \sim \text{Unif}([a_{i_k} + s, b_{i_k+1} - l])$

3. Select strain data for detector k from $\tau_k - s$ to $\tau_k + l$

This procedure is repeated N times, where N is the size of the training batch.

The idea behind this sampling procedure is that noise between different detector sites is highly uncorrelated. So, strain data sampled independently in time for each detector is a plausible, unique noise realization. This idea, well-known in the GW literature as time-slides, is commonly used to create realistic noise realizations for estimating the background distributions of search algorithms. During training, the detector strain data is stored on disk and the above sampling procedure is done on-the-fly.

To validate AMPLFI's robustness over time, we created 4 datasets of 1000 injections sampled from AMPLFI's training prior and added them into real detector noise. For each dataset, we utilized strain data at different epochs after the initial model training period. These periods spanned from immediately after, to 11 weeks after the training period. Each period consisted of ~ 1 day of live time. We analyze each of these datasets using the HLV AMPLFI network. In Fig 8, we show sky localization consistency and searched area performance. We see that AMPLFI's sky localizations remain well calibrated and accurate 11 weeks beyond AMPLFI's initial training period. This demonstrates that a single AMPLFI model maintains utility well beyond its initial training period.

D. Analysis of GWTC-3 Events

Several GW candidates with probability of astrophysical origin greater than 0.5 were identified in the third Gravitational-Wave Transient Catalog (GWTC-3) [4]. We analyze those candidates with posterior support within AMPLFI's training prior (Table II), and which occurred during or after the period of data used to train AMPLFI. In addition, we do not analyze candidates which required glitch mitigation. After these constraints, this leaves 5 candidates analyzed with the HLV network (Figures 9 to 13), and 3 candidates analyzed with the HL network (Figures 14 to 16). We compare posteriors and
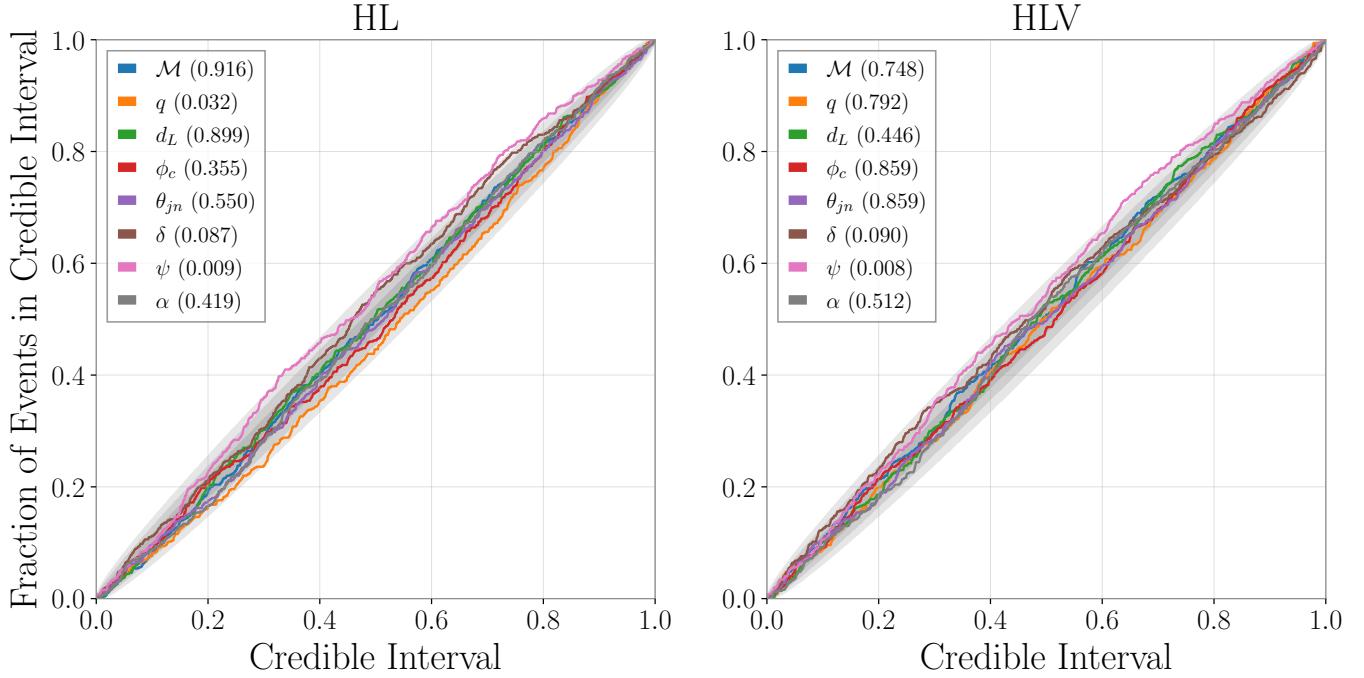
FIG. 6: P-P tests for AMPLFI's 1 dimensional marginal posterior distributions. 500 injections from AMPLFI's training prior were analyzed. The p-value for each parameter is provided in the legend. Shaded bands correspond to 1, 2 and 3 σ confidence intervals.
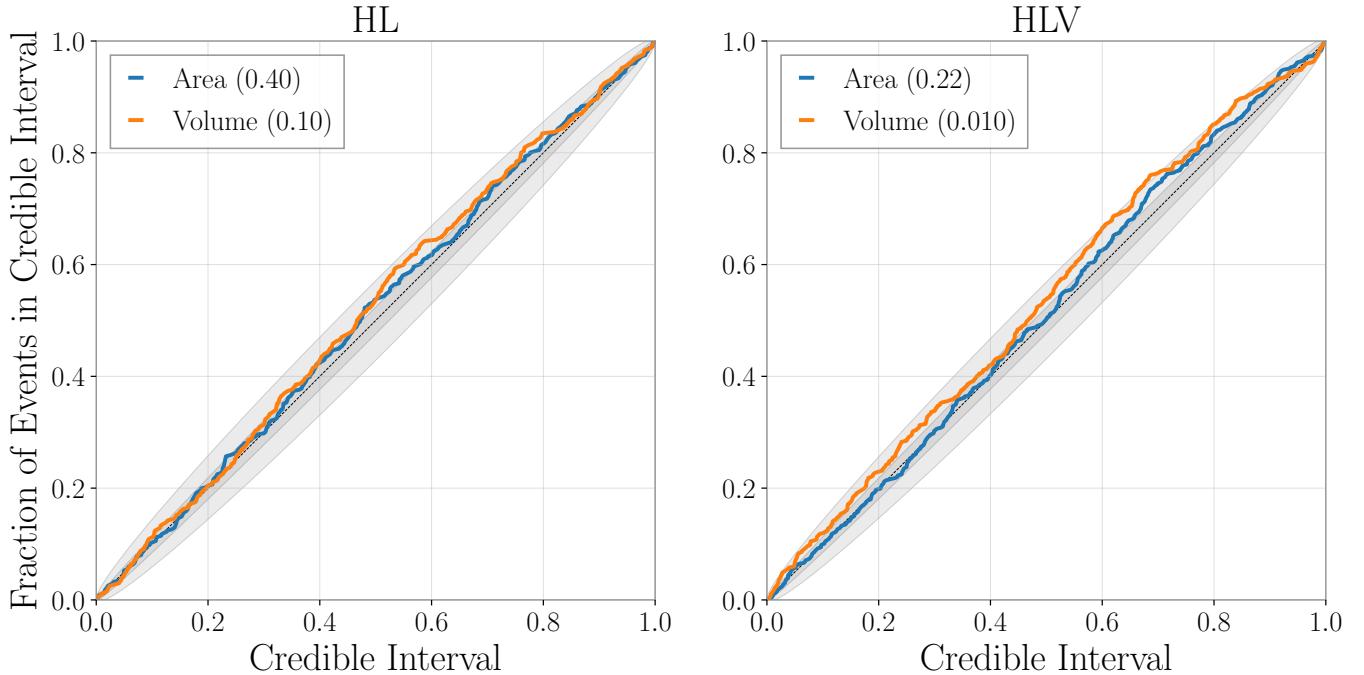

FIG. 7: P-P tests for AMPLFI's joint 2D (area) and 3D (volume) posterior distributions measuring sky map self consistency. The p-value for each consistency check is provided in the legend. Error bars corresponding to 1 and 3 σ confidence intervals are plotted as shaded bands.

FIG. 8: Left: 2D (area) localization P-P tests for 4 datasets of 1000 injections drawn from AMPLFI's training prior created at various epochs across O3b. P-values shown in the legend indicate that sky maps produced by AMPLFI remain well calibrated across an 11 week period. Right: Cumulative histogram of searched area for the same datasets. Bootstrapped 95% confidence intervals are plotted as shaded bands. The accuracy of AMPLFI sky maps does not degrade over time.

sky maps with those produced by the GWTC-3 analysis using samples from the IMRPhenomXPHM waveform. Where available, we also compare the low-latency sky map produced by BAYESTAR. Analyzing each HLV (HL) event takes ~ 2 (1.5) seconds using an NVIDIA A30 GPU.

For several candidates, source mass posteriors from the GWTC-3 analysis were completely within AMPLFI's training prior, but luminosity distance support only partially overlapped. We analyze these events, but note that AMPLFI's luminosity distance posterior rails against the prior. Still, AMPLFI's sky maps and posterior estimates for source parameters are consistent with the LVK analysis. See Figures 17 to 19 for examples of such events. A possible solution to this problem would be to train AMPLFI to infer the chirp distance ($d_{\rm chirp} \propto d_L/\mathcal{M}^{5/6}$) [48, 49] instead of the luminosity distance. Effectively, this adjusts the distance prior depending on the chirp mass of the source, allowing the recovery of intrinsically louder signals at farther distances. Exploring this is left to future work.

There exist several differences between the GWTC-3 and AMPLFI analyses. First, AMPLFI trains using a distance prior that is uniform, while the GWTC-3 analyses utilize a uniform in co-moving volume distribution. To account for this, we use importance sampling to re-weight AMPLFI's samples to the uniform in co-moving volume distance distribution reported by the GWTC-3 analysis. Second, AMPLFI was trained using the IMRPhenomPV2 waveform, while the LVK posteriors were produced using IMRPhenomXPHM. So, effects of higher order modes are not captured in AMPLFI's results. Third, the AMPLFI result was produced using the low-latency data, while the GWTC-3 result utilized data that includes an update to calibration, and cleaning of certain noise sources. Lastly, AMPLFI does not aim to explicitly model the Gaussian likelihood typically assumed in CBC PE. By training with real data, non-Gaussian statistics in the data is learned.

In general, the posteriors show close agreement. However, there are some instances with noticeable differences. For example, AMPLFI does not show multimodality in masses exhibited by the GWTC-3 result in GW200129_065458 (see Figure 10). For this same event, AMPLFI's distance and inclination estimates disagree with the GWTC-3 result. With an SNR of $\sim$ 27, waveform systematics become more important and could explain this discrepancy. For GW191215_223052 and GW200220_124850 AMPLFI does not show multimodality in inclination exhibited by the GWTC-3 result (see Figures 9 and 17). Generally, possible sources of the differences between AMPLFI and GWTC-3 results could be the physics of the assumed waveform (e.g. the inclusion of higher order modes in the GWTC-3 result), suboptimal model convergence and expressivity, or the difference in assumed noise statistics (i.e. GWTC-3 PE assumes Gaussian noise, whereas AMPLFI is trained using real data). We note that through importance sampling to a Gaussian likelihood, it is possible to mitigate differences due to suboptimal model convergence and assumed noise statistics [50].

VII. CONCLUSION AND FUTURE WORK

We have presented the performance of the AMPLFI algorithm across several metrics. We have demonstrated AMPLFI's sky localization performance is equivalent to BAYESTAR using injections from an online data replay. We have shown that a single AMPLFI algorithm trained using ~ 2 months of real data maintains performance months beyond its training period. Finally, we analyzed real gravitational wave candidates within AMPLFI's training prior and have shown posterior results consistent with the GWTC-3 analyses.

Still, there remains several avenues for improving the utility of AMPLFI. Broadening the parameter space to include neutron star black hole and and binary neutron star (BNS) mergers will be critical for analyzing more likely candidates of EM emission. For BNS signals that can last for minutes in the detectors' sensitive band, utilizing data compression methods will be important. In addition, exploring parameterizations of the CBC signal parameter space that eliminate known degeneracies could help the learning process [48]. For example, training AMPLFI to infer chirp distance [49] instead of luminosity distance could help alleviate the distance prior railing exhibited in AMPLFI's analysis of several real candidates.

As of August 2025, AMPLFI has been deployed in production for real-time follow-up of CBC candidates detected by the Aframe search algorithm, and has contributed to several public alerts[9].

VIII. ACKNOWLEDGMENTS

———

[9] https://gracedb.ligo.org/superevents/public/O4c/

[1] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, et al. (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. Lett. **116**, 061102 (2016), URL https://link.aps.org/doi/10.1103/PhysRevLett.116.061102.

[2] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, et al. (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. X **9**, 031040 (2019), URL https://link.aps.org/doi/10.1103/PhysRevX.9.031040.

[3] R. Abbott et al. (LIGO Scientific, VIRGO), Phys. Rev. D **109**, 022001 (2024), 2108.01045.

[4] R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, N. Adhikari, R. Adhikari, V. Adya, C. Affeldt, D. Agarwal, et al., Physical Review X **13** (2023), ISSN 2160-3308, URL http://dx.doi.org/10.1103/PhysRevX.13.041039.

[5] A. G. Abac et al. (LIGO Scientific, VIRGO, KAGRA) (2025), 2508.18082.

[6] The LIGO Scientific Collaboration, J. Aasi, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, K. Ackley, C. Adams, T. Adams, P. Addesso, et al., Classical and Quantum Gravity **32**, 074001 (2015), URL https://dx.doi.org/10.1088/0264-9381/32/7/074001.

[7] F. Acernese et al. (VIRGO), Class. Quant. Grav. **32**, 024001 (2015), 1408.3978.

[8] T. Akutsu, M. Ando, K. Arai, Y. Arai, S. Araki, A. Araya, N. Aritomi, Y. Aso, S. Bae, Y. Bae, et al., Progress of Theoretical and Experimental Physics **2021**, 05A101 (2020), ISSN 2050-3911, https://academic.oup.com/ptep/article-pdf/2021/5/05A101/37974994/ptaa125.pdf, URL https://doi.org/10.1093/ptep/ptaa125.

[9] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya, C. Affeldt, M. Agathos, et al., Living Reviews in Relativity **23** (2020), ISSN 1433-8351, URL http://dx.doi.org/10.1007/s41114-020-00026-9.

[10] E. Marx et al., Phys. Rev. D **111**, 042010 (2025), 2403.18661.

[11] N. Nagarajan and C. Messenger, arXiv preprint arXiv:2501.13846 (2025).

[12] P. Nousi, A. E. Koloniari, N. Passalis, P. Iosif, N. Stergioulas, and A. Tefas, Physical Review D **108** (2023), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.108.024022.

[13] L. P. Singer and L. R. Price, Physical Review D **93** (2016), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.93.024013.

[14] D. Chatterjee, S. Ghosh, P. R. Brady, S. J. Kapadia, A. L. Miller, S. Nissanke, and F. Pannarale, Astrophys. J. **896**, 54 (2020), 1911.00116.

[15] D. Chatterjee et al., Mach. Learn. Sci. Tech. **5**, 045030 (2024), 2407.19048.

[16] G. Ashton et al., Astrophys. J. Suppl. **241**, 27 (2019), 1811.02042.

[17] J. Veitch, V. Raymond, B. Farr, W. Farr, P. Graff, S. Vitale, B. Aylott, K. Blackburn, N. Christensen, M. Coughlin, et al., Physical Review D **91** (2015), ISSN 1550-2368, URL http://dx.doi.org/10.1103/PhysRevD.91.042003.

[18] P. Canizares, S. E. Field, J. Gair, V. Raymond, R. Smith, and M. Tiglio, Phys. Rev. Lett. **114**, 071104 (2015), 1404.6284.

[19] S. Morisaki, R. Smith, L. Tsukada, S. Sachdev, S. Stevenson, C. Talbot, and A. Zimmerman (2023), 2307.13380.

[20] S. Morisaki and V. Raymond, Phys. Rev. D **102**, 104020 (2020), 2007.09108.

[21] L. P. Singer and L. R. Price, Phys. Rev. D **93**, 024013 (2016), URL https://link.aps.org/doi/10.1103/PhysRevD.93.024013.

[22] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Journal of Machine Learning Research **22**, 1 (2021).

[23] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Physical Review Letters **127** (2021), ISSN 1079-7114, URL http://dx.doi.org/10.1103/PhysRevLett.127.241103.

[24] M. Dax, S. R. Green, J. Gair, N. Gupte, M. Pürrer, V. Raymond, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, Nature **639**, 49–53 (2025), ISSN 1476-4687, URL http://dx.doi.org/10.1038/s41586-025-08593-z.

[25] M. J. Williams, J. Veitch, and C. Messenger, Phys. Rev. D **103**, 103006 (2021), 2102.11056.

[26] F. De Santi, M. Razzano, F. Fidecaro, L. Muccillo, L. Papalini, and B. Patricelli, Phys. Rev. D **109**, 102004 (2024), URL https://link.aps.org/doi/10.1103/PhysRevD.109.102004.

[27] E. C. Bellm, S. R. Kulkarni, M. J. Graham, R. Dekany, R. M. Smith, R. Riddle, F. J. Masci, G. Helou, T. A. Prince, S. M. Adams, et al., Publications of the Astronomical Society of the Pacific **131**, 018002 (2018), ISSN 1538-3873, URL http://dx.doi.org/10.1088/1538-3873/aaecbe.

[28] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. J. Forteza, and A. Bohé, Phys. Rev. D **93**, 044007 (2016), URL https://link.aps.org/doi/10.1103/PhysRevD.93.044007.

[29] K. He, X. Zhang, S. Ren, and J. Sun, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[30] U. Bhardwaj, J. Alvey, B. K. Miller, S. Nissanke, and C. Weniger, Phys. Rev. D **108**, 042004 (2023), 2304.02035.

[31] T.-Y. Sun, C.-Y. Xiong, S.-J. Jin, Y.-X. Wang, J.-F. Zhang, and X. Zhang, Chin. Phys. C **48**, 045108 (2024), 2312.08122.

[32] L. Dinh, J. Sohl-Dickstein, and S. Bengio, *Density estimation using real nvp* (2017), 1605.08803, URL https://arxiv.org/abs/1605.08803.

[33] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Advances in neural information processing systems **32** (2019).

[34] F. Rozet et al., *Zuko: Normalizing flows in pytorch* (2022), URL https://pypi.org/project/zuko.

[35] S. S. Chaudhary et al., Proc. Nat. Acad. Sci. **121**, e2316474121 (2024), 2308.04545.

[36] A. Zonca, L. Singer, D. Lenz, M. Reinecke, C. Rosset, E. Hivon, and K. Gorski, Journal of Open Source Software **4**, 1298 (2019), URL https://doi.org/10.21105/joss.01298.

[37] Singer et al., The Astrophysical Journal Letters **829**, L15 (2016), URL http://stacks.iop.org/2041-8205/829/i=1/a=L15.

[38] I. Andreoni, R. Margutti, J. Banovetz, S. Greenstreet, C.-A. Hebert, T. Lister, A. Palmese, S. Piranomonte, S. Smartt, G. P. Smith, et al., arXiv preprint arXiv:2411.04793 (2024).

[39] L. P. Singer, H.-Y. Chen, D. E. Holz, W. M. Farr, L. R. Price, V. Raymond, S. B. Cenko, N. Gehrels, J. Cannizzo, M. M. Kasliwal, et al., The Astrophysical Journal Supplement Series **226**, 10 (2016), URL https://dx.doi.org/10.3847/0067-0049/226/1/10.

[40] P. Fernique, T. Boch, T. Donaldson, D. Durand, W. O'Mullane, M. Reinecke, and M. Taylor (2014), URL http://dx.doi.org/10.5479/ADS/bib/2014ivoa.spec.0602F.

[41] R. Essick, S. Vitale, E. Katsavounidis, G. Vedovato, and S. Klimenko, Astrophys. J. **800**, 81 (2015), 1409.2435.

[42] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, Physical Review D **98** (2018), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.98.024050.

[43] P.-A. Duverne, S. Hoang, T. Dal Canton, S. Antier, N. Arnaud, P. Hello, and F. Pannarale, Physical Review D **110** (2024), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.110.102002.

[44] B. Ewing et al., Phys. Rev. D **109**, 042008 (2024), 2305.05625.

[45] D. Davis, J. S. Areeda, B. K. Berger, R. Bruntz, A. Effler, R. C. Essick, R. P. Fisher, P. Godwin, E. Goetz, A. F. Helmling-Cornell, et al., Classical and Quantum Gravity **38**, 135014 (2021), URL https://dx.doi.org/10.1088/1361-6382/abfd85.

[46] S. Soni et al. (LIGO), Class. Quant. Grav. **42**, 085016 (2025), 2409.02831.

[47] J. Wildberger, M. Dax, S. R. Green, J. Gair, M. Pürrer, J. H. Macke, A. Buonanno, and B. Schölkopf, Physical Review D **107** (2023), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.107.084046.

[48] J. Roulet, S. Olsen, J. Mushkin, T. Islam, T. Venumadhav, B. Zackay, and M. Zaldarriaga, Physical Review D **106** (2022), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.106.123015.

[49] P. R. Brady and S. Fairhurst, Classical and Quantum Gravity **25**, 105002 (2008), ISSN 1361-6382, URL http://dx.doi.org/10.1088/0264-9381/25/10/105002.

[50] M. Dax, S. R. Green, J. Gair, M. Pürrer, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, Physical Review Letters **130** (2023), ISSN 1079-7114, URL http://dx.doi.org/10.1103/PhysRevLett.130.171403.
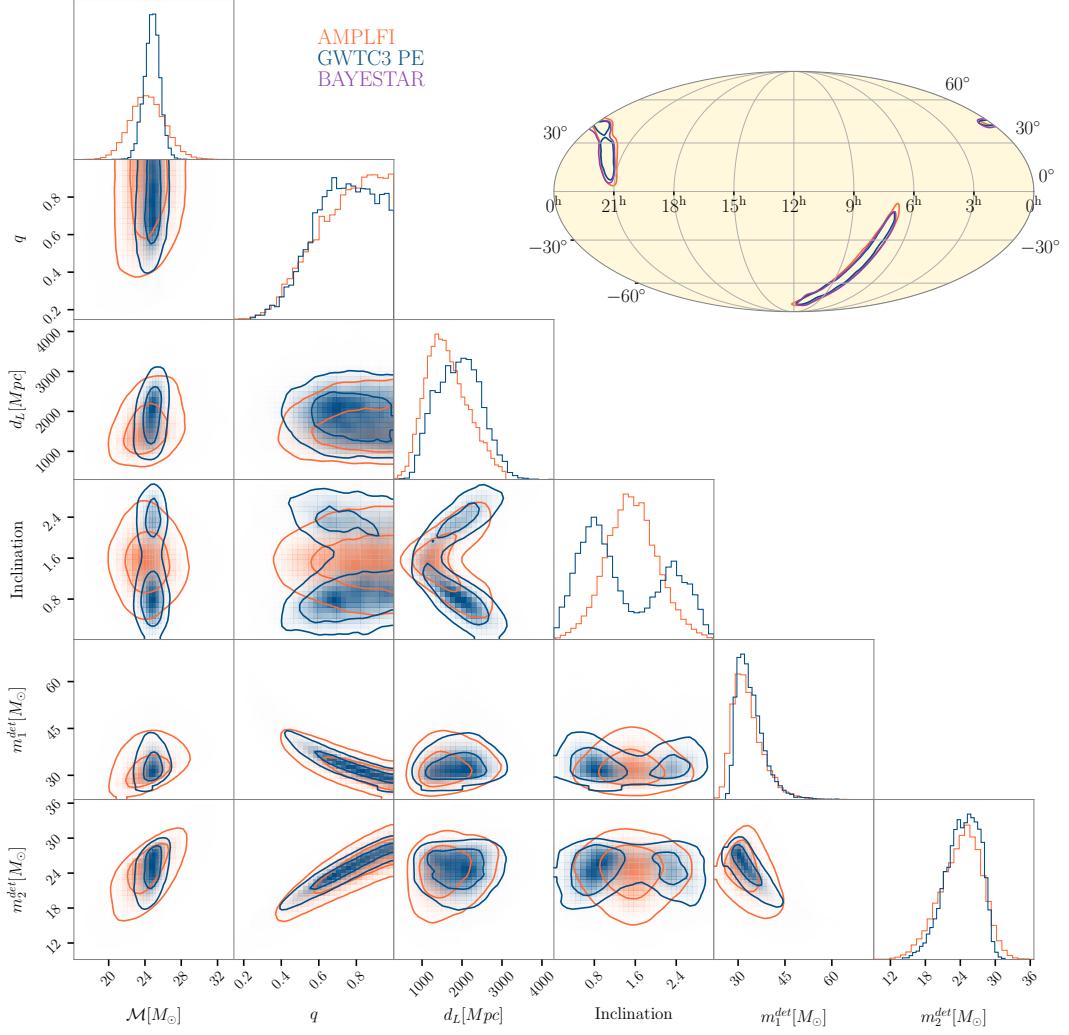
FIG. 9: Posterior comparison between AMPLFI and GWTC-3 result for GW191215_223052. Hanford, Livingston and Virgo data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. Sky map contours correspond to 90% confidence intervals.
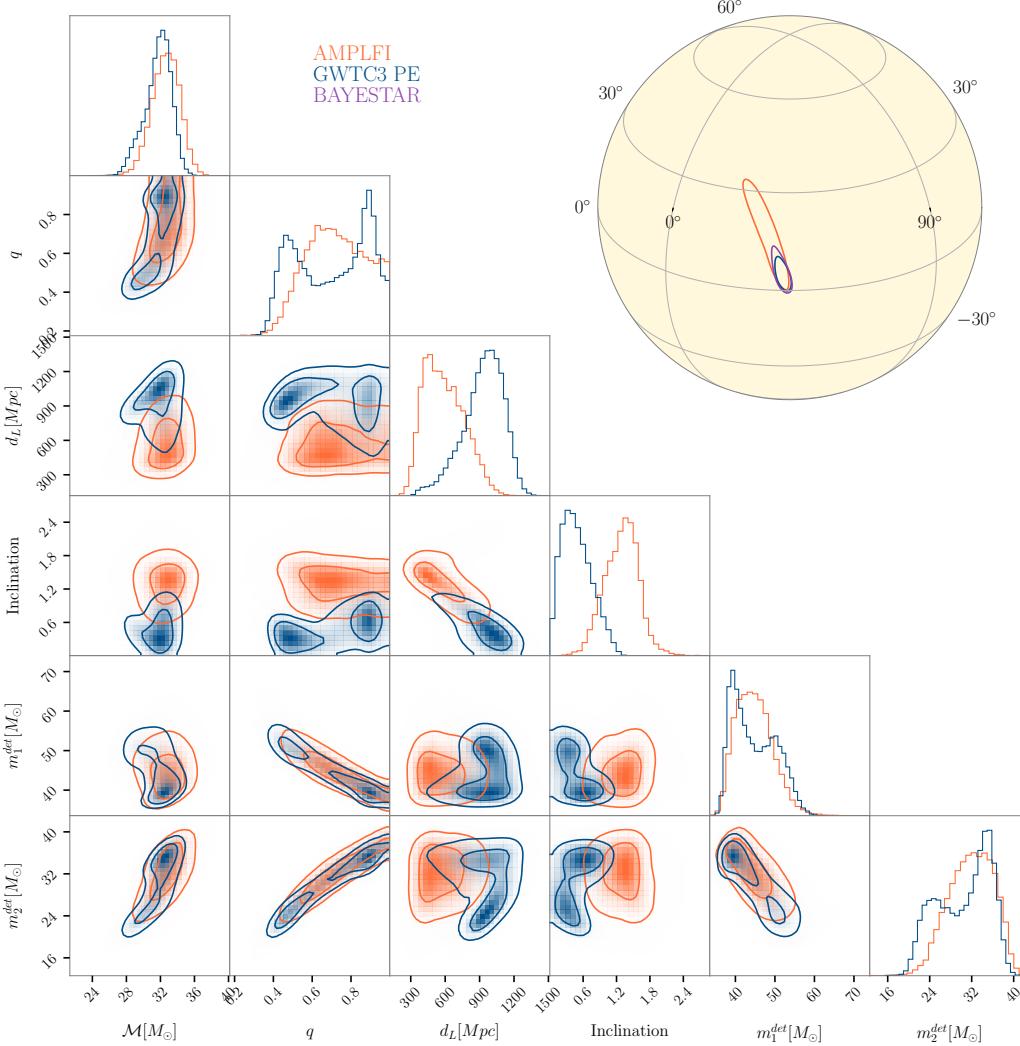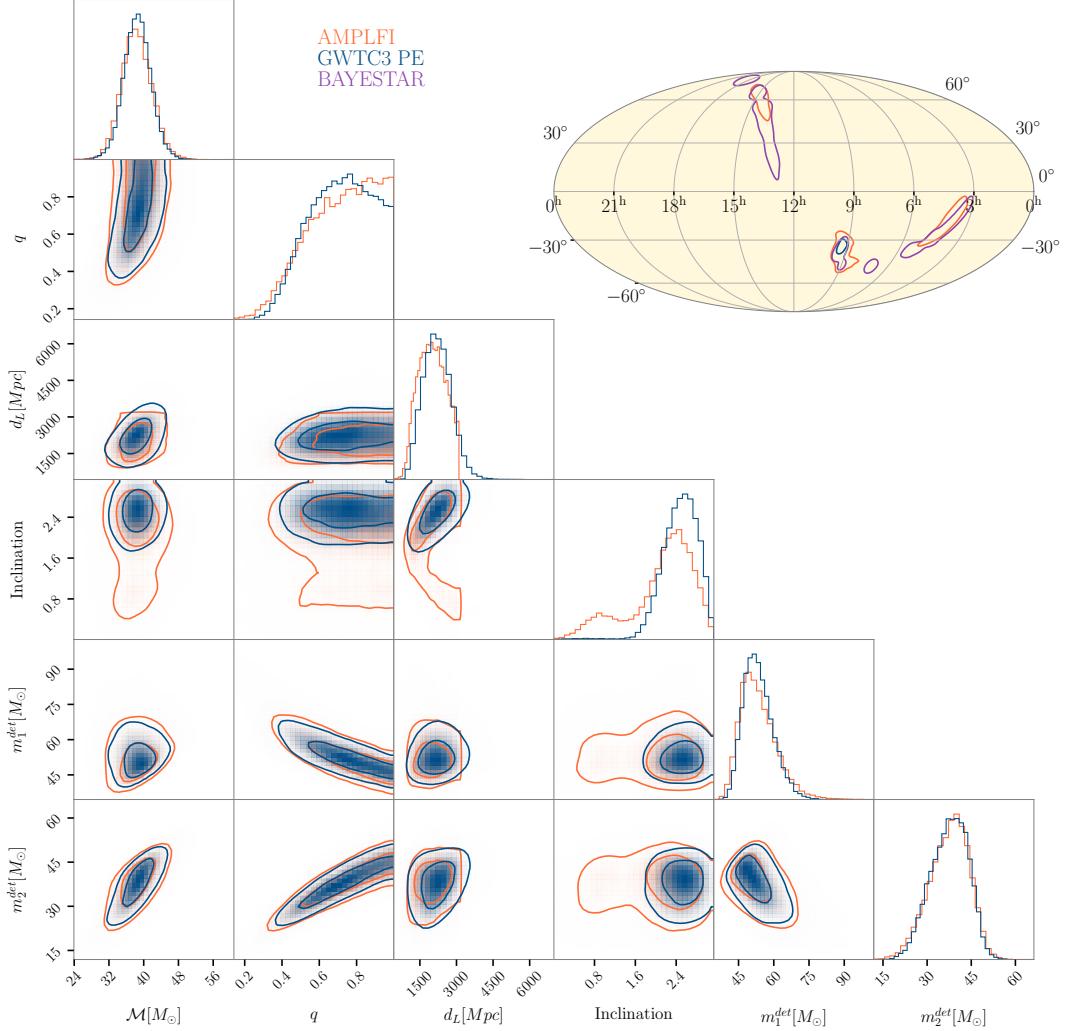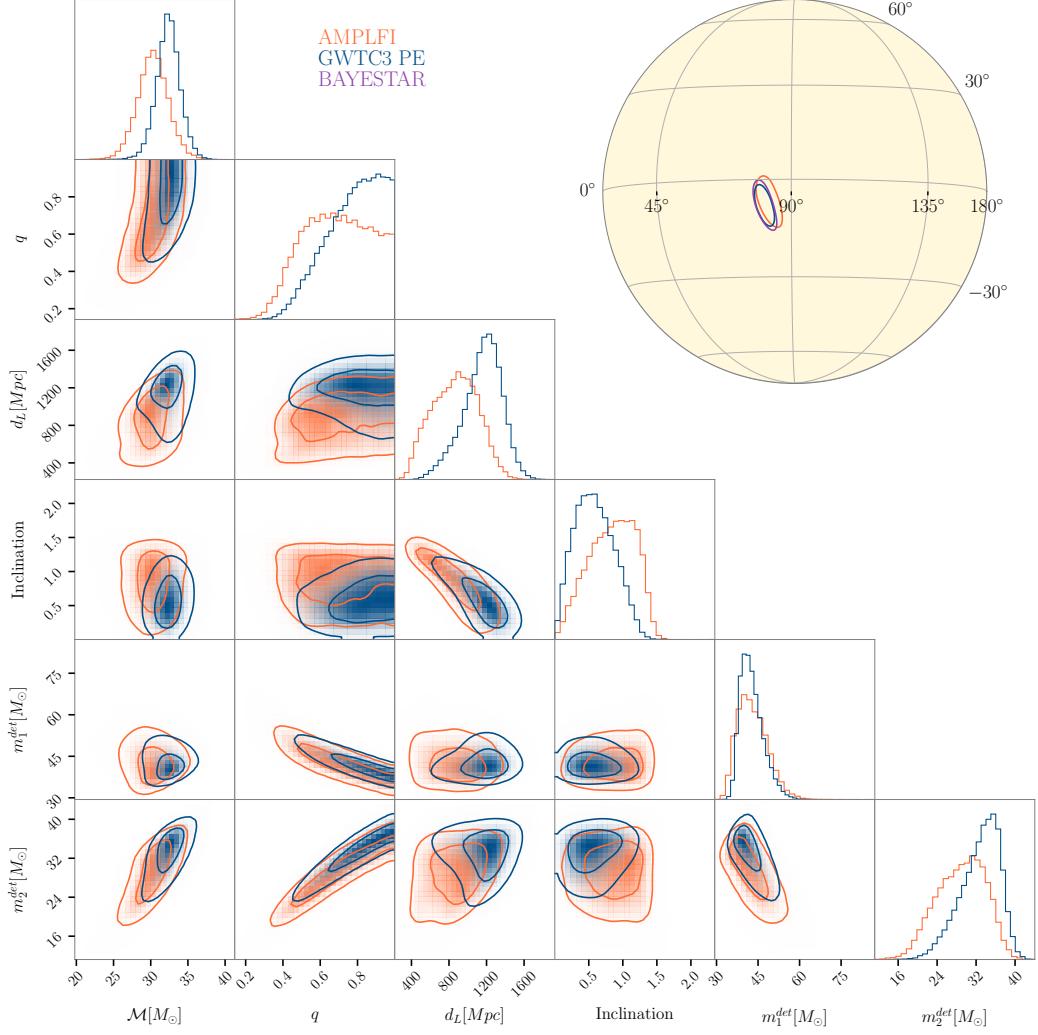
FIG. 10: Posterior comparison between AMPLFI and GWTC-3 result for GW200129_065458. Hanford, Livingston and Virgo data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. Sky map contour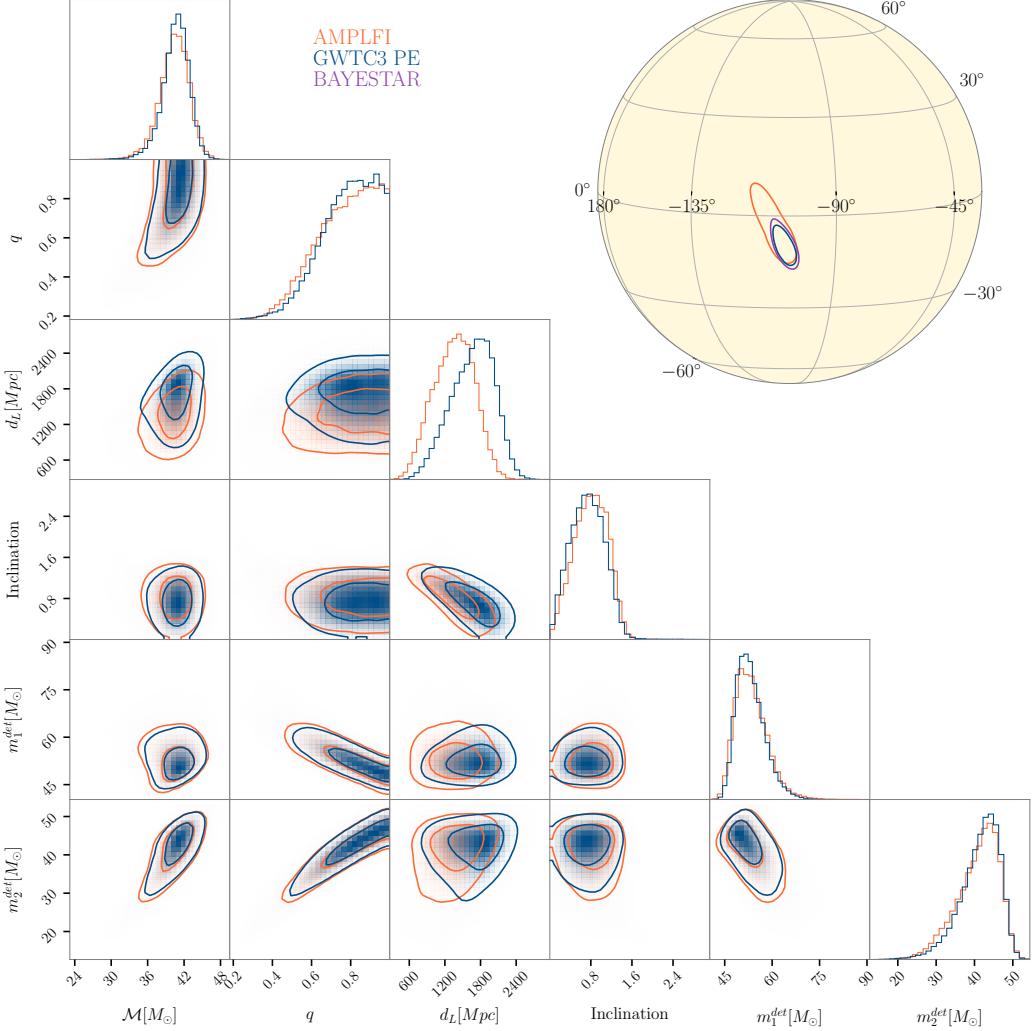s correspond to 90% confidence intervals. For this event, bi-modalities in m_1 and m_2 are not captured by AMPLFI. In addition, AMPLFI's distance and inclination posteriors disagree with the GWTC-3 result.

FIG. 11: Posterior comparison between AMPLFI and GWTC-3 result for GW200208_130117. Hanford, Livingston and Virgo data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. Sky map contours correspond to 90% confidence intervals.
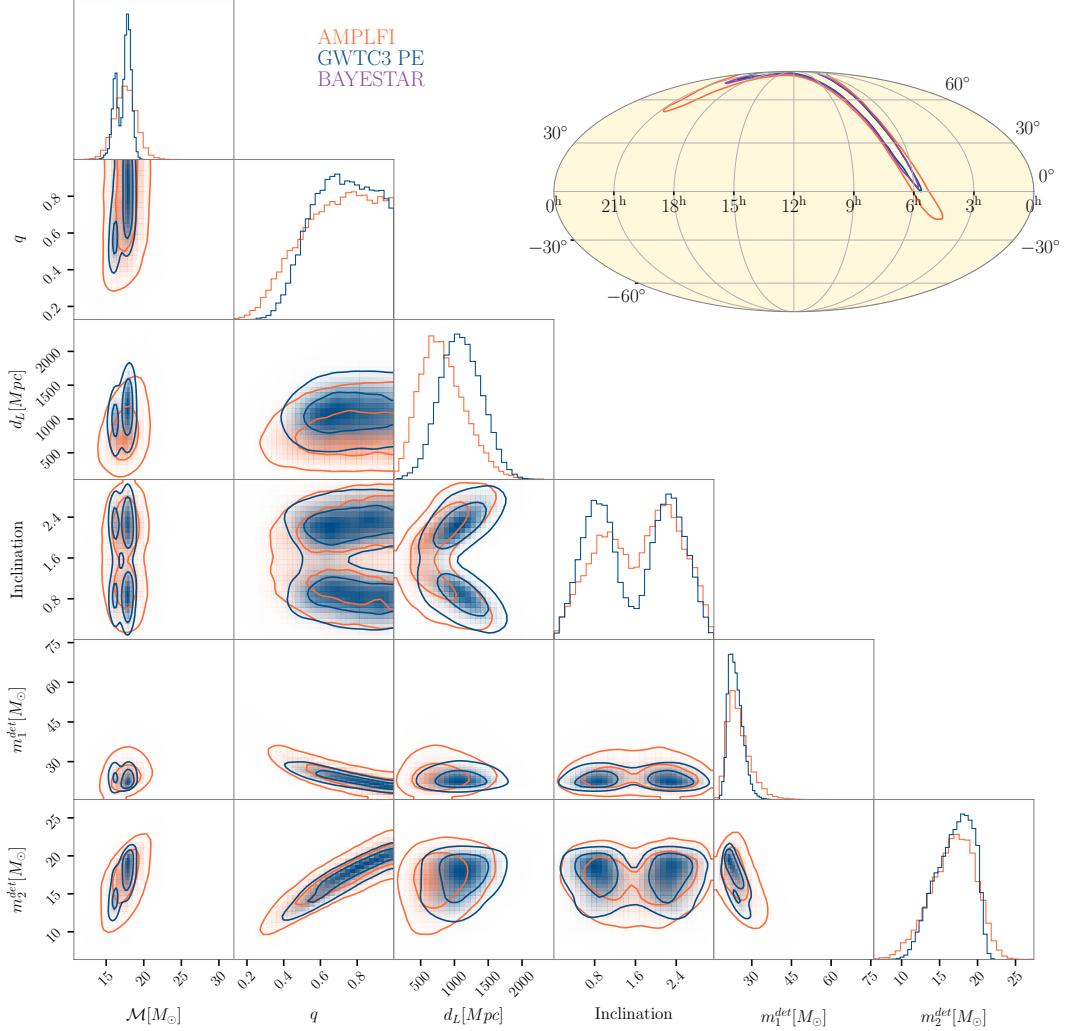
FIG. 12: Posterior comparison between AMPLFI and GWTC-3 result for GW200311_115853. Hanford, Livingston and Virgo data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. Sky map contours correspond to 90% confidence intervals.

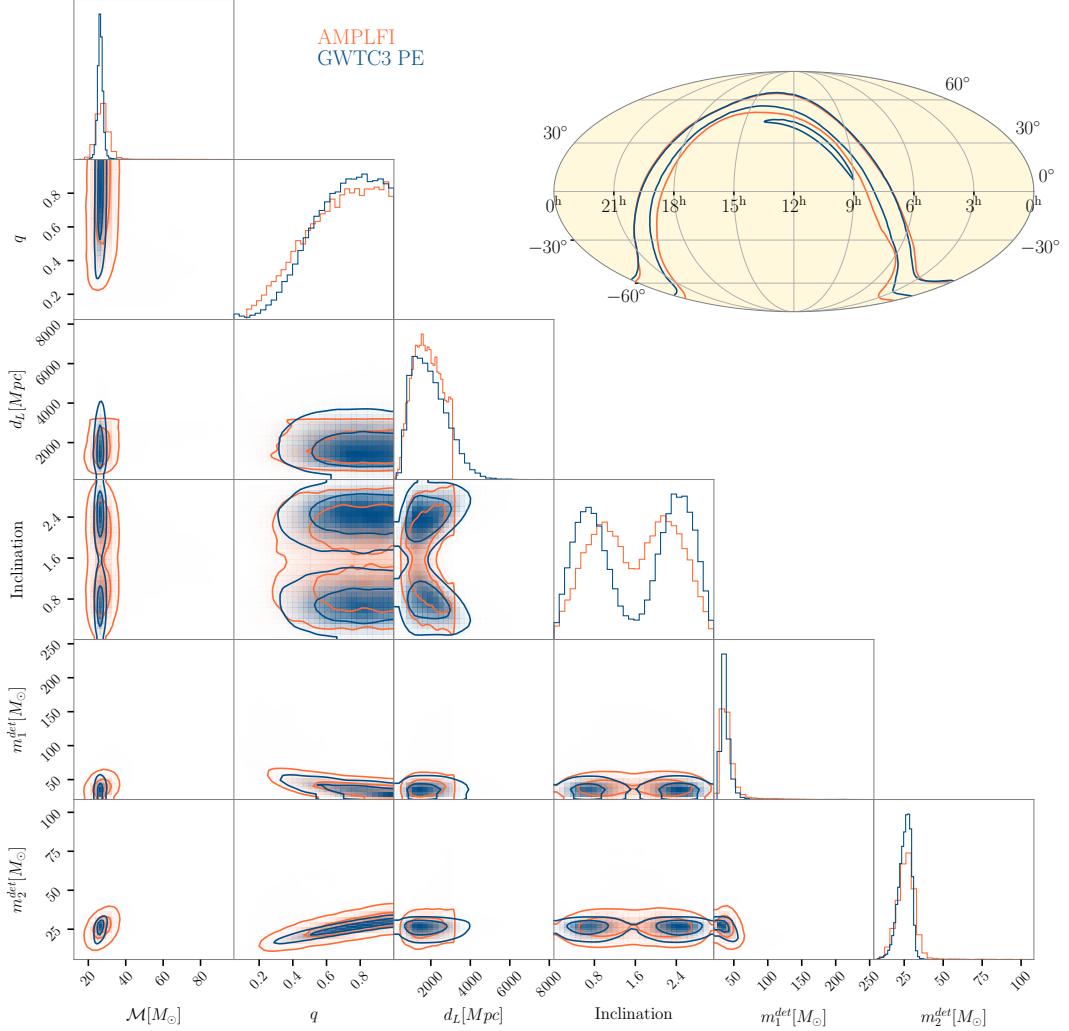FIG. 13: Posterior comparison between AMPLFI and GWTC-3 result for GW200224_222234. Hanford, Livingston and Virgo data is analyzed.The BAYESTAR sky map produced in low-latency is plotted in purple. Sky map contours correspond to 90% confidence intervals.

FIG. 14: Posterior comparison between AMPLFI and GWTC-3 result for GW200225_060421. Hanford and Livingston data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. Sky map contours correspond to 90% confidence intervals.

FIG. 15: Posterior comparison between AMPLFI and GWTC-3 result for GW191204_110529. Hanford and Livingston data is analyzed. Sky map contours correspond to 90% confidence intervals.
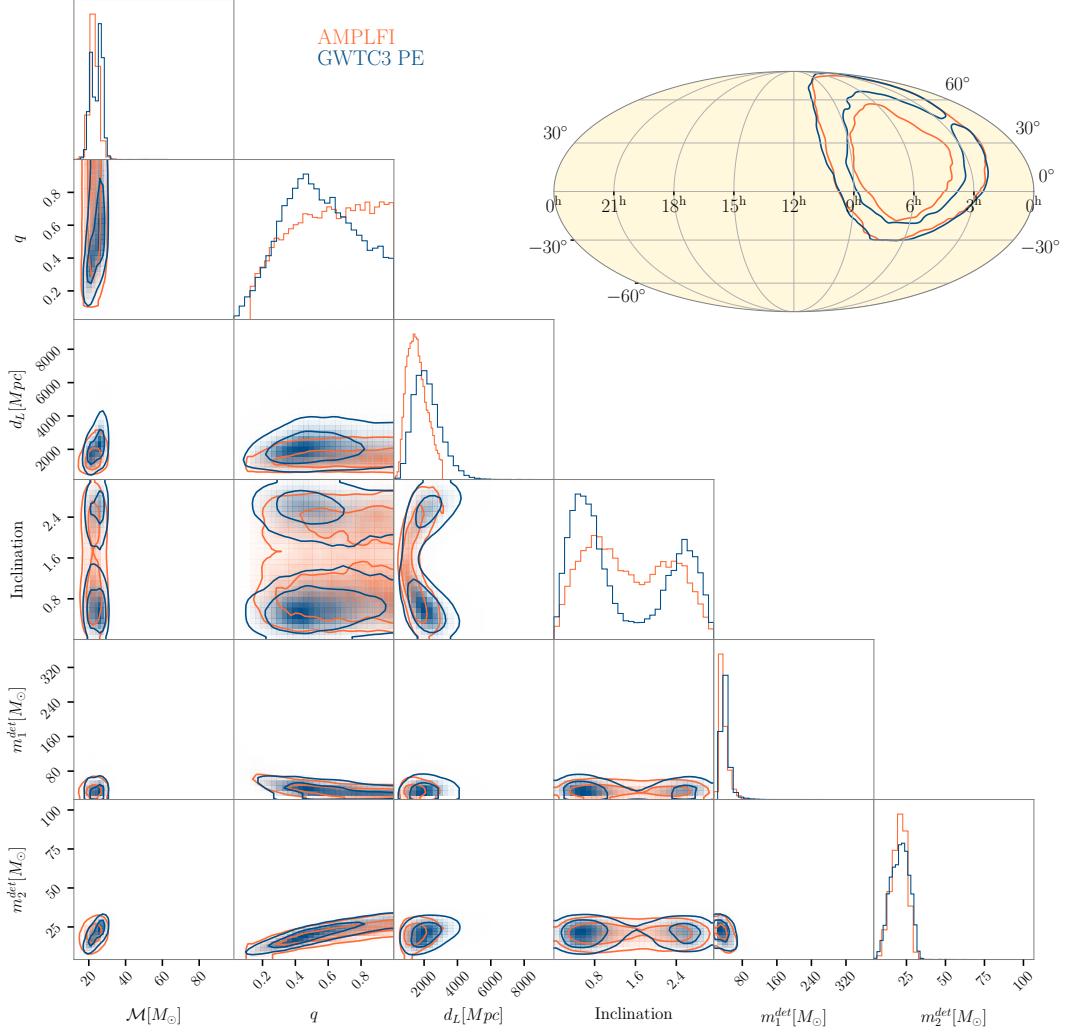
FIG. 16: Posterior comparison between AMPLFI and GWTC-3 result for GW200306_093714. Hanford and Livingston data is analyzed. Sky map contours correspond to 90% confidence intervals.
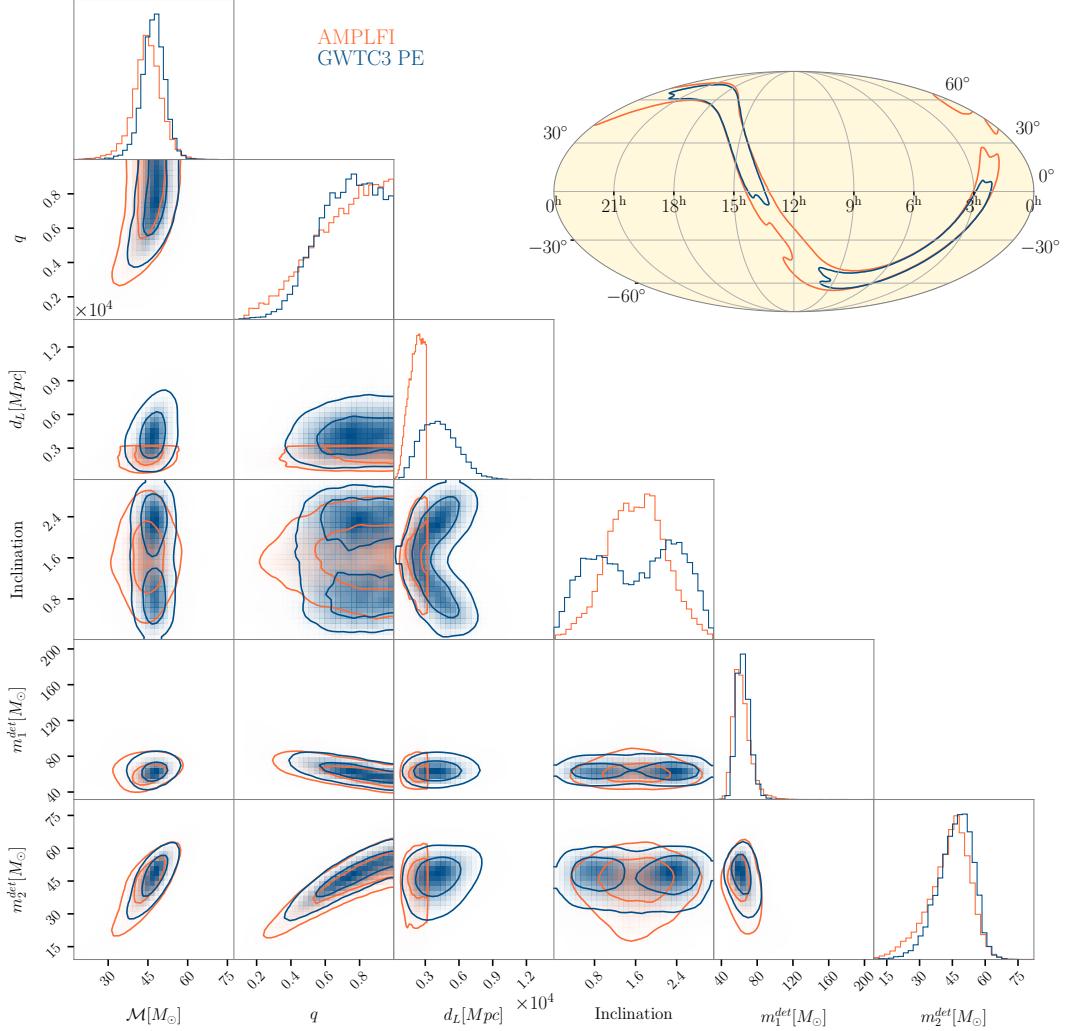
FIG. 17: Posterior comparison between AMPLFI and GWTC-3 result for GW200220_124850. Hanford and Livingston data is analyzed. This illustrates the scenario where the distance posterior has support well beyond AMPLFI's training prior. Still, AMPLFI is able to recover source parameters and localizations consistent with the GWTC-3 result.
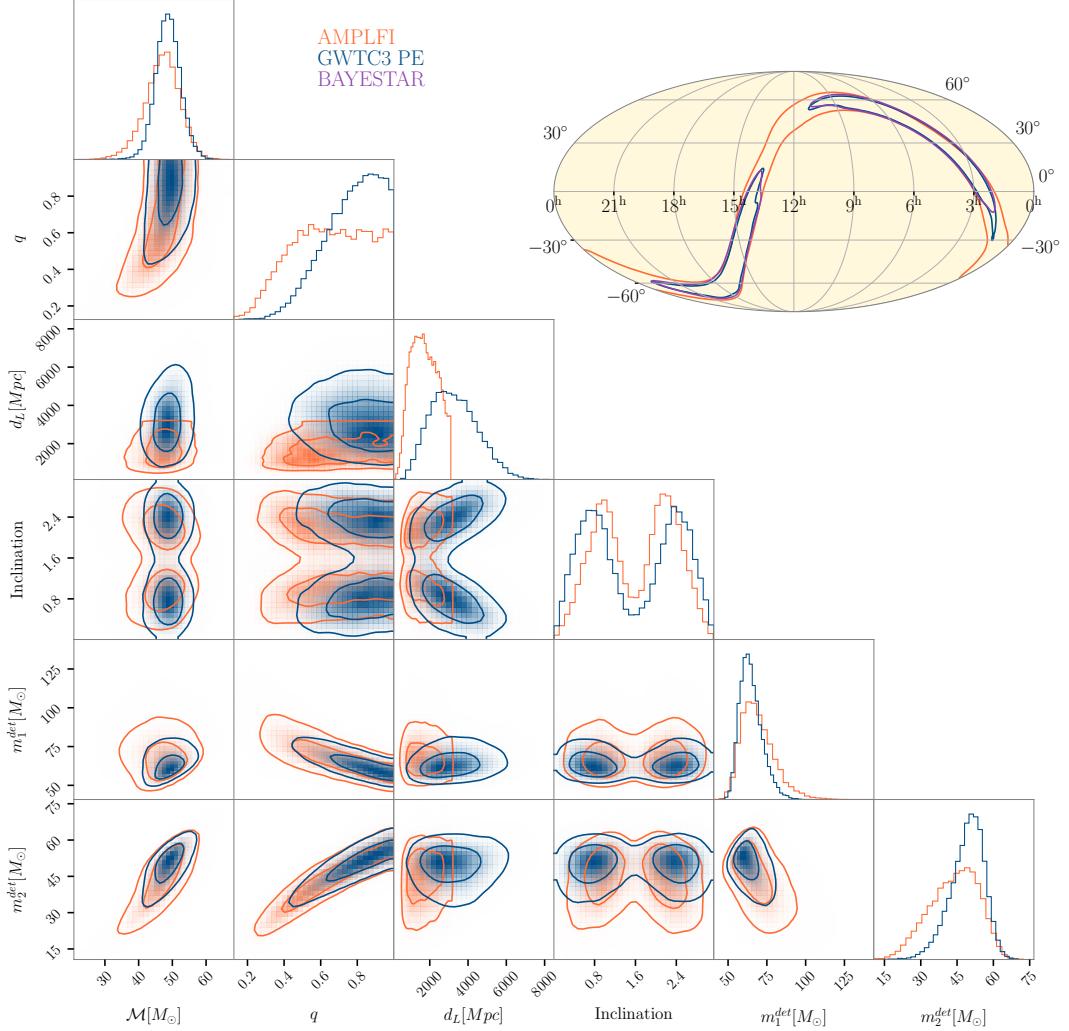
FIG. 18: Posterior comparison between AMPLFI and GWTC-3 result for GW200128_022011. Hanford and Livingston data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. This illustrates the scenario where the distance posterior has support well beyond AMPLFI's training prior. Still, AMPLFI is able to recover source parameters and localizations consistent with the GWTC-3 result.
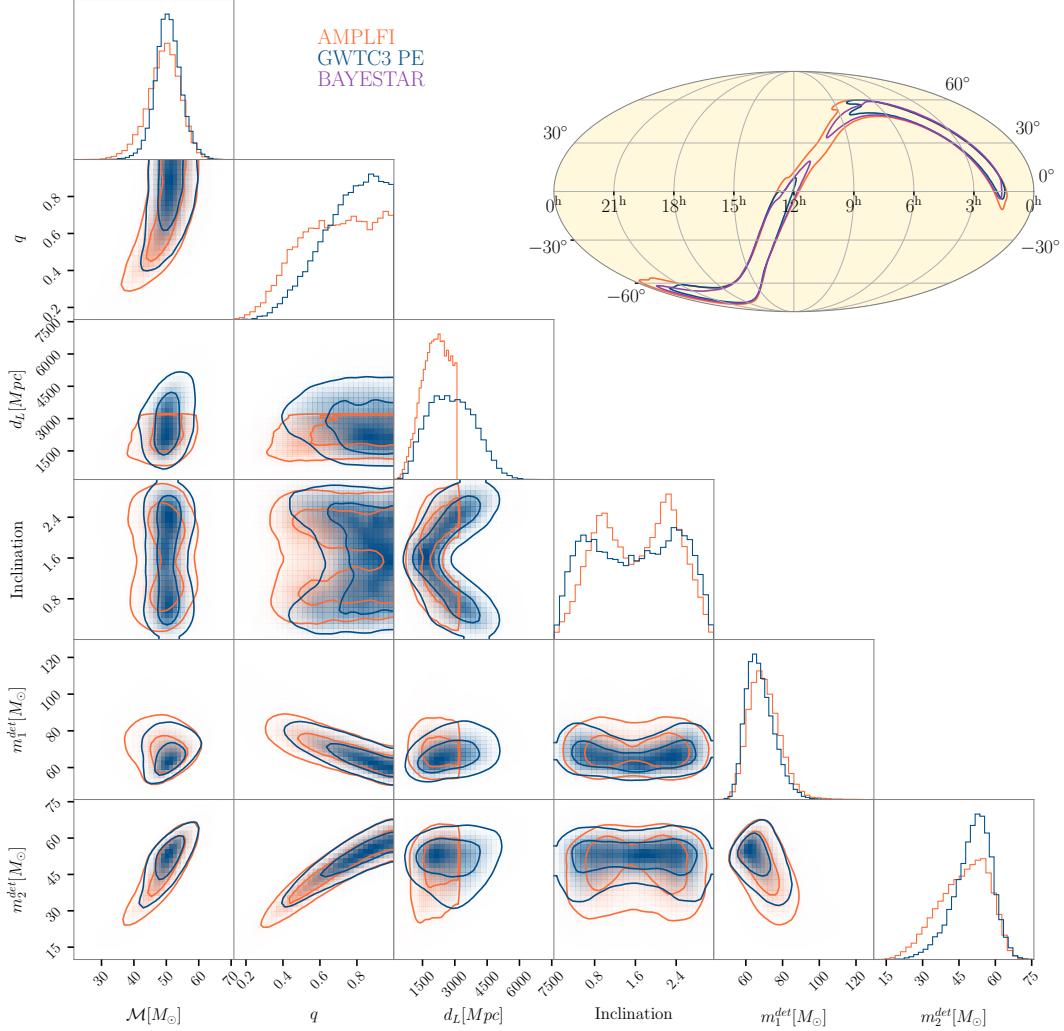
FIG. 19: Posterior comparison between AMPLFI and GWTC-3 result for GW191222_033537. Hanford and Livingston data is analyzed. The BAYESTAR sky map produced in low-latency is plotted in purple. This illustrates the scenario where the distance posterior has support well beyond AMPLFI's training prior. Still, AMPLFI is able to recover source parameters and localizations consistent with the GWTC-3 result.