Conditional noise deep learning for parameter estimation of gravitational wave events

Han-Shiang Kuo[1,†] and Feng-Li Lin[1,2,*]

[1]*Department of Physics, National Taiwan Normal University, Taipei 11677, Taiwan*
[2]*Center of Astronomy and Gravitation, National Taiwan Normal University, Taipei 11677, Taiwan*

(Received 4 August 2021; accepted 24 January 2022; published 9 February 2022)

We construct a Bayesian inference deep learning machine for parameter estimation of gravitational wave events of binaries of black hole coalescence. The structure of our deep Bayesian machine adopts the conditional variational autoencoder scheme by conditioning on both the gravitational wave strains and the variations of the amplitude spectral density (ASD) of the detector noise. We show that our deep Bayesian machine is capable of yielding posteriors compatible with the ones from the nested sampling method and better than the one without conditioning on the ASD. Our result implies that the process of parameter estimation can be accelerated significantly by deep learning even with large ASD drifting/variation. We also apply our deep Bayesian machine to the LIGO/Virgo O3 events, the result is compatible with the one by the traditional Bayesian inference method for the gravitational wave events with signal-to-noise ratios higher than typical threshold value. We use one GPU device, NVIDIA RTX3090, to train the deep learning machines, which takes about 12 hours. After training, it takes less than one second to generate the posterior of a gravitational wave event and is far faster than the conventional nested sampling method.

DOI: 10.1103/PhysRevD.105.044016

I. INTRODUCTION

Detection of gravitational waves (GW) from the distant compact binary coalescence has now become quite common since the first operation runs of LIGO started in 2015 [1], and up to now, about a hundred events have been found [2,3]. Due to the extreme weakness of the GW signal, the extraction of the source parameters from a given strain data requires heavy computational cost based on nested sampling [4–6] and Markov-Chain-Monte-Carlo algorithm [7,8], and such task of parameter estimation (PE) is very time consuming. This will then delay the announcement of the discoveries and the public sharing of the strain data for more general usages and PE results. Once the event rate of detection increases from few events per month to few events per day, this time delay issue of parameter inference will be more severe. Therefore, the acceleration of the PE for GW events is an urgent task in the vision of the improvement of the sensitivity for the new generation of gravitational wave detectors [9]. The main obstacle for accelerating the PE is the time-consuming scan of the likelihood function for obtaining the posteriors in Bayesian inference scheme [10–14]. One way to bypass this issue is to find a way of performing likelihood-free inference. This is indeed what deep learning can do by training the Bayesian inference machine with lots of mock data so that it can mimic the likelihood without event-by-event scanning. This deep-learning-based machine (or deep machine, for short) can then be implemented to extract the parameters of the GW events in a very efficient way. Some pioneer works in this direction have been done in [15,16] by adopting the variational autoencoder (VAE) [17,18] or the normalizing flow [19], and see [20,21] for the more recent progress. However, in these works, all training data share the same power spectral density (PSD) [or its square root, the amplitude spectral density (ASD)] of the detector's noise, which may not be realistic since the detector noise will drift in general. This means that the deep machine should be retrained for the events with different ASDs.

In this work, we extend the conditional VAE (CVAE) scheme developed in [15] to also conditioning on the ASD of the detector noise so that the resultant deep machine can deal with the GW events measured at different time intervals, for which the ASD will drift accordingly.[1] When finishing this note, we find that a similar consideration is also adopted in recent work [21] in the scheme of normalizing flow.

*Corresponding author.
fengli.lin@gmail.com
†hance30258@gmail.com

[1]Our machine-learning codes used for this work are implemented based on TensorFlow [22] and can be found in the open Gitlab forum: https://gitlab.com/hance30258/gwcvae. Besides, we adopt BILBY [14] to perform conventional Nested Sampling PE dynesty [6] to get the results for comparison.

The remainder of this paper is organized as follows. In the next section, we will sketch the scheme of CVAE for the inference of the source parameters of the GW events without or with the conditional ASD. In Sec. III we describe how we prepare the training data, especially on how to prepare the variations of ASD and the mock strain data. In Sec. IV we describe the detailed structure of our CVAE model, such as the layer structures and the hyperparameters. In Sec. V we first discuss the training procedure, including the way of avoiding KL collapse and the learning rate decay, and carry out the self-check of our Bayesian inference machine. We then show the performance of our machine when applying to the mock data by comparing it to the traditional PE method by their posteriors. We also consider the endurance of our machine to the drift of the ASD when compared to the CVAE model but without conditional ASD. In Sec. VI, we apply our Bayesian inference machine to LIGO/Virgo O3 events [23,24], and show their performance. Finally, we conclude our paper in Sec. VII.

II. CVAE FOR BAYESIAN INFERENCE OF GW EVENTS

The variational autoencoder (VAE) is a unsupervised machine learning scheme, which can be used to reveal the distribution functions of the input data. It first compresses the input data into the hidden layer by its encoder part, and then decompresses the hidden layer into the output by its decoder part. For example, if we prepare many mock strains as the training data, then the resultant well-trained machine can learn the distribution of the strains, and the hidden layers will encode the information about the distribution of the source parameters. However, to make the VAE be useful for the inference of the source parameters, we need to train the machine by simultaneously providing the strains $\{y\}$ and the associated source parameters $\{x\}$ as the input data but feeding to different encoders. The schematic structure of CVAE is similar to what is shown in Fig. 1. The loss function of this machine can be thought of as the upper bound on the negative of the posterior distribution $p(x|y)$, i.e., the so-called evidence lower bound (ELBO) and denoted by $\mathcal{L}_{\rm ELBO}$,

$$-\log p(x|y) \le \mathbf{E}_{z\sim E_{w_1}(z|x,y)}[-\log D_{w_3}(x|y,z)] + \mathbf{D}_{KL}[E_{w_1}(z|x,y)\|E_{w_2}(z|y)] \quad (1)$$

where E_{w_i} for $i=1,\ 2$ denote the distributions of the encoders with the associated weights and biases denoted by w_i, and D_{w_3} the one of the decoder with w_3 the associated weights and biases. Moreover, the arguments and the conditional arguments of the encoders and decoder denote their outputs and inputs, respectively. The right-handed side of the first line of (1) is the so-called reconstruction loss measuring the difference between input and output, and the second line is the KullbackLeibler (KL) loss measuring the difference between the hidden layer distributions of the two encoders.

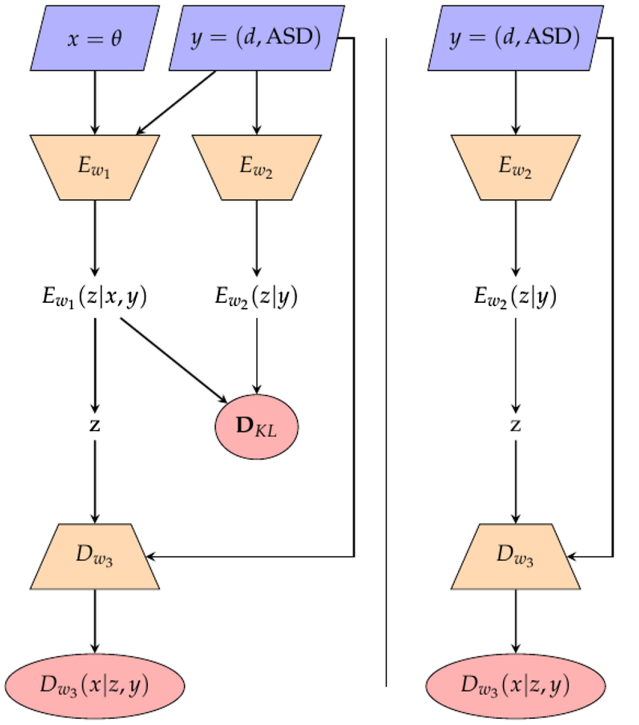

FIG. 1. The schematic structure of CVAE for the inference of source parameters of GW events. The goal is to generate the posteriors $p(x|y)$ of source parameters efficiently for a given strain data without knowing the likelihood $p(y|x)$. Left: the CVAE machine with two encoders E_{w_1}, E_{w_2} and one decoder D_{w_3}. Right: the Bayesian inference machine, which is obtained by removing the E_{w_1} part of CVAE after the CVAE is well trained, so that we expect $p(x|y) \approx \mathbf{E}_{z\sim E_{w_2}(z|y)}[D_{w_3}(x|z,y)]$. Therefore, its outputs are the posteriors of the source parameters. Our scheme shown here is a generalization of [15] by adding the ASD of the detector noise as the conditional inputs besides the associated strain data.

After the training, we can remove the part associated with the source parameters but keep only the one associated with the strains, so that the remaining part (as shown on the right part of Fig. 1) can be treated as the Bayesian inference machine to output the posteriors of the source parameters for a given input strain. Namely, we expect

$$p(x|y) \approx \mathbf{E}_{z\sim E_{w_2}(z|y)}[D_{w_3}(x|z,y)]. \quad (2)$$

Even though $D_{w_3}(x|z,y)$ is a Gaussian distribution, the average over $z\sim E_{w_2}$ will lead to non-Gaussian posterior approximation, as generally expected.

The above scheme was first proposed and implemented in [15], and can be shown to produce compatible posteriors in comparison to the conventional PE. However, in reality, the ASD of the detector's noise can drift so that ASD varies

TABLE I. Ranges of the priors for the BBH GW events adopted for the training data of the CVAE models used in this paper.

Parameters	Symbol	Prior	Range[a]	Units
Mass 1	m_1	Uniform	[20, 65]	Solar masses
Mass 2	m_2	Uniform	[20, 65]	Solar masses
Luminosity distance	d_L	Uniform Volume	[1200, 2200]	Mpc
Time of coalescence	t_c	Uniform	[0.65, 0.85]	Seconds
Phase at coalescence	ϕ_0	Uniform	$[0, 2\pi]$	Radians
Right ascension	α	·	1.84	Radians
Declination	δ	·	−0.62	Radians
Inclination	η	·	0	Radians
Polarization	ϕ	·	0	Radians
Epoch	·	·	1242459857	GPS time
Detector	·	·	Livingston	·

[a]The prior ranges chosen here are aiming at performing the PE of the real LIGO/Virgo's O3a events with some adjustments due to the limitation of our computing resources.

event by event. This drifting effect has not been taken into account in [15]. In this work, we extend the CVAE scheme of [15] to also include the variations of ASD as the conditional input data. The new scheme is shown in Fig. 1. This is the same as the one implemented in [15] except that an ensemble of ASD is also conditioned when training, and an ASD should be provided as the input along with the corresponding strain data when generating the posteriors of a GW event by the resultant Bayesian inference machine, i.e., the right part of the Fig. 1.

III. PREPARATION OF TRAINING DATA

As discussed, the training data include both the strain data and the ASD of the detector noise. As a proof-of-concept study, we only consider the binaries of black holes (BBH) without spin, which are labeled by two intrinsic parameters, i.e., the component masses m_1, m_2. Besides, we also have the extrinsic parameters describing the locations of the binaries. For simplicity, we fix all the extrinsic parameters except the luminosity distance d_L, which dictates the signal-to-noise ratio (SNR). Moreover, in the usual conventional PE, we need to optimize the matched filtering overlap by adjusting the time of coalescence t_c and phase at coalescence ϕ_0. Thus, we also include t_c and ϕ_0 as the parameters for inference. In total, we have five parameters for inference, and their ranges for flat priors and the fixed values of other parameters are given in Table I.

Unlike in [15], we adopt the frequency-domain templates to generate the strain data of one-second duration, instead of the time-domain ones as in [15]. We also change the sampling rate from 256 Hz to 1024 Hz to yield waveforms of 512 Hz bandwidth after fast Fourier transform and cover the high-frequency part of typical real-data waveforms. With the setup of priors given in Table I, we sample 2×10^6 sets of parameters to produce the theoretical waveforms by the IMRPhnomPv2 waveform model [25], which later will be used to superpose with the sample detector noise to produce the mock strain data.

Now, we turn to the preparation of the set of ASD templates used for training our CVAE. To take care of the ASD variations in LIGO/Virgo detectors, we would like to simulate an ASD model by fitting to a backbone set of 2000 sample ASDs, each of which is obtained from a 4096-second segment of the O3a strain data. These 2000 segments released by LIGO Scientific Collaboration cover the whole strain data of LIGO/Virgo's O3a run-time. Each 4096-second segment is divided into 4096 one-second pieces. Each piece is sampled with 1024 Hz to get a component ASD. The mean of the 4096 component ASDs yields a sample ASD in the backbone set. A 4096-second segment seems too long for a typical one-second BBH event due to significant noise-drifting. However, by taking the mean to yield an ASD, the glitches could be averaged out. Since our CVAE model is trained to combat ASD variations, thus we choose such a long segment to eliminate the glitches in advance.

To construct a stochastic ASD model to simulate ASD variations from the above backbone set of ASDs, we first construct a minimal ASD profile $A_m[f]$ obtained by collecting the minimum of each frequency bin from the above backbone set of ASDs. Based on $A_m[f]$ as shown in Fig. 4, we assume that the variations of the logarithm of the ASD $A[f]$ mainly obey the chi-square distribution $\alpha \equiv Z_1^2 + Z_2^2$ of two degrees of freedom with $Z_{1,2} = \mathcal{N}(0,1)$, but with a small but uniform residual variation $\beta \equiv \mathcal{N}(0,1/8)$ for all frequency bins. Noting that $\mathcal{N}(\mu,\sigma)$ denotes a Gaussian distribution with mean and variance (μ,σ^2). Thus, our ansatz stochastic ASD model is given by

$$\delta \log A[f] \equiv \log A[f] - \log A_m[f] = \alpha \log r[f] + \beta. \quad (3)$$

We then use the aforementioned backbone set of ASDs to fit the overall variation factor $r[f]$ of 513 frequency bins,

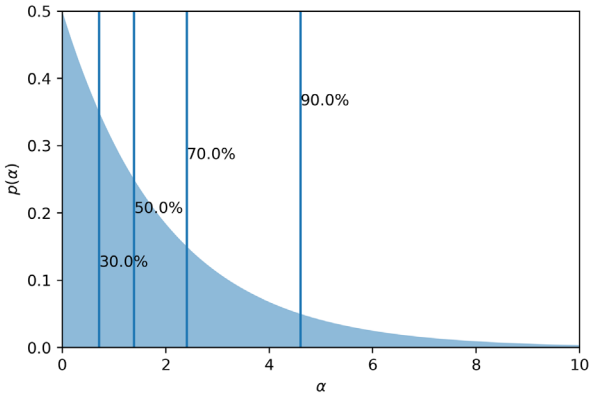

FIG. 2. Chi-square distribution for α used in (3), and the vertical lines indicate the corresponding confidence levels.

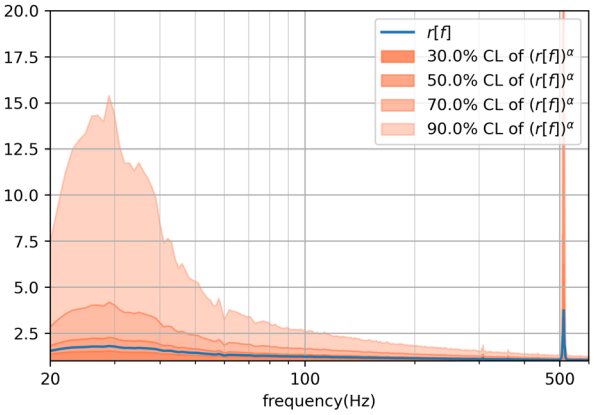

FIG. 3. Profiles of ASD variations from the stochastic ASD model (3). The blue curve is the fitted $r[f]$. The shaded pink regions indicate the various profiles of the overall variation factor $(r[f])^{\alpha}$ at 30%, 50%, 70% and 90% confidence levels, respectively. We see that the overall ASD variation factor can be larger than 15 for some frequency range.

see more discussion of the fitting procedure in Appendix. In Fig. 2, we show the chi-square distribution α and the corresponding confidence level, and in Fig. 3 we show the overall variation factor $r[f]$ which turns out to be almost a continuous function, and the histograms of the overall drifting factor $(r[f])^{\alpha}$. We see that the overall ASD variation factor can be larger than 15 for some frequency ranges at the 90% confidence level of the ASD variation.

Since our ASD model (3) is obtained by fitting the sample ASDs of the whole LIGO/Virgo's O3a run-time, thus the ASDs generated from it can catch the generic features of ASD variation/drifting during the O3a. In Fig. 4, we show the range of the ASD variations generated by the stochastic ASD model (3). However, we should also caution the readers that the task of constructing a perfect ASD model is almost impossible since not all origins of the

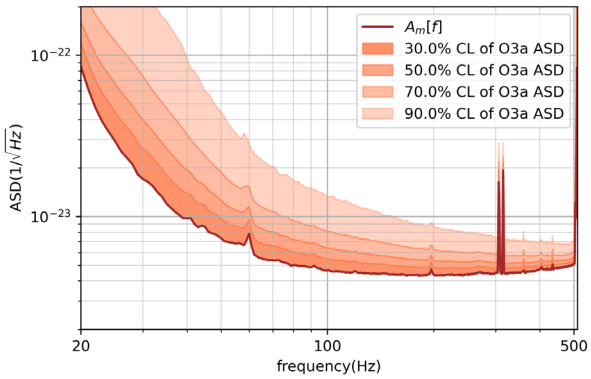

FIG. 4. Ranges of ASDs generated from our stochastic ASD model (3) with the pink shaded regions indicating the profile ranges at 30%, 50%, 70% and 90% confidence levels, respectively. The brown curve is the minimal ASD profile $A_m[f]$.

variations are well understood, e.g., the unexpected frequent glitches. Our ASD model (3) is simple and may not catch the full features of the ASD variations.

We will use the ASD model (3) to generate about two million ASDs as the training set for our CVAE model. Therefore, the resultant PE model will learn the generic feature of ASDs and can endure the drifting of ASD. This contrasts with the PE model considered in [15,16], in which the ASD used to whiten the strain data is fixed so that the model should be retrained if the drifting of the ASD is significant.

Based on the above discussions, we can generate a mock strain as follows. We randomly pick up a theoretical waveform $h[f]$ and ASD $A[f]$ from the above prepared sets, then we can form a noise $n[f]$ and a strain $d[f]$ in the frequency domain as follows

$$d[f] = h[f] + n[f], \tag{4}$$

$$n[f] = \frac{1}{\Delta f} W[f] \odot A[f] \tag{5}$$

where Δf is the frequency bin size which we set to 1 Hz in this work, and $W[f]$ is the white noise in the frequency domain, which is responsible for the unit Gaussian noise. In Fig. 5, we show a typical example.

With the above procedure, we generate about 2×10^6 mock strains, of which 80% will be used as the training data set for CVAE, and 20% as validation data set for the resultant Bayesian inference machine. This amount of the training data set is huge enough to exhaust almost all possible strain data realizations.

Moreover, to quantify how the ASD variations affect the quality of the strain data, we compare the histograms of the SNR obtained from the 2×10^6 mock strain data with and without ASD variation. The result is shown in Fig. 6, from

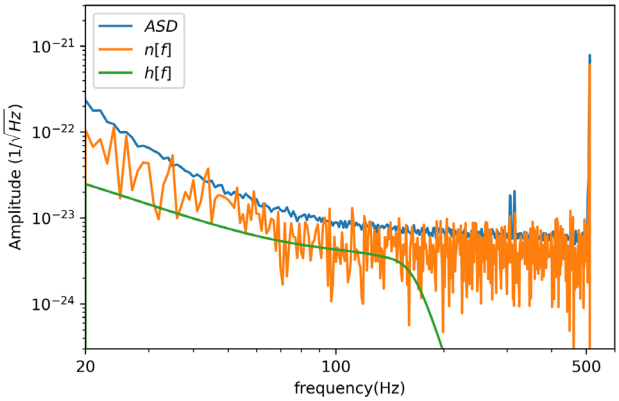

FIG. 5. A typical mock waveform $h[f]$ (green) and noise $n[f]$ (orange). The latter is generated from a given ASD (blue) obtained by (3). Here we only plot the amplitude part.

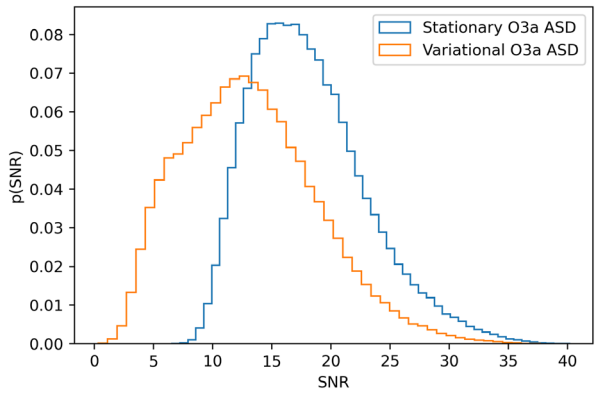

FIG. 6. Histograms of SNRs for all the training strain data generated by using the priors in Table I and the ASDs generated by $A_m[f]$ (blue) and $A[f]$ (orange) of (3). It shows a significant down-shift of the SNR distribution due to the noise-drifting.

which we can see that the SNR distribution is downshifted by quite an amount. This is expected, as the drifting factor at some particular frequency range can be as large as 15. It implies that a well-trained CVAE model augmented by our ASD model (3) should be able to combat the noise drifting and yield reliable PE results.

IV. THE DETAILED STRUCTURE OF CVAE MODEL

The schematic structure of our CVAE model and the resultant Bayesian inference machine has been shown in Fig. 1. Now we would like to expose its detailed structure. For simplicity, our CVAE model is composed of only dense layers but not other types of layers. However, it works. We simply stack the dense layers to construct three neural networks (NNs), i.e., two encoders and a decoder. Moreover, we adopt almost the same layer structure for all three NNs, see Fig. 7 for the details. The only

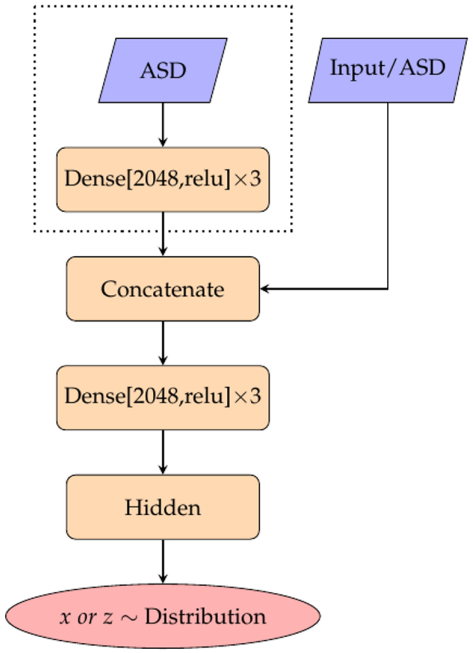

FIG. 7. Structure of neural network used in CVAE model of Fig. 1. Note that we adopt this same NN for all three NNs, i.e., E_{w_1}, E_{w_2}, and D_{w_3} of Fig. 1. The ASD is the common input, and there is an additional input denoted by Input/ASD. The Input/ASD and the dimensions of the hidden layer vary for different NNs, which we summarize in Table II. The output is a random latent vector, z or $x \sim$ distribution which is also specified in Table II. The dash-lined box contains the part associated with the conditional ASD, which is absent in the CVAE model of [15].

differences among them are the input data and the dimensions and the realizations of the hidden layers. Specifically, we use 8- and 5-dimensional multivariate Gaussian distributions for the hidden layers of E_{w_1} and D_{w_3}, and adopt a more powerful mixture Gaussian distribution layer for E_{w_2}, which has eight dimensions and each dimension is composed of eight components of Gaussian normal distributions. From our experience, adopting the mixture Gaussian for the hidden layer enhances a lot the performance of the model. All hidden layers with Gaussian distributions are realized by the standard reparameterization trick used for variational autoencoder [17]. In Table II, we summarize these differences.

Note that the latent vectors for the encoders E_{w_1} and E_{w_2} are denoted by z, which will then be input to the decoder. However, the output of the decoder is again a random vector, whose components are identified as the source parameters, i.e., $x = \theta$. The distribution of x gives the approximate posterior of the source parameter θ through the averaging procedure given in (2).

The hyperparameters specified in Fig. 7 and Table II achieve well training of our CVAE model, despite that they can be varied. However, we find that it is sufficient for well

TABLE II. Input and hidden layers of the CVAE model.

	$E_{w_1}(z\|x,y)$[a]	$E_{w_2}(z\|y)$	$D_{w_3}(x\|z,y)$
Input/ASD[b]	$[\theta, d]$	d	$[z, d]$
Hidden[c]	Dense$[16, \mathsf{linear}]$	Dense$[24, \mathsf{linear}]$	Dense$[10, \mathsf{linear}]$
Distribution[d]	$Gaussian(8)$	$MixtureNormal(8,8)$[e]	$Gaussian(5)$

[a]Here $x = \theta$ denoting the source parameters, $y = (\mathrm{ASD}, d)$ with d the strain, and z the random latent vector.
[b]This means the additional input other than ASD.
[c]This is the hidden layer whose outputs are means and variances.
[d]This is the distribution used to generate the random latent vector z, whose means and variances are given by the outputs of the hidden layers.
[e]This is the linear combination of 8 Gaussian distributions.

training if the dimension of the hidden layer is greater than the dimension of the target variable. Using more dimensions may need a longer time to train but improve the performance just slightly.

V. TRAINING THE CVAE MODEL AND THE PERFORMANCE

With the above structure of the CVAE model, we train the model by the aforementioned training data set of batch size 2048. We then calculate the loss function, i.e., $\mathcal{L}_{\mathrm{ELBO}}$ and update the model by Adam optimizer [26] with learning rate 10^{-4}. The reconstruction loss is evaluated by replacing x in $\mathbf{E}_{z\sim E_{w_1}(z|x,y)}[-\log D_{w_3}(x|y,z)]$ by the input source parameter θ. The averaging procedure over z in the above and in evaluating the KL loss is done by the Monte-Carlo method. There are two effective ways to achieve well training. The first way is addressed to the so-called KL collapse [27], which states that KL loss may happen to be extremely small so that the variational nature of CVAE is lost. To avoid the KL collapse, we can adopt the annealing procedure by introducing an annealing factor $b \in [0,1]$ so that the ELBO is changed to

$$\mathcal{L}^{(b)}_{\mathrm{ELBO}} = \mathbf{E}_{z\sim E_{w_1}}[-\log D_{w_3}] + b\mathbf{D}_{\mathrm{KL}}[E_{w_1}\|E_{w_2}]. \quad (6)$$

In the early training phase, we slowly tune up the annealing factor to avoid the KL collapse. When b is far smaller than one, we are mainly training the VAE, i.e., ignoring the E_{w_2} which will be optimized again when b is close to one. Specifically, we proceed the KL annealing for the first 5 epochs with the following annealing behavior

$$b(t) = b_0 \sin\left(\frac{\pi}{2} t/c\right), \quad (7)$$

where t denotes the number of generations (each generation means finishing a batch training) and $c \approx 10^3$ is the number of generations within an epoch, and the values of b_0 for these 5 epochs are set to $[10^{-2}, \frac{1}{4}, \frac{1}{2}, 1, 1]$. Note that the annealing rate gradually approaches zero at the end of each epoch. After these 5 epochs, b will be set to one for the remaining training period, which is about 10^3 epochs.

The second effective way to achieve well training more efficiently is to reduce the learning rate gradually. We reduce the learning rate lr at every generation at such a rate $\mathrm{lr}(t) = 2^{-\frac{t}{2\times 10^5}}\mathrm{lr}_0$ in the total training period of 10^6 generation. With the implementation of the above two effective ways, we can achieve well training of our CVAE model. A typical example for the evolution of the reconstruction and KL losses at the training and validation periods is shown in Fig. 8, which indicates the KL annealing at the early training phase. Moreover, the perfect overlap between training and validation losses indicates there is no overfitting.

In the following, we will compare our conditional-ASD CVAE model, which we denote as $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$, and the one used in [15] but with KL annealing and learning rate decay incorporated, which we denote as $\overline{\mathrm{CVAE}}_{\text{nc-ASD}}$ with "no-conditioning" short-handed by nc. That is, $\overline{\mathrm{CVAE}}_{\text{nc-ASD}}$ has the same layer structure as shown in Fig. 7 except for the part inside the dash-lined box, which is used for conditioning on ASD.[2] Note that we implement $\overline{\mathrm{CVAE}}_{\text{nc-ASD}}$ by our own code and then train it with strain data whitened by stationary ASD, i.e., the $A_m[f]$ given in (3). Also, the mixture Gaussian distribution is used for E_{w_2} in $\overline{\mathrm{CVAE}}_{\text{nc-ASD}}$ rather than the simple diagonal Gaussian distribution used in [15]. Here, the over-line is to remind that the KL annealing and learning rate decay are implemented in the training procedure. This is in contrast to the CVAE model used in [15], which we denote as $\mathrm{CVAE}_{\text{nc-ASD}}$. It turns out that the implementation of KL annealing and learning rate decay in the training procedure is important in achieving better accuracy of final posteriors, as shown below in comparing the P-P plots and histograms of KL divergences.

[2]This is the layer structure used in the version 1 and 2 of [15]. In the latest version (version 3) of [15], a more complicated structure with convolutional neural networks is adopted. However, the performance of P-P plot [29] and KL divergence [30] of $\overline{\mathrm{CVAE}}_{\text{nc-ASD}}$ shown below is still better than the latest ones in [15].

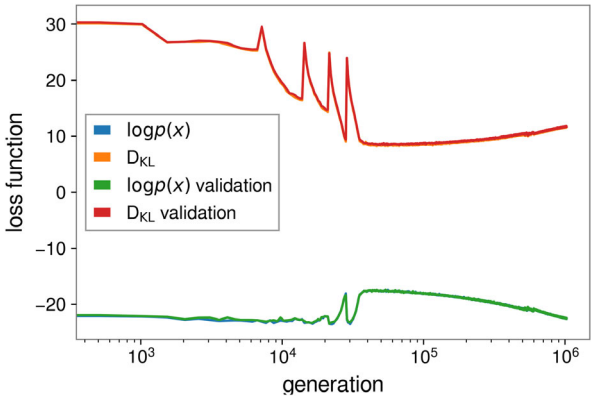

FIG. 8. Training and validation loss for each generation. The variation at early stage are caused by cyclic KL annealing [28]. The perfect overlap between training and validation loss indicates there is no overfitting.

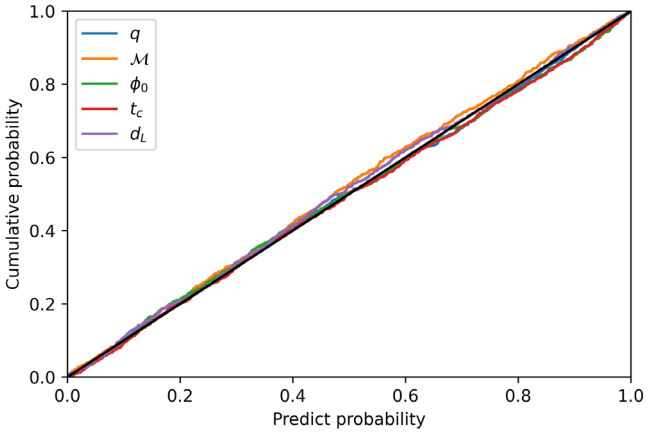

FIG. 9. P-P plot for our CVAE model. The CDF is calculated by 10^3 mock data. For each mock data, we use 2×10^4 samples to estimate the p-value of each parameter.

Our models are all trained by one GPU device, NVIDIA RTX3090, where the training time is 12 hours for $\overline{\text{CVAE}}_{\text{ASD}}$ and 6 hours for $\overline{\text{CVAE}}_{\text{nc-ASD}}$. However, the computational time for generating the posterior of a GW event by evaluating 10^5 distribution samples is less than 1 second for both models. This is far faster than the time required by dynesty running on a single CPU core, which is about 12 hours.

After training the CVAE models, the first thing is to check the self-consistency of the resultant Bayesian inference machine, i.e., calculating the P-P plot, which is the cumulative distribution function of the p-value of the posteriors, i.e., p-value $= p[p(x|y) > x|\text{null hypothesis}]$. By construction, the distributions of the input parameters should equal the posteriors, so that p-value should be the uniform of unity. Thus, the P-P plot should be diagonal for self-consistency. The result is shown in Fig. 9 and indicates that our Bayesian inference machine $\overline{\text{CVAE}}_{\text{ASD}}$ is self-consistent. Compared to the P-P plot shown in Fig. 4 of [15] obtained for $\text{CVAE}_{\text{nc-ASD}}$, the one shown here is far more convergent. This is due to the implementation of KL annealing and learning rate decay, and also to the conditional ASDs.

Next, we compute the posterior of a typical mock GW event by $\overline{\text{CVAE}}_{\text{ASD}}$. To produce this posterior, we need to sample about 8×10^4 latent vectors from $z \sim E_{w_2}(z|y)$, and then use (2) to average over z by Monte-Carlo method to obtain the posterior $p(\theta|d, \text{ASD})$ for the source parameters θ.[3] The results are shown in Fig. 10, in which we also compare with the results obtained from the traditional PE algorithm dynesty. We can see that the marginal posteriors from both methods are compatible.

One essential question about the performance of our CVAE Bayesian machine $\overline{\text{CVAE}}_{\text{ASD}}$ is how good it is when compared to $\overline{\text{CVAE}}_{\text{nc-ASD}}$. One way to characterize such a performance is to compare their KL divergences with the posterior obtained from dynesty, i.e., to compare $\mathbf{D}_{\text{KL}}(p_{\text{dynesty}}\|p_{\text{ASD}})$, and $\mathbf{D}_{\text{KL}}(p_{\text{dynesty}}\|p_{\text{nc-ASD}})$, where

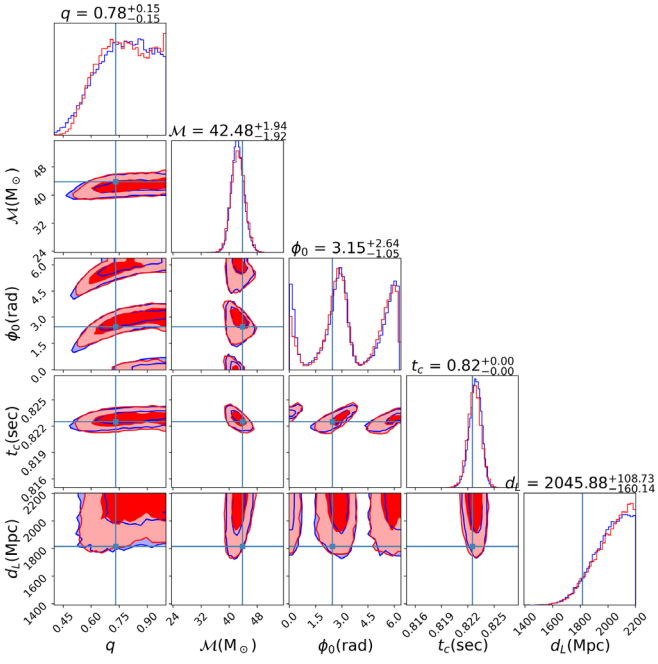

FIG. 10. The marginal posteriors of a typical mock GW event evaluated from $\overline{\text{CVAE}}_{\text{ASD}}$ (red) and the traditional PE method, i.e., dynesty (blue). The contour represents 50% and 90% credible level and the true parameter are shown by the blue lines. The KL divergences between posteriors of these two method are (0.0005, 0.0061, 0.0184, 0.0262, 0.0010) in the following order of the parameters: $(q, \mathcal{M}, \phi_0, t_c, d_L)$.

[3]The amount of the samples, i.e., 8×10^4, used here to obtain the posterior is chosen to be the same as in [15] so that we can compare the performance of our CVAE model with theirs on an equal footing. Later on, we will also use the same amount of samples to obtain the posterior by the nested sampling method dynesty for comparisons.

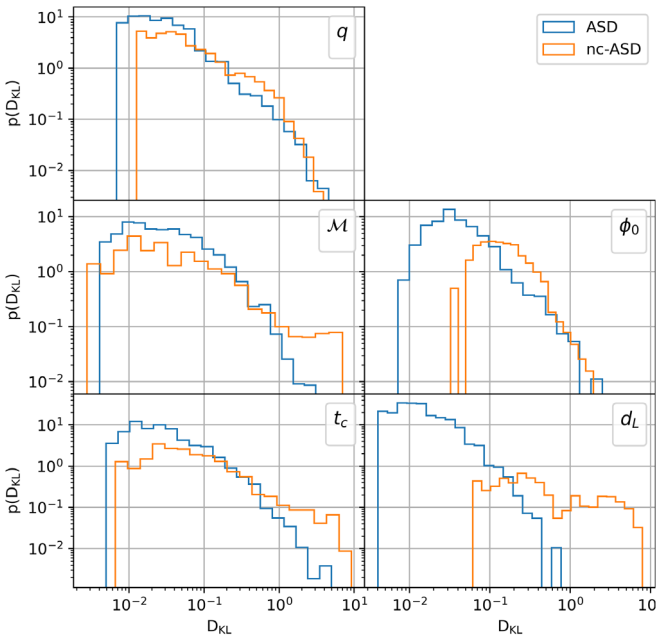

FIG. 11. Histograms of KL divergences, i.e., $\mathbf{D}_{\mathrm{KL}}(p_{\mathrm{dynesty}}\|p_{\mathrm{ASD}})$ (blue) and $\mathbf{D}_{\mathrm{KL}}(p_{\mathrm{dynesty}}\|p_{\mathrm{nc\text{-}ASD}})$ (orange) for all five parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$ over 512 mock GW strains of BBH with ASD variations similar to the one in Fig. 4. The preparation of these mock strains is described in the main text. Note that p_{dynesty}, p_{ASD} and $p_{\mathrm{nc\text{-}ASD}}$ are the posteriors obtained from dynesty, $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$, and $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$, respectively. We see that $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ performs better than $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$, especially for ϕ_0 and d_L at $\mathbf{D}_{\mathrm{KL}} \sim \mathcal{O}(1)$.

p_{dynesty}, p_{ASD} and $p_{\mathrm{nc\text{-}ASD}}$ denote the posteriors obtained from dynesty, $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ and $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$, respectively. Note that smaller KL divergence means the posteriors from both CVAE models are close to the one from dynesty. Usually, the threshold for an acceptable "nice" result is for the KL divergence to be smaller than 0.1. Moreover, compared to the KL divergences shown in Fig. 5 of [15] obtained for $\mathrm{CVAE}_{\mathrm{nc\text{-}ASD}}$, the results shown here are about one to two orders better. Again, this is due to the implementation of KL annealing and learning rate decay in the training procedure.

We prepare 512 mock GW strains as the inputs to the three Bayesian machines for comparison. These mock GW strains are generated according to the BBH priors in Table I and the ASDs obtained from (3). We then evaluate the distributions of $\mathbf{D}_{\mathrm{KL}}(p_{\mathrm{dynesty}}\|p_{\mathrm{ASD}})$, and $\mathbf{D}_{\mathrm{KL}}(p_{\mathrm{dynesty}}\|p_{\mathrm{nc\text{-}ASD}})$ for all five parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$ over the above mock strains. To obtain p_{dynesty} we use BILBY to perform the dynesty sampling [14] with 5000 live points and dlogz equal to 0.1. The results are shown in Fig. 11. We see that $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ performs better than $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$, especially for d_L at $\mathbf{D}_{\mathrm{KL}} \sim \mathcal{O}(1)$ by about one order of improvement.

Besides the histograms of KL divergences shown in Fig. 11, we also list these KL divergences according to the SNR of each mock event, and the result is shown in Fig. 12.

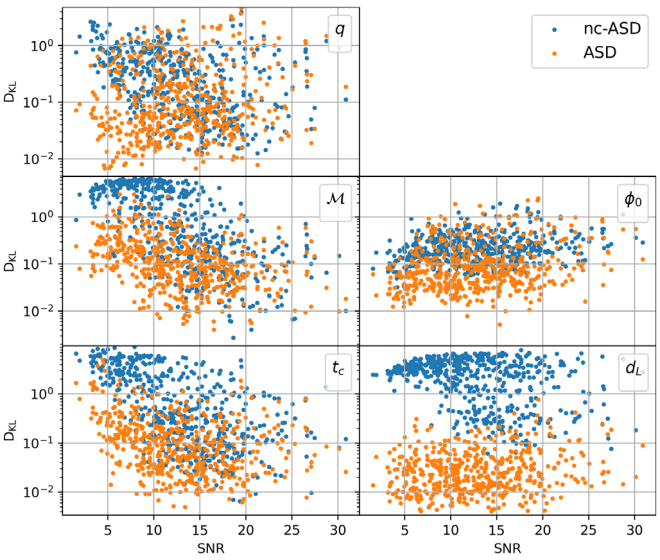

FIG. 12. Comparison of the dependence on SNR for the two KL divergences for all five parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$, i.e., $\mathbf{D}_{\mathrm{KL}}[p_{\mathrm{dynesty}}\|p_{\mathrm{ASD}}]$ (blue) and $\mathbf{D}_{\mathrm{KL}}[p_{\mathrm{dynesty}}\|p_{\mathrm{nc\text{-}ASD}}]$ (orange), which are already evaluated in Fig. 11. We see that our CVAE model has better performance, especially at low SNR.

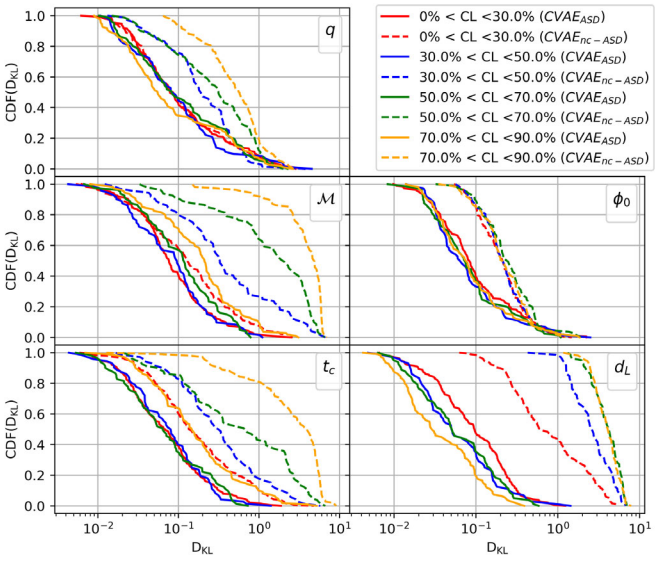

FIG. 13. Comparison of the capability of $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ and $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$ in fighting against the ASD variations at various confidence levels (CLs): $0\% < \mathrm{CL} < 30\%$ (red), $30\% < \mathrm{CL} < 50\%$ (blue), $50\% < \mathrm{CL} < 70\%$ (green) and $70\% < \mathrm{CL} < 90\%$ (orange). This is characterized by plotting the cumulative distribution functions (CDFs) of KL divergences $\mathbf{D}_{\mathrm{KL}}[p_{\mathrm{dynesty}}\|p_{\mathrm{ASD}}]$ (solid line) and $\mathbf{D}_{\mathrm{KL}}[p_{\mathrm{dynesty}}\|p_{\mathrm{nc\text{-}ASD}}]$ (dashed line) of parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$. The plots show that $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ is better than $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$ in fighting against ASD variations. Besides, $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ is also less sensitive to the ASD variations than $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$.

In general, the CVAE models perform better at higher SNR when benchmarking by the dynesty results. This could be expected for the CVAE models with primitive structures. We can also see that $\overline{\text{CVAE}}_{\text{ASD}}$ has better overall performance than $\overline{\text{CVAE}}_{\text{nc-ASD}}$, especially for $\mathcal{M}$, t_c and d_L, and at low SNR. The improvement of PE performance by conditioning on ASDs seems quite uniform for all SNR. This implies that our scheme can help to sort out more GW events of low SNR.

Finally, we would like to compare the capability of CVAE_{ASD} and $\text{CVAE}_{\text{nc-ASD}}$ in fighting against ASD variations by tuning the confidence level of the overall variation factor shown in Fig. 4. To proceed, we first classify the mock strains used in Figs. 11 and 12 by the confidence level of $A[f]/A_m[f]$, and then plot the KL divergences $\mathbf{D}_{\text{KL}}[p_{\text{dynesty}}\|p_{\text{ASD}}]$ (solid line) and $\mathbf{D}_{\text{KL}}[p_{\text{dynesty}}\|p_{\text{nc-ASD}}]$ (dashed line) for each class. We show the results in Fig. 13 for four different ranges of confidence level. We see that $\overline{\text{CVAE}}_{\text{ASD}}$ is always better than $\overline{\text{CVAE}}_{\text{nc-ASD}}$ in fighting against ASD variations, as expected. Especially, the PE results of $\overline{\text{CVAE}}_{\text{nc-ASD}}$ are quite sensitive to the confidence level of ASD variations, however, the ones of $\overline{\text{CVAE}}_{\text{nc-ASD}}$ are not. This indicates the superiority of our CVAE model due to conditioning on an extensive set of ASDs generated by our ASD model (3).

VI. APPLICATION TO LIGO/VIRGO'S O3A EVENTS

In the previous section, we have shown that our CVAE model can in general perform better in obtaining PE results for the mock events than the one without conditioning on ASD variations. We now would like to continue the similar comparison for the real LIGO/Virgo's 39 BBH O3a events, from each of which we take one-second event strain data, along with the 64-second surrounding data to construct the corresponding ASD for the standard PE practice. For simplicity, we only use the strain data from LIGO/Livingston. We choose the same priors as listed in I to perform PE of the parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$ for these 39 BBH events.[4] Of course, the result will depend on how well our ASD model (3) can catch the ASD variation of these 39 BBH events with respect to $A_m[f]$. We will first present the PE results and the comparison with ones obtained from $\overline{\text{CVAE}}_{\text{nc-ASD}}$, and then discuss the effect of the quality of our ASD model on the results.

In Fig. 14 we show the marginal posteriors of the O3a event GW190910_112807, of which the SNR is 12.6, one of few events with SNR larger than 10 among 39 BBH events. We see that the results obtained by $\overline{\text{CVAE}}_{\text{ASD}}$ are

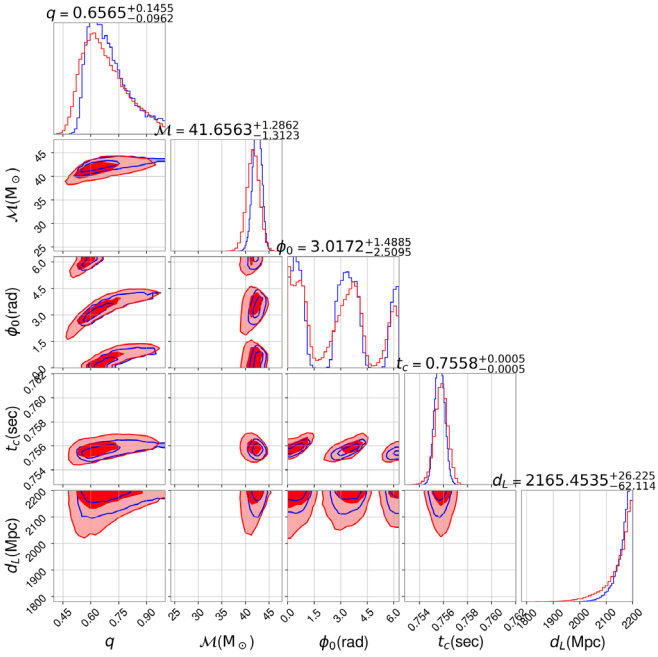

FIG. 14. Marginal posteriors of GW190910_112807 event obtained by $\overline{\text{CVAE}}_{\text{ASD}}$ (red) and the dynesty (blue). The SNR of this event is 12.6. The KL divergences, i.e., $\mathbf{D}_{\text{KL}}[p_{\text{dynesty}}\|p_{\text{ASD}}]$, of this event for the parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$ are (0.088, 0.29, 0.12, 0.12, 0.12). It shows that the PE results obtained by $\overline{\text{CVAE}}_{\text{ASD}}$ are compatible with the ones by dynesty. The values and the error margins of the parameters shown in this figure are the ones obtained by $\overline{\text{CVAE}}_{\text{ASD}}$, and the ones obtained by the dynesty are $q = 0.67^{+0.13}_{-0.08}$, $\mathcal{M} = 42.4^{+0.90}_{-0.92}$, $\phi_0 = 3.07^{+1.22}_{-2.62}$, $t_c = 0.75^{+4e-4}_{-4e-4}$ and $d_L = 2175.77^{+17.8}_{-36.88}$.

compatible with the ones obtained from the ones from the dynesty. This can be more precisely characterized by the values of the KL divergence for the parameters $(q, \mathcal{M}, \phi_0, t_c, d_L)$, which are (0.088, 0.29, 0.12, 0.12, 0.12). The PE performance of $\overline{\text{CVAE}}_{\text{ASD}}$ varies event by event, and the resultant KL divergences $\mathbf{D}_{\text{KL}}[p_{\text{dynesty}}\|p_{\text{ASD}}]$ and $\mathbf{D}_{\text{KL}}[p_{\text{dynesty}}\|p_{\text{nc-ASD}}]$ plotted according to the SNR for all 39 BBH O3a events are summarized in Fig. 15. The results show that overall $\overline{\text{CVAE}}_{\text{ASD}}$ performs better than $\overline{\text{CVAE}}_{\text{nc-ASD}}$. However, the superiority of $\overline{\text{CVAE}}_{\text{ASD}}$ is not as impressive as in the case of mock events, for which the ASDs are generated simply by our ASD model (3). On the other hand, the key feature of the ASDs for these 39 O3a events could deviate from the ones given by (3).

In Fig. 16 we compare the ASDs of the 39 O3a BBH events with the range of ASD variations from our ASD model (3), which is fitted by 2000 4096-second segments of O3a strain data. We can see that the ASDs of the GW events fall within the range of model ASD variations at the confidence level below 34%. This means that our ASD model indeed generates a far wider spectrum of ASD variations needed for the PE of these 39 GW events. It

[4]The prior ranges listed in Table I, especially the one for d_L, are narrower than the ones used by standard LIGO/Virgo's data analysis. However, as a proof-of-concept study, we just trade the prior range/accuracy for efficiency.

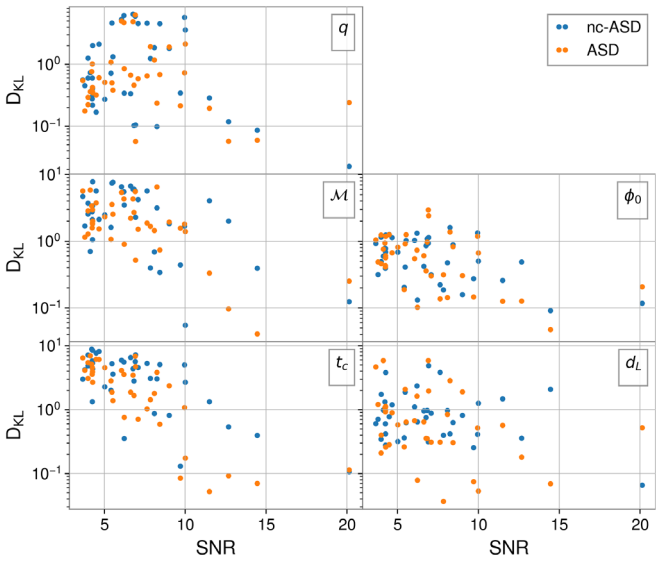

FIG. 15. KL divergences, i.e., $\mathbf{D}_{\mathrm{KL}}[p_{\mathrm{dynesty}}\|p_{\mathrm{ASD}}]$ (orange) and $\mathbf{D}_{\mathrm{KL}}[p_{\mathrm{dynesty}}\|p_{\mathrm{nc\text{-}ASD}}]$ (blue) listed according to the SNRs of the 39 BBH LIGO/Virgo O3 events. The result shows that $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ is better than $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$ in the overall PE performance. However, the PE superiority of $\overline{\mathrm{CVAE}}_{\mathrm{ASD}}$ is not as impressive as in the cases of mock events shown in Fig. 12.

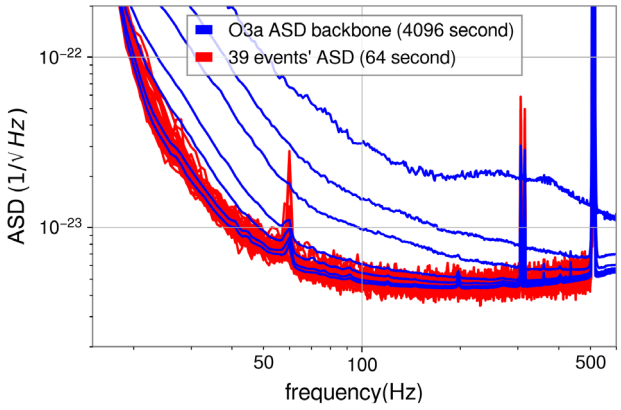

FIG. 16. Comparison of the ASDs surrounding the 39 O3a BBH events (red) and from the ASD model (3) simulated from 2000 4096-second segments (blue). The ASD for each GW event is constructed from a surrounding 64-second segment. From the bottom to the top, the blue curves show the range of ASD variations at the confidence levels of $[1\%, 10\%, 34\%, 50\%, 76\%, 90\%, 99\%]$.

implies that we may need to enlarge our machine structure for training to suppress the unnecessary ASD uncertainty for achieving higher PE accuracy. The wider range of ASD variation can be attributed to the long 4096-second duration of the segments, which are taken to average out the glitches but then introduce large variations away from the 64-second ASD segment of the GW events.

As curiosity about the effect of changing the duration of segments for the conditional ASD variations, in Fig. 17 we

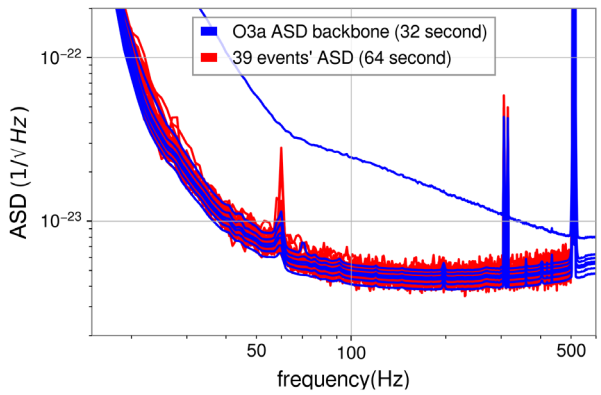

FIG. 17. Similar to Fig. 16 but now the ASD variations are given by 16000 32-second ASDs. The legends are the same as in Fig. 16.

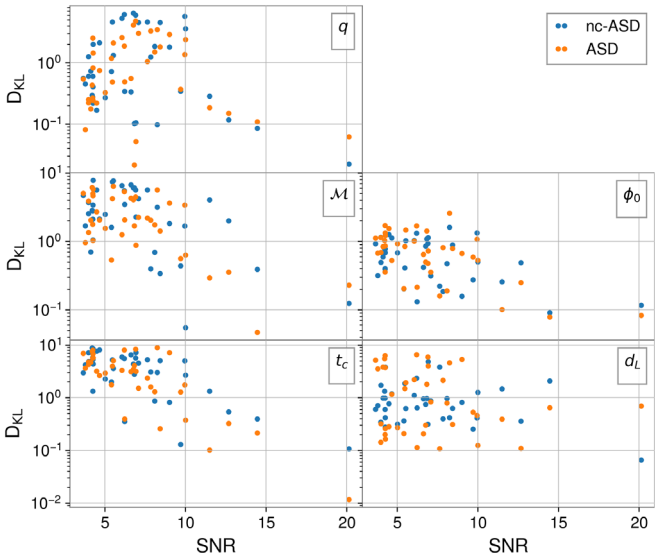

FIG. 18. The PE result of our CVAE model trained by 16000 32-second ASDs. The ASD for $\overline{\mathrm{CVAE}}_{\mathrm{nc\text{-}ASD}}$ is just $A_m[f]$ obtained from the minimum of the collection of 32-second ASDs. This is to compare with its counterpart trained by 4096-second ASDs as shown in Fig. 15.

show the ASD variations obtained from the collection of 16000 32-second ASDs of O3a data.[5] Compared to the results in Fig. 16, we can see that the new ASD distribution produces far tighter ASD variations to the 64-second ones surrounding the GW events. However, we may expect that the glitches in the ASDs may not be suppressed as much as for the 4096-second ones. Moreover, in order to compare with the 4096-second ones shown in Fig. 15, in Fig. 18 we

[5]In this case, we do not try to construct a stochastic ASD model like (3) as done for the 4096-second ones. Instead, we just treat these 16000 32-second ASDs as the training set for the CVAE model. A more sophisticated ASD model based on this collection is wanted to generate more amount of variational ASDs for future improvement.

show the PE results obtained from our CVAE models by conditioning on this collection of ASDs. On the other hand, the ASD for $\overline{\text{CVAE}}_{\text{nc-ASD}}$ is just $A_m[f]$ obtained from the minimum of this new collection of 32-second ASDs. We find that the overall PE performances of both models are not much different. This suggests that one will face the trade-off between the long-time variations and short-time glitches when simulating an ASD model to take care of its drifting. To further improve the PE performance of our CVAE model, we may either construct a more sophisticated ASD model than the simple (3) or enlarge the complexity of the neural network to accommodate more subtle ASD variations. Either way needs more trials and errors, such as the one done by [21].

VII. CONCLUSION

In this work, we construct a deep Bayesian machine conditioning on the variations of the detector's noise to perform the parameter estimation (PE) of binary black holes' gravitational wave (GW) events based on the deep learning scheme of conditional variational autoencoder (CVAE). This is a simple extension of the CVAE model proposed in [15] in which only strains but not the amplitude spectral density (ASD) of the detector noise are adopted as the conditional inputs to CVAE. Our motivation is to demonstrate the viability of a deep Bayesian machine that can adapt to the variations or drift of the detector's noise. This kind of machine can save time for retraining when performing PE for various GW events with slight variations of the detector noise.

As a proof of concept study, we choose a very simple layer structure, i.e., three dense layers, for two encoders and one decoder of CVAE. Despite such a humble deep machine, we show that the PE results for the mock strains with variations from a theoretical ASD are compatible with the ones obtained from the traditional PE method such as the dynesty once the tricks of KL annealing and learning-rate decay are implemented in the training procedure. Besides, we also show that our CVAE machine is better than the one of [15] in fighting against the ASD variations.

To test our CVAE model for real events and demonstrate the relevance of conditioning on the detector's noise, we also apply our CVAE Bayesian machine to 39 BBH LIGO/Virgo O3 GW events. We find that our PE performance can be compatible with the traditional nested sampling method for SNR larger than some threshold value, and overall is better than the CVAE model without conditioning on the ASDs. We also discuss the possible ways to further improve the PE performance of our CVAE model by constructing more sophisticated ASD models or more complicated neural networks. Finally, we hope that what we have considered in this paper can help to boost the PE tasks by deep learning for future GW events.

ACKNOWLEDGMENTS

This work is supported by Taiwan's Ministry of Science and Technology (MoST) through Grant No. 109-2112-M-003-007-MY3. We thank Guo-Chin Liu for generosity in providing support of the computing facility. We also thank NCTS for partial financial support. Finally, We thank TGWG members for the helpful discussions.

APPENDIX: FITTING A STOCHASTIC ASD MODEL FOR O3a RUN

Here we present the method about how we fix the drifting factor $r[f]$ in (3). We would like to simulate the ASD of O3a and its variations so that we can rewrite (3) into:

$$\log A_{O3a}[f] = \alpha \log r[f] + \log A_m[f], \tag{A1}$$

where $A_{O3a}[f]$ is picked up from the 2000 samples of O3a ASD curves, and $r[f]$ is a function of frequency bins to be fixed. Here we neglect the residual variation β since it introduces a small variation relative to the distribution $\alpha \log r[f]$. On the left side of (A1), we can form a probability density functions (PDF) out of the 2000 samples of $A_{O3a}[f]$, and on the right side, it is a PDF associated with the chi-square distribution α. The overlap between these two PDFs then depends on $\log r[f]$. By maximizing the overlap, we can determine $\log r[f]$ bin by bin. In this way, we obtain our stochastic ASD model (3).

The above construction has assumed the PDF of $\log A_{O3a}[f]$ is nothing but a chi-square distribution, which is however justified by the fact that the maximal overlap is almost perfect, i.e., almost 100%. Besides, the bin-by-bin construction may also ignore the correlations between different bins, such that our ASD model (3) may not capture the full characteristics of O3a's ASD and its variations. Despite that, the fitted $r[f]$ shown in Fig. 3 is indeed a smooth function of frequency. This suggests that the above construction is reasonable.

[1] B. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), Phys. Rev. Lett. **116**, 061102 (2016).
[2] B. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), Phys. Rev. X **9**, 031040 (2019).
[3] R. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), Phys. Rev. X **11**, 021053 (2021).
[4] J. Skilling, AIP Conf. Proc. **735**, 395 (2004).
[5] W. Del Pozzo and J. Veitch, GitHub, https://github.com/johnveitch/cpnest (2015).
[6] J. S. Speagle, Mon. Not. R. Astron. Soc. **493**, 3132 (2020).
[7] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, Publ. Astron. Soc. Pac. **125**, 306 (2013).
[8] T. G. F. Li, Extracting physics from gravitational waves: Testing the strong-field dynamics of general relativity and inferring the large-scale structure of the universe, Ph.D. thesis, Vrije University, Amsterdam, 2013.
[9] B. P. Abbott *et al.* (KAGRA, LIGO Scientific, Virgo Collaborations), Living Rev. Relativity **23**, 3 (2020).
[10] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, Phys. Rev. D **85**, 122006 (2012).
[11] C. Messick, K. Blackburn, P. Brady, P. Brockill, K. Cannon, R. Cariou, S. Caudill, S. J. Chamberlin, J. D. Creighton, R. Everett *et al.*, Phys. Rev. D **95**, 042001 (2017).
[12] J. Veitch, V. Raymond, B. Farr, W. Farr, P. Graff, S. Vitale, B. Aylott, K. Blackburn, N. Christensen, M. Coughlin *et al.*, Phys. Rev. D **91**, 042003 (2015).
[13] C. M. Biwer, C. D. Capano, S. De, M. Cabero, D. A. Brown, A. H. Nitz, and V. Raymond, Publ. Astron. Soc. Pac. **131**, 024503 (2019).
[14] G. Ashton, M. Hübner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divakarla, P. J. Easter, B. Goncharov *et al.*, Astrophys. J. Suppl. Ser. **241**, 27 (2019).
[15] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, Nat. Phys. **18**, 112 (2022).
[16] S. R. Green, C. Simpson, and J. Gair, Phys. Rev. D **102**, 104057 (2020).
[17] D. P. Kingma and M. Welling, Found. Trends Mach. Learn. **12**, 307 (2019).
[18] R. Yu, arXiv:2006.10273.
[19] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, arXiv:1606.04934.
[20] S. R. Green and J. Gair, Mach. Learn. **2**, 03LT01 (2021).
[21] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schlkopf, Phys. Rev. Lett. **127**, 241103 (2021).
[22] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.
[23] M. Vallisneri, J. Kanner, R. Williams, A. Weinstein, and B. Stephens, J. Phys. Conf. Ser. **610**, 012021 (2015).
[24] Gracedb gravitational-wave candidate event database (ligo/virgo o3 public alerts), https://gracedb.ligo.org/superevents/public/O3/.
[25] S. Khan, K. Chatziioannou, M. Hannam, and F. Ohme, Phys. Rev. D **100**, 024059 (2019).
[26] D. P. Kingma and J. Ba, arXiv:1412.6980.
[27] J. Lucas, G. Tucker, R. Grosse, and M. Norouzi, https://openreview.net/forum?id=r1xaVLUYuE (2019).
[28] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, arXiv:1903.10145.
[29] A. Ghasemi and S. Zahediasl, Int. J. Endocrinol. Metab. **10**, 486 (2012).
[30] F. Pérez-Cruz, in *Proceedings of the 2008 IEEE International Symposium on Information Theory* (IEEE, New York, 2008), pp. 1666–1670.