Conditional autoencoder for generating binary neutron star waveforms with tidal and precession effects

Mengfei Sun,[1,2] Jie Wu,[1,2] Jin Li,[1,2,3,*] Brendan Mccane,[4] Nan Yang,[5,2] Xianghe Ma,[1,2] Borui Wang,[6] and Minghui Zhang[7]

[1]*Department of Physics, Chongqing University, Chongqing 401331, People's Republic of China*
[2]*Chongqing Key Laboratory for Strongly Coupled Physics, Chongqing University, Chongqing 401331, People's Republic of China*
[3]*Institute of Advanced Interdisciplinary Studies, Chongqing University, Chongqing 401331, China*
[4]*School of Computing, University of Otago, Otago 9016, New Zealand*
[5]*Department of Electronical Information Science and Technology, Xingtai University, Xingtai 054001, People's Republic of China*
[6]*Department of Earth and Sciences, Southern University of Science and Technology, Shenzhen 518055, People's Republic of China*
[7]*Department of Physics, Southern University of Science and Technology, Shenzhen 518055, People's Republic of China*

(Received 13 April 2025; accepted 16 September 2025; published 7 October 2025)

Gravitational waves from binary neutron star mergers provide insights into dense matter physics and strong-field gravity, but waveform modeling remains computationally challenging. We develop a deep generative model for gravitational waveforms from binary neutron star mergers, covering the late inspiral, merger, and ringdown, incorporating precession and tidal effects. Using the conditional autoencoder, our model efficiently generates waveforms with high precision across a broad parameter space, including component masses (m_1, m_2), spin components $(S_{1x}, S_{1y}, S_{1z}, S_{2x}, S_{2y}, S_{2z})$, and tidal deformability (Λ_1, Λ_2). Trained on 1×10^6 waveforms from the IMRPhenomXP_NRTidalv2 waveform model, our model achieves a mean mismatch of 2.13×10^{-3}. The model accelerates waveform generation. For a single sample, it requires 0.12 s (s), compared to 0.66 s for IMRPhenomXP_NRTidalv2 making it approximately 5 times faster. When generating 1000 waveforms, the network completes the task in 0.75 s, while IMRPhenomXP_NRTidalv2 requires 7.12 s, making it approximately 10 times faster. This speed advantage enables rapid parameter estimation and real-time gravitational wave searches. With higher precision, it will support low-latency detection and broader applications in multimessenger astrophysics.

DOI: 10.1103/kmlw-y7yw

I. INTRODUCTION

Since the first direct detection of the binary black hole (BBH) merger GW150914 by LIGO and Virgo in 2015 [1], gravitational wave (GW) astronomy has entered a new era, enabling direct exploration of extreme astrophysical phenomena. With continuous advancements in detector sensitivity, an increasing number of BBH and binary neutron star (BNS) mergers have been observed [2–4], providing valuable constraints on the equation of state (EOS) of nuclear matter and insights into tidal interactions in neutron stars [2,5]. BNS mergers serve as natural laboratories for testing general relativity in the strong-field regime and probing high-density nuclear matter [6]. In particular, measurements of tidal deformability impose stringent constraints on the nuclear EOS, shedding light on neutron star structure and ultradense matter properties [7,8]. Additionally, multimessenger observations, which combine gravitational waves with electromagnetic counterparts, offer an independent method for measuring cosmological parameters, including the Hubble constant [9,10].

Because of the significance of BNS systems, accurately modeling gravitational waveforms from their mergers is essential for both detection sensitivity and parameter estimation precision. GW searches rely on matched filtering techniques, which require highly accurate waveform templates, while extracting key physical parameters—such as masses, spins, and tidal deformabilities—demands waveform models with high precision. However, modeling BNS waveforms remains challenging due to complex physical effects, particularly spin precession and tidal interactions [11–13]. Waveform modeling has progressed from computationally expensive numerical relativity (NR) simulations [14–16], which solve Einstein's equations

*Contact author: cqujinli1983@cqu.edu.cn

directly, to more efficient semianalytical methods. NR simulations yield high-precision waveforms by capturing strong-field, nonlinear effects but are too costly for large-scale parameter-space studies. Post-Newtonian (PN) approximations [17,18] describe the inspiral phase under weak-field, slow-motion assumptions but lose accuracy near merger. The effective-one-body (EOB) approach [19–21] improves upon PN by mapping the two-body problem to a single-body motion in a modified spacetime, and with NR calibration, balances accuracy and efficiency. Phenomenological models (IMRPhenom) [12,22–24] further enhance efficiency by fitting frequency-domain templates to extensive datasets, enabling rapid waveform generation and facilitating large-scale searches. Despite advancements, challenges remain in computational cost, accuracy, and full parameter-space coverage [25–27], limiting real-time GW detection and precise parameter estimation.

The rapid development of deep learning has introduced an efficient and accurate approach to gravitational waveform modeling [28–31]. With strong nonlinear fitting capabilities and high computational efficiency [32], deep learning enables high-precision waveform generation at significantly reduced cost. George *et al.* [33] first applied deep learning to BBH waveforms and achieved real-time performance beyond traditional methods. Schmidt *et al.* [34] used principal components analysis (PCA) with machine learning to reduce the dimensionality of EOB waveforms and improve efficiency. Dax *et al.* [35] accelerated waveform generation using the JAX framework (a high-performance numerical computing and automatic differentiation library developed by Google) for highly efficient computation and real-time inference. Beyond BBH systems, deep learning has been applied to BNS and EMRI (Extreme Mass Ratio Inspiral) waveform modeling. Whittaker *et al.* [36] used a conditional variational autoencoder (cVAE) to model postmerger signals with EOS uncertainties. Chua *et al.* [37] combined reduced-order modeling with deep learning to accelerate EMRI waveform generation, reducing the cost by over 4 orders of magnitude. These works show that deep learning accelerates waveform generation and handles high-dimensional parameter spaces effectively.

Despite progress in deep learning-based waveform modeling, most existing models focus on BBH systems or simplified BNS mergers, with precession and tidal effects remaining underexplored. To address this, we propose a conditional autoencoder (cAE) model for rapid BNS waveform generation, with applications in GW data analysis. Our model efficiently generates waveforms conditioned on system parameters (Θ), including component masses (m_1, m_2), spin components $(S_{1x}, S_{1y}, S_{1z}, S_{2x}, S_{2y}, S_{2z})$, and tidal deformability (Λ_1, Λ_2), while capturing the high-dimensional evolution of GW signals. Trained on a dataset of 3×10^5 BNS waveforms from the IMRPhenomXP_NRTidalv2 [38] model, it incorporates both precession and tidal effects. To enhance learning efficiency, we adopt the amplitude (A)-phase (Φ) representation, where $h_+(t)$ and $h_\times(t)$ are expressed in terms of amplitude and phase independently, to reduce data oscillation. The cAE architecture employs a dual-encoder structure, separately encoding physical parameters and waveform data, which are mapped in latent space before reconstruction. By relying solely on forward propagation, cAE achieves high acceleration in large-scale waveform generation. Benchmark tests show that generating a single waveform takes 0.12 s with cAE on an NVIDIA A800 80 GB GPU, compared to 0.66 s with IMRPhenomXP_NRTidalv2 on two Intel Xeon Silver 4214R CPUs (24 cores), corresponding to a speedup of about 5 times; for a batch of 10^3 waveforms, cAE needs 0.75 s compared with 7.12 s, yielding roughly 10 times speedup. The model's accuracy is evaluated through waveform overlap calculations, yielding an average mismatch 2.13×10^{-3}, corresponding to accuracy 99.79%. These results demonstrate that the proposed model enables efficient, accurate, and scalable BNS waveform generation with precession and tidal effects, making it well suited for real-time signal detection and parameter estimation.

The structure of the article is as follows: Section II describes the waveform representation and the construction of our dataset. Section III introduces the fundamental concepts of autoencoders and presents the architecture and hyperparameter settings of our neural network. Section IV details the model training and validation process. Section V evaluates the accuracy and generation efficiency of our model. Finally, Sec. VI provides a summary and discusses future research directions.

II. DATA SIMULATION

This study constructs a dataset of simulated BNS gravitational waveforms to train a cAE. The dataset spans a broad range of physical parameters (Θ), including component masses, spins, and tidal deformability, and provides the corresponding amplitude and phase representations. This formulation enhances the efficiency of deep learning models in capturing waveform structures and their dependencies on Θ.

A. Waveform representation

Gravitational waves are typically characterized by two polarization components, h_+ and $h_\times$, expressed as

$$h(t) = h_+(t) + ih_\times(t). \tag{1}$$

However, directly learning $h_+(t)$ and $h_\times(t)$ in the time domain is computationally demanding and may hinder training convergence due to waveform complexity. To improve learning efficiency, we adopt an amplitude-phase representation, where h_+ is treated as the real part and $h_\times$ as

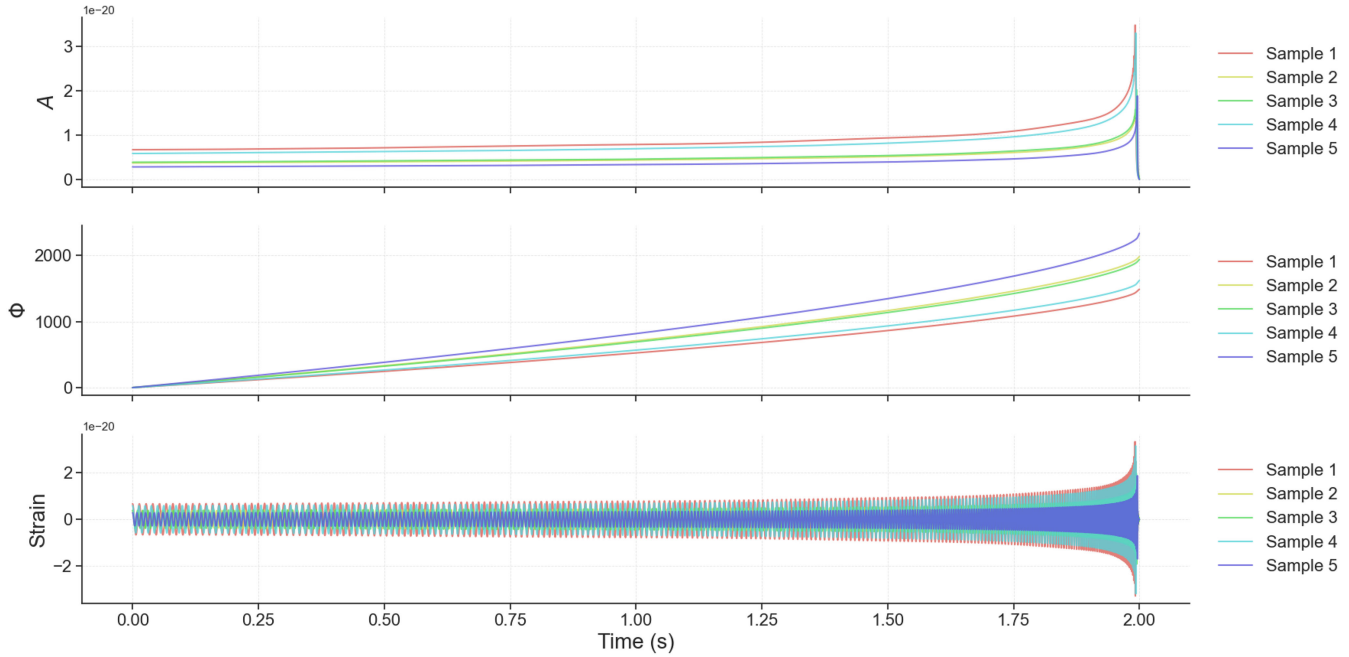

FIG. 1. Input samples in time domain. Top: amplitude curve $A(t)$; middle: phase curve $\Phi(t)$; bottom: waveform strain $h(t)$.

the imaginary part. The corresponding amplitude $A(t)$ and cumulative phase $\Phi(t)$ are given by

$$A(t) = \sqrt{h_+^2 + h_\times^2}, \qquad \Phi(t) = \tan^{-1}\left(\frac{h_\times}{h_+}\right). \quad (2)$$

This representation reduces data oscillatory while enhancing physical interpretability. The amplitude $A(t)$ captures the overall intensity variation of the gravitational wave, while the cumulative phase $\Phi(t)$ describes its temporal evolution, offering a depiction of the underlying dynamics. To further standardize waveform properties, we apply phase normalization,

$$\Phi(t) = \Phi(t) - \Phi(t_0), \quad (3)$$

which aligns all waveforms to an initial phase of zero. This adjustment improves dataset consistency and stabilizes model training by minimizing phase discrepancies across waveforms, facilitating a more effective learning of parameter dependencies in waveform evolution, as shown in Fig. 1 which illustrates the sample examples we used.

B. System parameter selection

Previous studies on gravitational waveform modeling have primarily focused on BBH systems, while investigations of BNS waveforms remain relatively limited. Most existing deep learning models assume binary neutron stars with spins aligned to the orbital angular momentum, and thus do not systematically account for spin precession effects. Additionally, although tidal deformation can significantly affect the phase evolution of BNS waveforms, it is often simplified using point-mass approximations, which may result in the omission of tidal contributions. In this work, we fix the luminosity distance to 1 Mpc, and set both the inclination angle and the coalescence phase to zero. The remaining parameters are listed in Table I, where the spin magnitude ranges are chosen based on the benchmark settings provided in [39].

C. Construction of dataset

We construct a dataset of gravitational waveforms for BNS systems to train and evaluate our conditional autoencoder model. The training set consists of 1×10^6 samples,

TABLE I. Range of the sampling parameters Θ for the BNS training set (with $m_2 < m_1$).

Θ	Description	Range
m_1	Primary mass	Uniform $[1, 3]M_\odot$
m_2	Secondary mass	Uniform $[1, 3]M_\odot$
$\lvert\vec{s}_1\rvert, \lvert\vec{s}_2\rvert$	Spin magnitudes of two neutron stars	Uniform [0, 0.5]
$\vec{s}_1/\lvert\vec{s}_1\rvert, \vec{s}_2/\lvert\vec{s}_2\rvert$	Spin directions (unit vectors)	Isotropic over 3D sphere
Λ_1, Λ_2	Tidal deformabilities of two neutron stars	Uniform [0, 500]

while the test set comprises 1×10^5 samples. Both sets are generated using the IMRPhenomXP_NRTidalv2 waveform model [39], which incorporates precession and tidal effects.

The data generation process includes parameter sampling, waveform computation, preprocessing, and normalization. The sampled parameters Θ include component masses, spin vectors, and tidal deformabilities, with ranges summarized in Table I. Component masses m_1 and m_2 ($m_2 < m_1$) are independently drawn from a uniform distribution over $[1, 3]M_\odot$, with values reordered postsampling to ensure $m_1 > m_2$. The dimensionless spin magnitudes $|\vec{s}_1|$ and $|\vec{s}_2|$ are sampled uniformly in [0, 0.5], and their directions are drawn from an isotropic distribution on the unit sphere to allow for generic spin precession. Tidal deformability parameters Λ_1 and Λ_2 are sampled uniformly in [0, 500].

Both training and test sets are constructed using the same stochastic (nongrid) sampling strategy without predefined step sizes. This randomized approach avoids artifacts introduced by grid-based sampling, enhances waveform diversity, and promotes broad coverage of the high-dimensional parameter space—factors essential for improving generalization and avoiding overfitting in deep learning models. The distributions of sampled parameters in both the training and testing sets are visualized in the Appendix, Figs. 14 and 15, respectively.

The time-domain GW signals $h_+(t)$ and $h_\times(t)$ are computed using PYCBC.WAVEFORM.GET_TD_WAVEFORM [40], followed by trimming to remove leading and trailing zero values. Each waveform is standardized to a fixed duration of 2 s with a sampling rate of 4096 Hz. This segment is taken from the final 2 s before the end of the ringdown, ensuring that the dataset captures the complete merger and ringdown stages while also covering part of the inspiral. The chosen duration is sufficiently long to encompass the stage of prominent tidal effects preceding the merger [41], enabling the model to learn the characteristic evolution of waveforms across different dynamical regimes.

To maintain data consistency, waveforms are to a fixed duration of 2 s with a sampling rate of 4096 Hz. Each waveform segment is taken from the 2 s before the end of the ringdown, ensuring that the dataset captures the complete merger and ringdown stages while also covering part of the inspiral. The two-second simulated waveform is sufficiently long to encompass the stage of tidal effects preceding the merger [41].

The dataset consists of three components: X_{train}, which contains the BNS system's Θ as conditional inputs; y_A, representing the amplitude data; and y_Φ, representing the phase data. During cAE training, we normalize the data to ensure stability and consistency. The X_{train} are processed using min-max normalization [42], which rescales the data to the range [0, 1], ensuring a uniform scale across different parameters and improving training stability:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \tag{4}$$

where X represents the original data, $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of each parameter, and X' is the normalized data. For y_A and y_Φ, we apply standardization [43], which ensures zero mean and unit variance to eliminate scale differences and enhance training stability:

$$X' = \frac{X - \mu}{\sigma}, \tag{5}$$

where μ and σ denote the mean and standard deviation of the data, respectively. Since the amplitude and phase exhibit distinct evolution patterns, two separate cAEs are trained: one for learning a compact representation and reconstruction of y_A, and another for modeling y_Φ. During training, X_{train} is provided as a conditional variable to the autoencoder, ensuring that the latent representation z effectively captures the dependence of waveforms on Θ.

III. AUTOENCODER AND CONDITIONAL AUTOENCODER

The autoencoder (AE) is an unsupervised learning model widely used for data dimensionality reduction, feature extraction, and generative tasks. It consists of an encoder and a decoder, learning a low-dimensional representation of the data by minimizing reconstruction error. The variational autoencoder (VAE) extends this model by introducing probabilistic modeling, enforcing a smoother latent variable distribution, which enhances the generative capability. The cAE further incorporates external conditional constraints, enabling the model to generate samples corresponding to specific data distributions based on input conditions, making it particularly relevant for GW waveform modeling.

This section first introduces the fundamental concepts of AE, VAE, and cAE, discussing their applicability to GW waveform generation. Subsequently, we provide a detailed description of the proposed cAE-based waveform generation model, including the separate cAE architectures designed for phase and amplitude modeling, along with their respective hyperparameter settings.

A. Concepts of autoencoder and conditional autoencoder

We employ autoencoders (AEs) [44] to reduce the dimensionality of complex GW waveforms (amplitude A and phase Φ) while ensuring accurate reconstruction. As shown in Fig. 2(a), an AE consists of an input $h^{(i)}$ (A or Φ), an encoder $q_\alpha(z|h)$, a latent variable $z^{(i)} \in \mathbf{R}^d$, and a decoder $p_\beta(\hat{h}|z)$. The encoder projects the input (A or Φ) into a lower-dimensional latent space $z^{(i)}$, where the dimension d of $z^{(i)}$ can be adjusted based on specific task requirements. The decoder then reconstructs the $\hat{h}^{(i)}$ from

the latent representation through an upsampling process. The α and β refers to the learned model parameters, such as weights and biases, obtained after training. To measure the similarity between the reconstructed $\hat{h}^{(i)}$ and the target $h^{(i)}$, the AE employs the mean squared error (MSE) as the reconstruction loss:

$$L_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|h^{(i)} - \hat{h}^{(i)}\|^2, \tag{6}$$

where N is the total number of training samples, $h^{(i)}$ represents the target (A or Φ), and $\hat{h}^{(i)}$ is the reconstructed A or Φ. By minimizing L_{MSE}, the model updates its weights and biases, ensuring that $\hat{h}^{(i)}$ closely approximates $h^{(i)}$. Unlike traditional linear methods such as PCA [45], autoencoders (AEs) can capture the nonlinear features of GW signals more effectively [46]. While PCA is efficient for simple signals, its linear projections may miss important features in the nonlinear phases of GW evolution [47,48]. In contrast, AEs reduce dimensionality through nonlinear mappings, preserving key physical features such as orbital dynamics, tidal effects, and ringdown. This enables better generalization, parameter recovery, and interpolation across the waveform space [49,50]. Figure 2(b) illustrates the structure of a VAE [51]. Unlike standard AEs, VAEs introduce probabilistic modeling between the encoder and decoder, ensuring that the latent variable $z^{(i)}$ is not a fixed deterministic value but is instead sampled from a distribution defined by the encoder's output mean $\mu_\alpha(h^{(i)})$ and variance $\sigma_\alpha^2(h^{(i)})$. Specifically, VAEs utilize the reparametrization trick to obtain latent variables:

$$z^{(i)} = \mu_\alpha(h^{(i)}) + \sigma_\alpha(h^{(i)}) \cdot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

This approach allows gradients to propagate through the sampling process, making it possible to optimize the network using gradient-based methods. The VAE training objective consists of the reconstruction loss and the Kullback-Leibler (KL) divergence loss [52,53]. The reconstruction loss measures the difference between the decoder's output $\hat{h}^{(i)}$ and the input waveform $h^{(i)}$, typically computed using the negative log-likelihood:

$$L_{\mathrm{recon}} = \mathbb{E}_{q_\alpha(z|h)}[-\log p_\beta(\hat{h}|z)]. \tag{8}$$

The KL divergence loss ensures that the learned latent variable distribution $q_\alpha(z|h)$ approximates a predefined prior distribution, typically a standard normal distribution $p(z) = \mathcal{N}(0, I)$, where I is the identity matrix:

$$L_{\mathrm{KL}} = D_{\mathrm{KL}}(q_\alpha(z|h)\|p(z)). \tag{9}$$

The final VAE objective function is given by

$$L_{\mathrm{VAE}} = L_{\mathrm{recon}} + \kappa L_{\mathrm{KL}}, \tag{10}$$

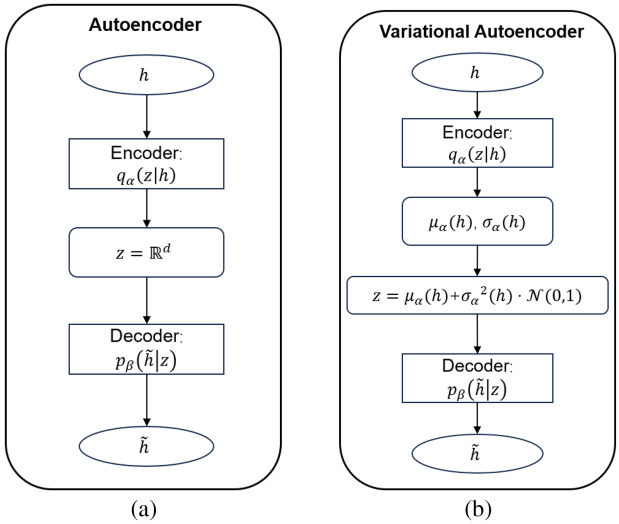

FIG. 2. (a) Structure of an AE. (b) Structure of a variational VAE.

where the hyperparameter κ controls the weight of the KL divergence loss, regulating the structure of the latent space.

Although traditional AEs and VAEs perform well in capturing the low-dimensional structure and nonlinear features of data, their generative processes typically rely solely on the data itself. As a result, they lack the capacity to explicitly incorporate known physical priors into the latent representations. In other words, standard AE/VAE models in unsupervised learning tend to capture the dominant variations in the data, but they cannot guarantee that the generated waveforms strictly adhere to physical constraints. To address this limitation and further enhance the physical interpretability and controllability of waveform generation, we use the cAE [54]. In the cAE model, additional physical parameters Θ (as shown in Table I) are incorporated as conditional inputs and jointly mapped with waveform data into a low-dimensional latent space. In this manner, the cAE not only inherits the advantages of AE/VAE in nonlinear dimensionality reduction and data reconstruction, but also enables the explicit embedding of physical constraints into the latent variables, thereby generating waveforms that better reflect realistic astrophysical properties.

In the following, we provide a detailed description of our cAE model architecture.

B. Architecture and hyperparameters of the model

Our study employs the cAEs to model the amplitude and phase of GW waveforms from BNS mergers. The model consists of two independent cAEs, each responsible for learning a low-dimensional representation of either the amplitude y_A or phase y_Φ and reconstructing waveforms conditioned on Θ. In our implementation, the dimensionality of the latent representation is set to 300. Each cAE comprises two encoders (encoder 1 and encoder 2) and

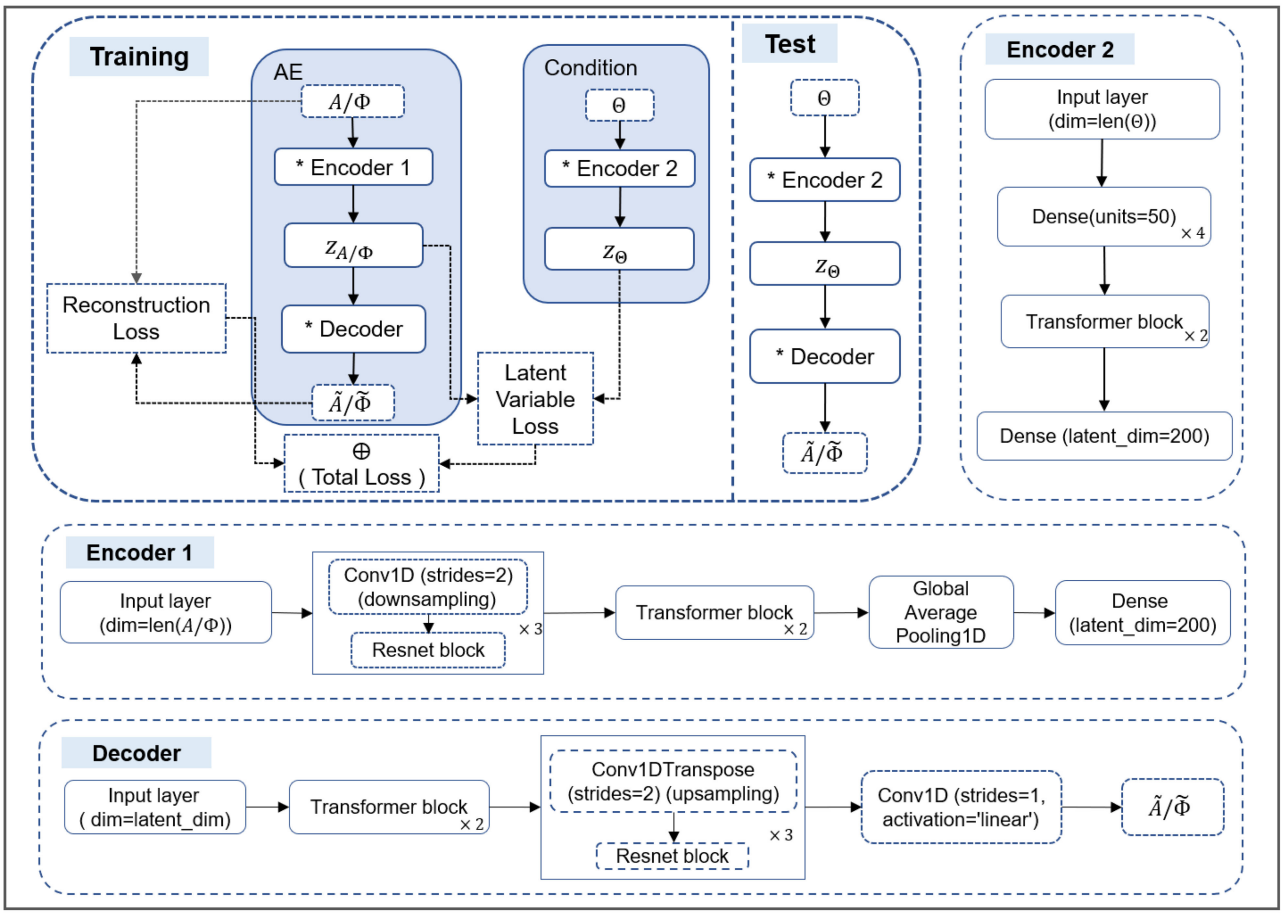

FIG. 3. Training and testing architecture of our model, along with detailed structures of individual components, the structure of the Transformer and ResNet block are shown in the Appendix (Fig. 12).

a decoder. Encoder 1 processes the amplitude or phase data, while encoder 2 encodes the Θ, and the two latent representations are combined in the latent space before being mapped back to a complete waveform by the decoder. The overall model architecture is shown in Fig. 3, where training and test sections correspond to the training and inference workflows, while the remaining sections detail the structural components.

As shown in Fig. 3, our model consists of two main branches: one for waveform encoding and reconstruction, and one for encoding Θ. The goal is to align latent representations from both branches while ensuring accurate waveform reconstruction.

Encoder 1 is used independently for amplitude and phase inputs. Each encoder processes a normalized 1D data through a Conv1D layer with stride 2, followed by three ResNet blocks for local feature extraction and two Transformer blocks for capturing long-range dependencies. A global average pooling layer compresses the temporal dimension, and a final dense layer maps the features into a latent space (z_A or z_Φ).

Encoder 2 takes Θ (Table I) as input. These are passed through four fully connected layers, followed by two Transformer blocks. The output is projected into the same latent space as encoder 1, producing z_Θ.

Decoder takes the latent variable from encoder 1 and reconstructs the waveform. It expands the latent dimension via a dense layer, then applies two Transformer blocks and three upsampling ResNet blocks using transposed Conv1D layers, recovering the waveform shape.

Latent variables and loss functions: During training, both the waveform and the Θ are encoded into their respective latent representations, denoted as $z_{A/\Phi}$ and z_Θ. To ensure that the decoder can accurately reconstruct the input waveform and that both latent spaces are aligned, we define two loss components: First, the *reconstruction loss* is computed as the mean absolute error (MAE) between the input waveform x and the reconstructed waveform $\hat{h}$:

$$L_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \|h^{(i)} - \hat{h}^{(i)}\|_{\text{MAE}}. \tag{11}$$

Second, the *latent consistency loss* penalizes the difference between the latent vector produced by encoder 1 (A/Φ) and encoder 2 (Θ):

$$L_{\text{latent}} = \frac{1}{N}\sum_{i=1}^{N} \|z_{A/\Phi}^{(i)} - z_{\Theta}^{(i)}\|_{\text{MSE}}^2. \tag{12}$$

The total loss is then defined as a weighted sum of the reconstruction and latent consistency losses, as shown in Eq. (13):

$$L_{\text{total}} = L_{\text{rec}} + \lambda L_{\text{latent}}, \tag{13}$$

where λ is a balancing coefficient. In our study, we set $\lambda = 1$ to equally weight reconstruction accuracy and latent alignment.

With the model architecture defined, we describe the training procedure and validation setup used to evaluate the model's performance in the following section.

IV. TRAINING AND VERIFICATION

A. Training

Amplitude and phase models are trained separately on an NVIDIA A800 GPU with 80 GB of memory. The full training process, including both models, takes approximately ten days. The training dataset consists of 1×10^6 waveform samples, split into 90% for training and 10% for validation. Before training, input parameters and target waveforms are normalized using min-max and standard scaling, respectively. Data is preprocessed into ten blocks and fed into the model with a batch size of 128. Figure 4 shows the training and validation loss curves for both amplitude and phase models.

Our model is trained using the Adam optimizer with an initial learning rate of 10^{-4}. To enhance convergence and prevent stagnation, a learning rate scheduling strategy is employed: if the validation loss does not improve for seven consecutive epochs, the learning rate is reduced by a factor of 0.7, with a floor value of 10^{-8}. In addition, early stopping is activated when the validation loss shows no improvement over 15 consecutive epochs, and the model is restored to the state with the best validation performance.

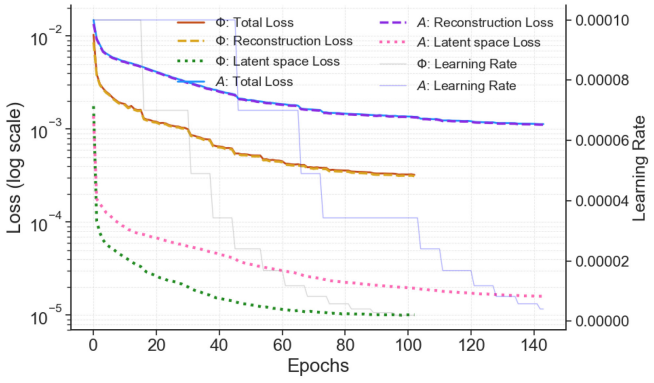

FIG. 4. Training and validation loss curves for the amplitude and phase models.

B. Verification

To evaluate the precision of cAE-generated waveforms, we compute the mismatch between the model-generated waveforms and the IMRPhenomXP_NRTidalv2 waveforms. The evaluation is based on two metrics: overlap [55,56] and mismatch, which quantify the waveform reconstruction quality. The analysis is conducted in the frequency domain after applying a Fourier transform to the time-domain waveforms, allowing for a more effective comparison of waveform similarities. In GW data analysis, the inner product of two waveforms is typically defined as a noise-weighted integral over the frequency domain, incorporating the power spectral density $S_n(f)$:

$$\langle h_1|h_2\rangle = 4\text{Re}\int_{f_{\min}}^{f_{\max}} \frac{\tilde{h}_1(f)\tilde{h}_2(f)}{S_n(f)}df, \tag{14}$$

where $\tilde{h}_1(f)$ and $\tilde{h}_2(f)$ are the Fourier transforms of $h_1(t)$ and $h_2(t)$, respectively, and $S_n(f)$ represents the power spectral density of the detector noise. This weighted inner product provides a measure of how well two waveforms match in the presence of detector noise. We set $S_n(f)$ to the LIGO O4 sensitivity curve [57].

To eliminate the influence of normalization, each waveform is rescaled to satisfy the unit-norm condition:

$$\hat{h}(t) = \frac{h(t)}{\sqrt{\langle h|h\rangle}}. \tag{15}$$

The overlap between two waveforms is then computed by maximizing the inner product over different time shifts t_c and phase shifts ϕ_c:

$$O(h_1, h_2) = \max_{t_c,\phi_c} \frac{\langle \hat{h}_1|\hat{h}_2\rangle}{\sqrt{\langle \hat{h}_1|\hat{h}_1\rangle\langle \hat{h}_2|\hat{h}_2\rangle}}. \tag{16}$$

Based on this overlap metric, the mismatch is defined as

$$M(h_1, h_2) = 1 - O(h_1, h_2). \tag{17}$$

Here, $M(h_1, h_2)$ quantifies the dissimilarity between the two waveforms, where lower values indicate a higher similarity between the model-generated waveforms and the target physical waveforms.

V. RESULTS AND ANALYSIS

In this section, we not only compare our proposed conditional autoencoder with traditional waveform approximants, but also with a recent deep learning architecture [58], which employs a residual-stacked multilayer perceptron (MLP) and convolutional neural network (CNN). Traditional waveform approximants are included as baseline references in both the mismatch analysis and the

generation time comparison. We present the generation precision and generation speed of both the cAE and the residual-stacked MLP-CNN model, using the traditional models as a benchmark. In addition, we further analyze the latent space to examine whether physically meaningful correlations are preserved when varying parameters. Moreover, we perform a tidal phasing consistency test by varying tidal deformabilities (Λ_1, Λ_2) with fixed masses and spins, and compare the cumulative phase evolution against the IMRPhenomXP NRTidalv2 model.

A. Mismatch evaluation

To evaluate the accuracy of waveform reconstruction between neural network models and traditional waveform approximants, we compare our conditional autoencoder model with the residual-MLP-CNN model, whose architecture is reproduced from [58] and shown in Fig. 13 of the Appendix. Both models are trained using the same dataset and identical preprocessing pipelines, following the same procedure used for our cAE model. The training strategy, including optimizer, learning rate scheduler, early stopping, and callbacks, is also kept consistent with that of the cAE model. Using the test set introduced in Sec. II C, we evaluate waveform reconstruction accuracy using the mismatch. Figure 5 shows the mismatch distributions of h_+ and $h_\times$ for both models. The average mismatch is calculated as $\text{mismatch}_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} \text{mismatch}_i$. The average mismatch of the cAE model is 2.13×10^{-3}, while that of the residual-MLP-CNN model is 5.72×10^{-3}, indicating that the cAE achieves higher reconstruction accuracy.

We analyze the relationship between the number of orbital cycles and the mismatch by computing the cycle count for the test set. The time-domain signal $s(t)$ undergoes a Hilbert transform to obtain its analytic representation [59]:

$$s_a(t) = s(t) + i\mathcal{H}\{s(t)\}, \tag{18}$$

where $\mathcal{H}\{\cdot\}$ denotes the Hilbert transform. The instantaneous phase is then extracted from the analytic signal:

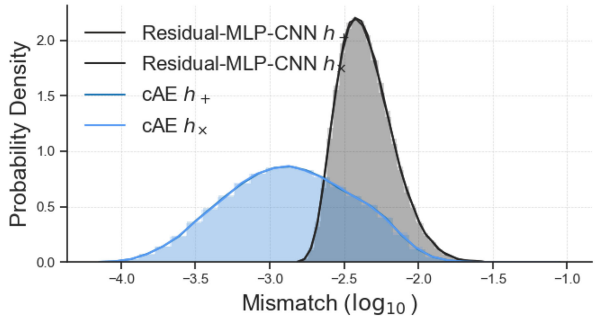

FIG. 5. Mismatch distributions of h_+ and $h_\times$ for both the cAE and residual-MLP-CNN models over the test set.

$$\phi(t) = \arg(s_a(t)). \tag{19}$$

To eliminate phase discontinuities, we apply phase unwrapping to obtain a monotonic phase function $\tilde{\phi}(t)$ and compute the total phase difference:

$$\Delta\phi = \tilde{\phi}(T) - \tilde{\phi}(0). \tag{20}$$

Finally, the orbital cycle number is given by

$$\text{Cycles} = \frac{\Delta\phi}{2\pi}. \tag{21}$$

Figure 6 illustrates how the waveform mismatch varies with the number of orbital cycles in the test set. Across a wide range of cycles (200–400), there is no clear trend of mismatch increasing with cycle count. In particular, our cAE exhibits consistently low mismatch values, indicating robust learning performance. The residual-MLP-CNN model also maintains stable mismatch across the cycle range, though its overall mismatch is higher than that of cAE. These results demonstrate that, given a sufficiently large training dataset (1×10^6 samples), both models are capable of learning accurate waveform representations without degradation at higher cycle counts. Compared to residual-MLP-CNN, cAE achieves lower mismatch over

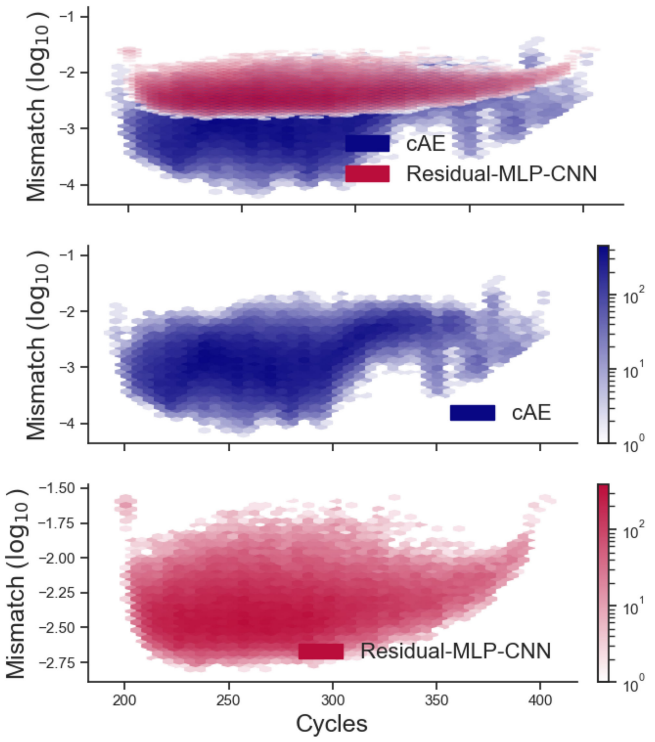

FIG. 6. The horizontal axis shows the number of orbital cycles in the test set, and the vertical axis shows the mismatch. Color indicates sample density.

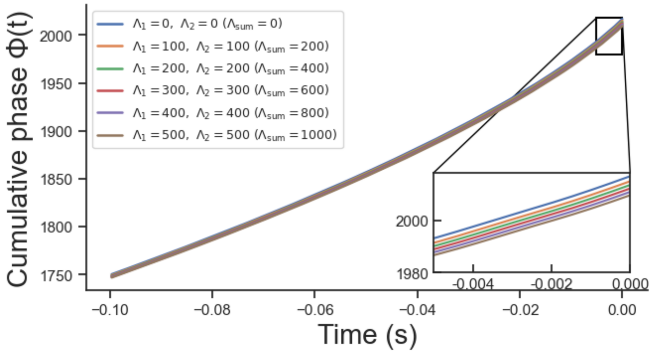

FIG. 7. Cumulative phase $\Phi(t)$ evolution for cAE-generated waveforms under different tidal deformabilities (Λ_1, Λ_2). The inset shows the enlargement around the merger time.

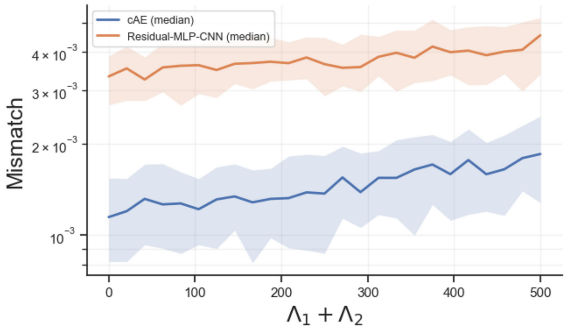

FIG. 9. Mismatch between cAE and IMRPhenomXP_NRTidalv2 waveforms across different values of total tidal deformability $\Lambda_\Sigma = \Lambda_1 + \Lambda_2$. The shaded regions indicate the spread of mismatch values within approximately 1 standard deviation around the median at each total tidal deformability Λ_Σ.

nearly the entire cycle range, further highlighting its generalization capacity.

To further evaluate the physical consistency of our cAE model, we conduct a tidal phasing consistency test. In this test, both component masses are fixed at $1.4M_\odot$, and the spins are set to zero for both stars, while the tidal deformabilities (Λ_1, Λ_2) are varied. We analyze the cumulative phase evolution $\Phi(t)$ of IMRPhenomXP_NRTidalv2 waveforms under different tidal deformability parameters. Figure 7 shows the cumulative phase $\Phi(t)$ over time for several representative choices of (Λ_1, Λ_2). As the total tidal deformability $\Lambda_\Sigma = \Lambda_1 + \Lambda_2$ increases, the resulting phase curves exhibit consistent upward shifts, reflecting the acceleration of phase evolution due to stronger tidal effects.

In Fig. 8, we present the phase difference $\Delta\Phi(t)$ between waveforms generated by our cAE model and the IMRPhenomXP_NRTidalv2. As time increases, the cumulative phase difference gradually grows, reflecting the accumulation of small discrepancies during the inspiral evolution. Nevertheless, the absolute magnitude of $\Delta\Phi(t)$ remains within a relatively small range throughout the entire duration time, indicating that the cAE maintains high accuracy in phase reconstruction. Importantly, when keeping the component masses and spins fixed and varying only

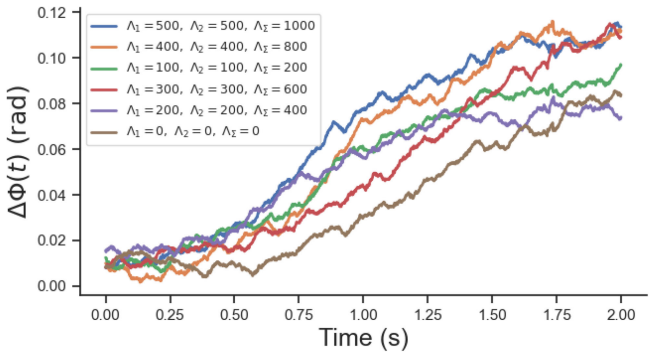

FIG. 8. Phase difference $\Delta\Phi(t)$ between the cAE model and IMRPhenomXP_NRTidalv2 over time for different total tidal deformabilities Λ_Σ.

the tidal deformabilities (Λ_1, Λ_2), we observe a small upward trend in the mismatch as Λ_Σ increases. This behavior is physically consistent, since stronger tidal effects lead to larger phase shifts in binary neutron star waveforms. However, the mismatch stays at a low level, showing that our cAE can model tidal effects reliably and accurately.

To further verify the robustness of the cAE in the strong tidal regime, we examined how the waveform mismatch changes with the total tidal deformability $\Lambda_\Sigma = \Lambda_1 + \Lambda_2$, and compared it with the residual-MLP-CNN model. As shown in Fig. 9, both models exhibit a small increasing trend in mismatch as Λ_Σ grows, which is consistent with the cumulative phase shifts induced by tidal effects. However, the cAE maintains its mismatch at a consistently lower level (below 2×10^{-3}).

B. Latent space analysis

To more systematically examine whether the proposed cAE model captures physically meaningful structures within its latent representation, we have added three controlled experiments, with the corresponding results shown in Fig. 10. Specifically, in each experiment, we vary only a kind of physical parameter while keeping the others fixed to observe the model's latent vector response: (a) we fix the component masses $m_1 = m_2 = 1.4M_\odot$ and spins $\chi_{1z} = \chi_{2z} = 0$, and gradually increase the tidal deformability $\Lambda_1 = \Lambda_2$; (b) we fix $\Lambda_1 = \Lambda_2 = 0$ and $\chi_{1z} = \chi_{2z} = 0$, and vary the total mass $M_{\rm tot}$; (c) we fix the masses and tidal parameters, and vary the total spin $\chi_{\rm tot}$. For each configuration, we input the physical parameters into the cAE and extract the latent vector $z \in \mathbb{R}^{300}$ from the encoder trained on waveform phase.

To compare how each physical variation affects the latent dimensions, we compute the difference $\Delta z = z_i - z_0$ for each sample relative to the baseline (i.e., the first sample in each set with the smallest parameter value), and sort the latent dimensions by their overall sensitivity. This ensures

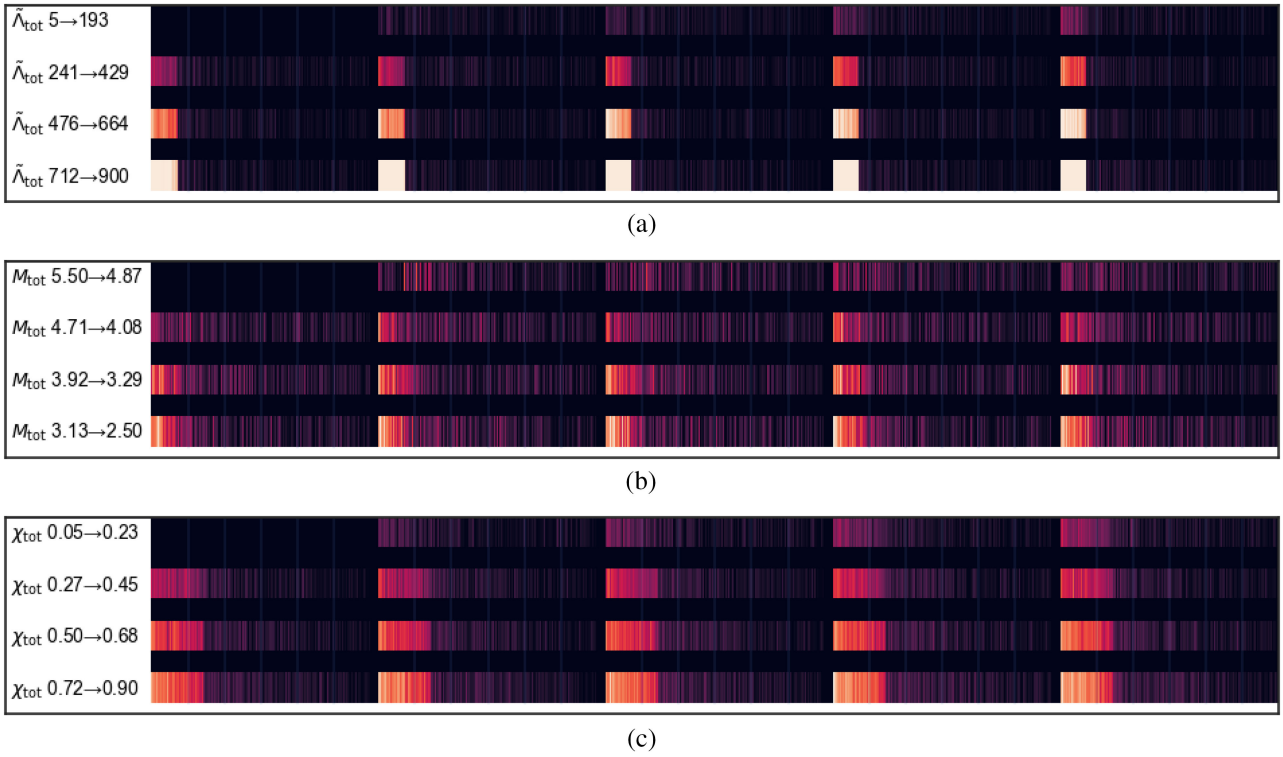

FIG. 10. Latent space analysis of the cAE model under controlled variations of physical parameters. Each part contains 20 samples arranged in a 4×5 grid, where each row contains five waveform configurations. Each sample is represented by a horizontal strip of 300 cells, corresponding to the 300-dimensional latent vector output by the encoder. Each column thus represents a specific latent dimension across all samples. The color of each cell indicates the relative activation magnitude Δz compared to a baseline (the first sample in each part), with brighter colors representing stronger positive deviations. The baseline row appears uniformly dark by construction. All parts share a unified color scale for consistent visual comparison. Experimental setups: (a) varying tidal deformabilities ($\Lambda_1 = \Lambda_2$) with fixed component masses ($m_1 = m_2 = 1.4M_\odot$) and spins ($\chi_{1z} = \chi_{2z} = 0$); (b) varying total mass with fixed spins ($\chi_{1z} = \chi_{2z} = 0$) and tidal deformabilities ($\Lambda_1 = \Lambda_2 = 0$); and (c) varying total spin $\chi_{\rm tot}$ with fixed component masses and tidal deformabilities.

that dimensions with the most pronounced changes are shown on the left in each figure. Each figure displays 20 samples arranged in a 4×5 grid, where each row contains five samples corresponding to different physical configurations. For each sample, we obtain a 300-dimensional latent vector from the encoder (trained on waveform phases). The first sample (top-left corner) serves as the baseline, and the remaining 19 samples are compared against it. Specifically, we compute the absolute difference $\Delta z = |z_i - z_0|$ between each sample's latent vector z_i and the baseline vector z_0. Each column represents a latent dimension, and the color intensity reflects the magnitude of Δz in that dimension. Brighter colors indicate stronger deviation from the baseline. In addition, a brighter color can be interpreted as a stronger activation of the corresponding latent neuron. A unified color scale is used across all parts to ensure consistency in visual interpretation.

The results show interpretable correlations between physical parameters and latent responses. For example, in the tidal variation experiment, only a small subset of latent dimensions exhibits significant and stable increases in activation, suggesting that the cAE has learned to encode tidal effects—which primarily impact late-inspiral phase evolution—into a compact subspace [see Fig. 10(a)]. In the mass variation experiment, lower total mass leads to both stronger and more widespread activations, consistent with the fact that lower-mass binaries produce longer inspirals with richer phase features [see Fig. 10(b)]. In contrast, for spin variation, the response is more diffuse across many latent dimensions, reflecting the more complex and nonlocal influence of spin, which affects both phase evolution and precession modulation [see Fig. 10(c)].

In the Appendix, Figs. 16–19 illustrate several examples of reconstructed waveforms generated by our conditional autoencoder model for test cases with different physical parameters.

C. Waveform generation efficiency

We evaluate waveform generation speed for three categories of models: (i) our conditional autoencoder, (ii) the reproduced residual-MLP-CNN, and (iii) four CPU-based approximants (SpinTaylorT1, IMRPhenomPv2_NRTidal, IMRPhenomPv2_NRTidalv2, and NRTidalv2). All timings

TABLE II. Time to generate a single waveform. Entries marked with an asterisk (*) denote AI-based models. GPU-based results are obtained on an NVIDIA A800 80 GB, while CPU results are from two Intel Xeon Silver 4214R. Traditional approximants are evaluated on CPU only.

Waveform	Time (s)
*cAE (GPU)	0.1204
*cAE (CPU)	0.2314
*Residual-MLP-CNN (GPU)	0.1221
*Residual-MLP-CNN (CPU)	0.1675
SpinTaylorT1	0.9415
IMRPhenomPv2_NRTidal	1.1026
IMRPhenomPv2_NRTidalv2	0.8383
IMRPhenomXP_NRTidalv2	0.6663

were obtained on an NVIDIA A800 80 GB GPU and two Intel Xeon Silver 4214R CPU (24 cores).

Single waveform. Table II shows that our cAE requires 0.1204 s on GPU and 0.2314 s on CPU to generate a single waveform. Compared to traditional CPU-based approximants, the cAE achieves a speedup of approximately 5–9 times on GPU and remains slightly faster on CPU. When compared to the residual-MLP-CNN, the two models exhibit nearly identical performance on GPU (0.1204 vs 0.1221 s), and similarly on CPU (0.2314 vs 0.1675 s), indicating broadly consistent generation times across both architectures. Overall, both neural models significantly outperform traditional approximants in single waveform generation time.

Batched generation. Table III presents waveform generation times for batch sizes ranging from 1 to 1000. The conditional autoencoder (cAE) and residual-MLP-CNN exhibit similar performance across all batch sizes. On GPU, their generation times are nearly identical: at batch size 1, cAE–GPU requires 0.1204 s, compared to 0.1221 s for residual-MLP–GPU; at batch size 1000, the times are 1.0210 and 1.0493 s, respectively. On CPU, their run times also remain close, for example, at batch size 50, cAE–CPU takes 0.6773 s while residual-MLP–CPU takes 0.5567 s.

Compared to the traditional approximant IMRPhenomXP_NRTidalv2 running on CPU, both neural models executed on GPU yield speedups. For batch size 1, cAE–GPU is approximately 5.54 times faster (0.1204 vs 0.6663 s), and for batch size 1000, the speedup increases to 9.92 times (1.0210 vs 10.1279 s). Residual-MLP–GPU shows a similar trend, reaching 9.65 times faster at batch size 1000.

Bayesian inference efficiency. To further demonstrate the practical utility of our cAE model in downstream applications, we performed a simplified Bayesian inference experiment on the binary neutron star masses (m_1, m_2). The injected (true) source parameters were set to $m_1^{\rm true} = 1.45M_\odot$, $m_2^{\rm true} = 1.25M_\odot$, with dimensionless spins $\chi_{1z} = \chi_{2z} = 0.1$ and tidal deformabilities $\Lambda_1 = \Lambda_2 = 200$. The detector noise was modeled using the aLIGO O4 sensitivity curve, and posterior sampling was performed with the EMCEE package [60]. We employed four chains of 5000 steps each, discarding the first 1000 steps as burn-in. Since the purpose of this experiment is primarily to compare the efficiency of waveform templates, rather than to perform full high-dimensional parameter estimation, we fixed the spin and tidal parameters and inferred only (m_1, m_2). This simplification is sufficient to reflect the computational cost and efficiency differences between traditional approximants and the deep-learning cAE templates.

The inference results are shown in Fig. 11. For the cAE template, the estimated parameters are $m_1 = 1.469$ with a standard deviation of 0.075 ($m_1 = 1.469 \pm 0.075M_\odot$) and $m_2 = 1.245$ with a standard deviation of 0.060 ($m_2 = 1.245 \pm 0.060M_\odot$), with 68% credible intervals of (1.387,1.550) and (1.178,1.312), respectively. For the traditional approximant, the estimates are $m_1 = 1.530$ with a standard deviation of 0.078 ($m_1 = 1.530 \pm 0.078M_\odot$) and $m_2 = 1.191$ with a standard deviation of 0.059 ($m_2 = 1.191 \pm 0.059M_\odot$), with 68% credible intervals of (1.427,1.607) and (1.134,1.353), respectively.

The results suggest that the cAE model delivers parameter estimates with accuracy on par with the IMRPhenomXP_NRTidalv2 approximant. In terms of run-time, the cAE model provides a meaningful

TABLE III. Waveform generation time for different batch sizes. Residual-MLP-CNN is abbreviated as ResMC. ResMC–GPU and ResMC–CPU refer to ResMC evaluated on an A800 GPU and on two Xeon Silver 4214R CPUs, respectively. Similarly, cAE–GPU and cAE–CPU denote our conditional autoencoder model evaluated on GPU and CPU. IMR–CPU refers to IMRPhenomXP_NRTidalv2 evaluated on the same CPU system. The speedup ratios are defined as $\mathrm{SU(IMR/cAE-G)} = \mathrm{t(IMR-CPU)/t(cAE-GPU)}$ and $\mathrm{SU(IMR/ResMC-G)} = \mathrm{t(IMR-CPU)/t(ResMC-GPU)}$, where t denotes the time.

Batch size	cAE–GPU	cAE–CPU	ResMC–GPU	ResMC–CPU	IMR–CPU	SU(IMR/cAE–G)	SU(IMR/ResMC–G)
1	0.1204 s	0.2314 s	0.1221 s	0.1675 s	0.6663 s	5.54	5.46
10	0.1500 s	0.3235 s	0.1318 s	0.2797 s	0.7113 s	4.74	5.40
50	0.1663 s	0.6773 s	0.1726 s	0.5567 s	1.0646 s	6.40	6.17
100	0.2501 s	1.2440 s	0.2291 s	1.1831 s	1.1377 s	4.55	4.97
500	0.5902 s	5.8146 s	0.5927 s	5.5440 s	5.2263 s	8.86	8.82
1000	1.0210 s	11.7420 s	1.0493 s	11.1276 s	10.1279 s	9.92	9.65

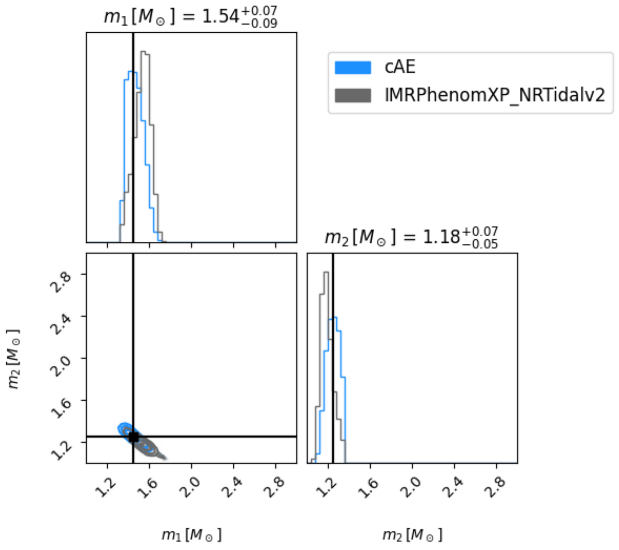

FIG. 11. Posterior distributions of binary neutron star component masses (m_1, m_2) obtained from Bayesian inference.

improvement in computational efficiency. Under the same sampling setup, the traditional approximant required approximately 312 s using 24 CPU cores, while the cAE completed the inference in only 61 s on a single NVIDIA A800 GPU—resulting in a speedup of approximately 5.1 times. These findings provide initial evidence that the cAE model not only maintains high waveform reconstruction fidelity but also offers significant computational benefits in practical inference workflows. This makes it particularly well suited for time-sensitive applications such as rapid alerts or large-scale population analyses. We hope this experiment offers a useful reference point for future extensions to more complex, higher-dimensional inference scenarios.

VI. SUMMARY AND DISCUSSION

This study presents an efficient gravitational waveform generation method based on a cAE and applies it to amplitude-phase modeling of BNS systems. Compared to traditional waveform approximation methods such as IMRPhenomXP_NRTidalv2, cAE significantly improves computational efficiency while maintaining high reconstruction accuracy. On a large-scale test dataset, the averaged waveform mismatch is 2.13×10^{-3}, corresponding to an average accuracy exceeding 99.79%. Even with precession and tidal effects, cAE maintains high precision across different Θ ranges. For efficiency, the cAE model also performs well. On GPU, it takes 0.1204 s to generate a single waveform, while IMRPhenomXP_NRTidalv2 on CPU takes 0.6663 s, giving a speedup of about 5.54 times. As the batch size increases, this advantage becomes more clear. For batch size 1000, cAE–GPU takes 1.0210 s, while IMRPhenomXP_NRTidalv2 on CPU takes 10.1279 s, giving a speedup of about 9.92 times. Residual-MLP–GPU shows similar results across all batch sizes. These results suggest that the cAE model maintains both high precision and relatively higher efficiency in waveform generation, especially when running on GPU.

Although this study has made some progress, several aspects remain worthy of further exploration. Future work can improve waveform reconstruction accuracy and generation efficiency by expanding the training dataset and incorporating more advanced model architectures. Generation speed is also a key factor, particularly critical for real-time gravitational wave applications. We plan to explore inference optimization strategies such as TensorRT [61,62] and ONNX Runtime [63] to further reduce latency and enhance practical applicability. In addition, we will systematically evaluate the impact of latent variable dimensionality on waveform modeling and extend the parameter space to cover more complex physical systems, such as eccentric binaries and sources with stronger spin and tidal effects. On the architectural side, future work may explore diffusion models [64], Transformer variants [65], and variational autoencoders [VAEs, see Fig. 2(b)] to enhance generation performance and improve generalization.

In conclusion, the cAE-based waveform generation method proposed in this study offers an efficient and accurate approach to BNS waveform modeling. It shows strong potential for real-time data analysis, large-scale parameter estimation, and GW event identification. As deep learning continues to advance, data-driven methods are expected to play a growing role in GW astronomy, providing more precise and computationally efficient tools for signal modeling and fundamental physics research.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (Grant No. 2021YFC2203004), the Fundamental Research Funds for the Central Universities Project (Grant No. 2024IAIS- ZD009), the National Natural Science Foundation of China (Grants No. 12575072 and No. 12347101), and the Natural Science Foundation of Chongqing (Grant No. CSTB2023NSCQ-MSX0103). The source code of this study is available [66].

DATA AVAILABILITY

The data supporting this study's findings are available within the article.

APPENDIX

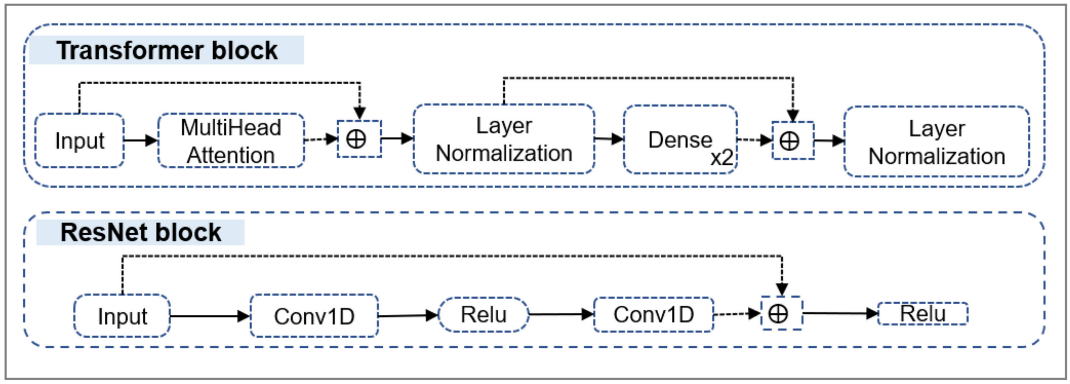

FIG. 12. Architectures of the Transformer and ResNet blocks used in our model.

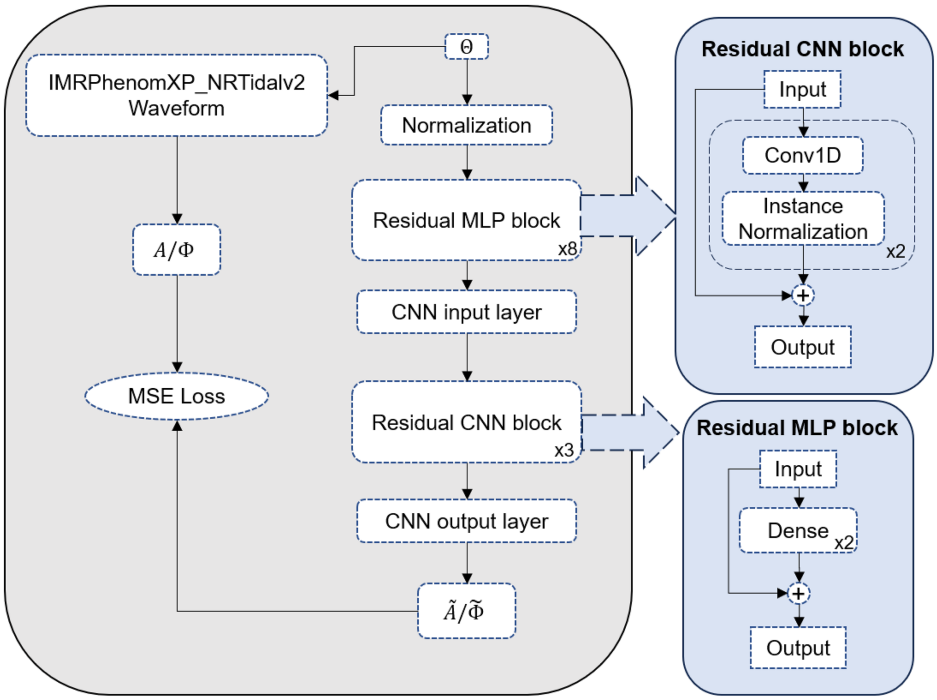

FIG. 13. Architectures of the residual-stacked multilayer perceptron (MLP) and convolutional neural network (CNN) model.

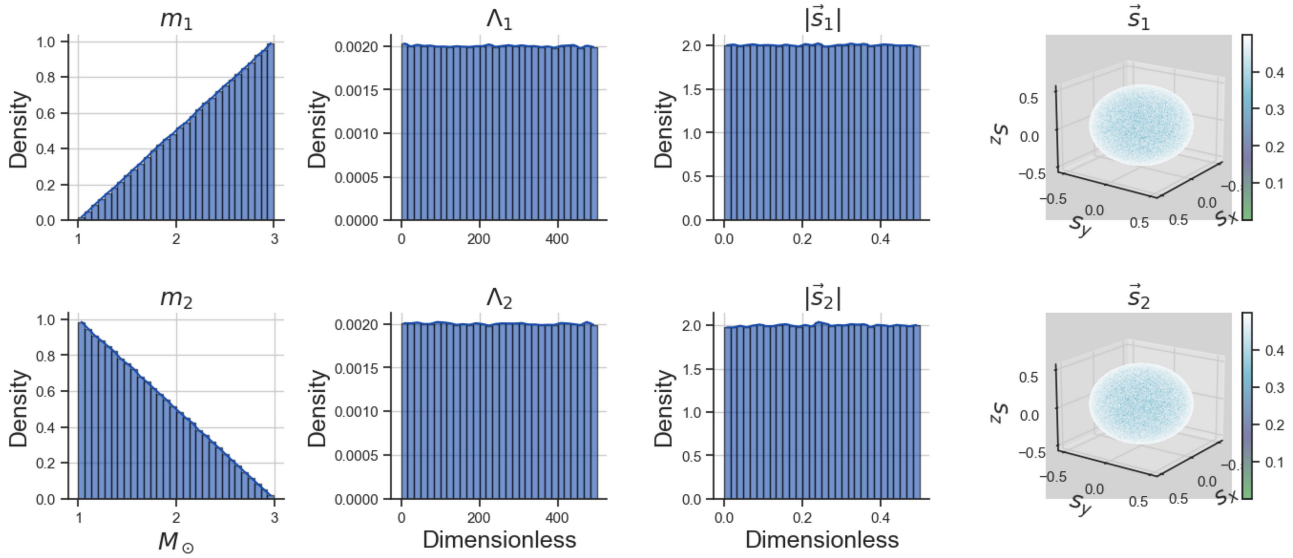

FIG. 14. Distribution of the 1×10^6 training samples in the parameter space Θ, including component masses, tidal deformabilities, and spin components. All parameters are sampled independently using uniform or isotropic priors.

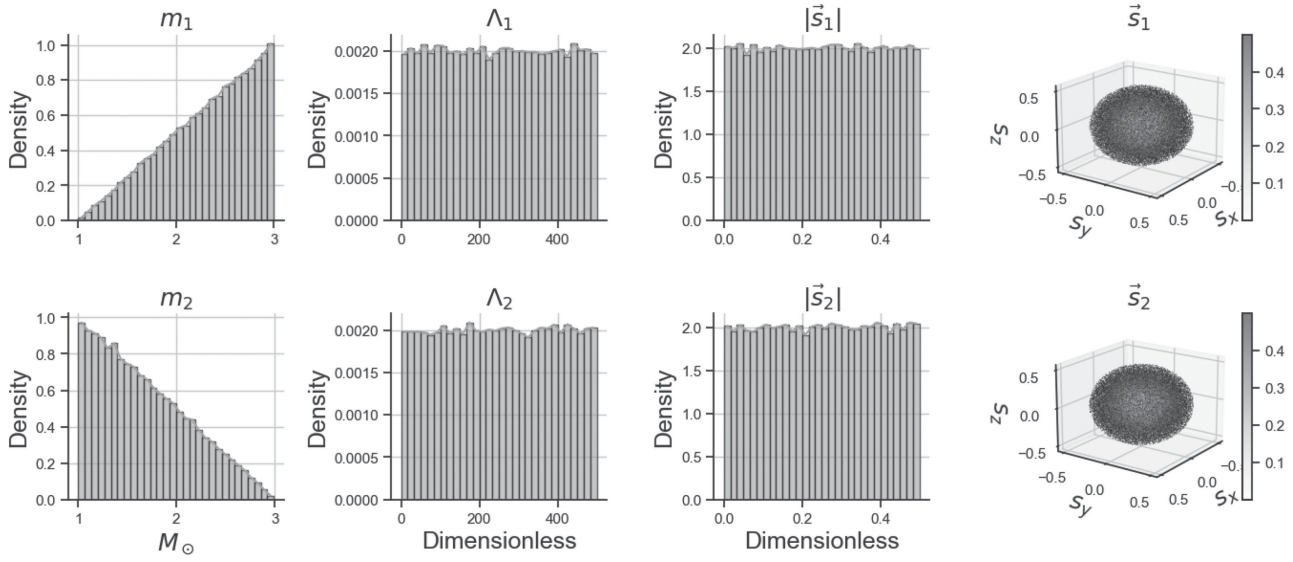

FIG. 15. Distribution of the 1×10^5 testing samples. The parameter priors are identical to those used in training, ensuring a consistent and representative evaluation set.

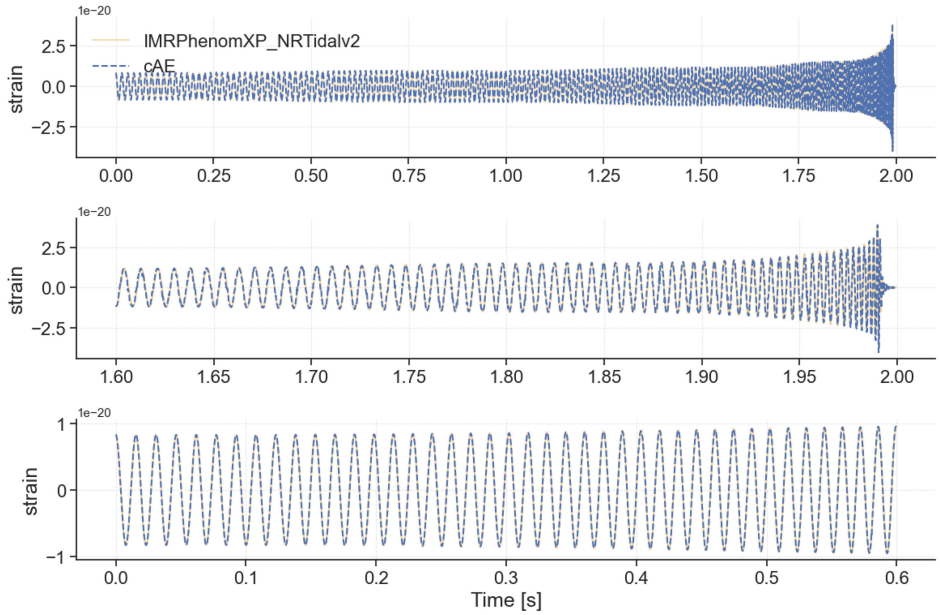

FIG. 16. Comparison of gravitational waveforms generated by the cAE (blue) and IMRPhenomXP_NRTidalv2 (yellow). The top subplot shows the full waveform over 0–2 s, with mismatch = 8.1×10^{-4} and 200 cycles. The middle subplot enlarges into $t = 1.5998$–1.9995 s. The bottom subplot enlarges into $t = 0$–0.5996 s. The test waveform parameters are $m_1 = 2.84M_\odot$, $m_2 = 2.81M_\odot$, $\Lambda_1 = 188.48$, $\Lambda_2 = 233.67$, $\text{spin}_{1x} = 0.13$, $\text{spin}_{1y} = -0.22$, $\text{spin}_{1z} = -0.01$, $\text{spin}_{2x} = -0.09$, $\text{spin}_{2y} = -0.10$, and $\text{spin}_{2z} = -0.21$.

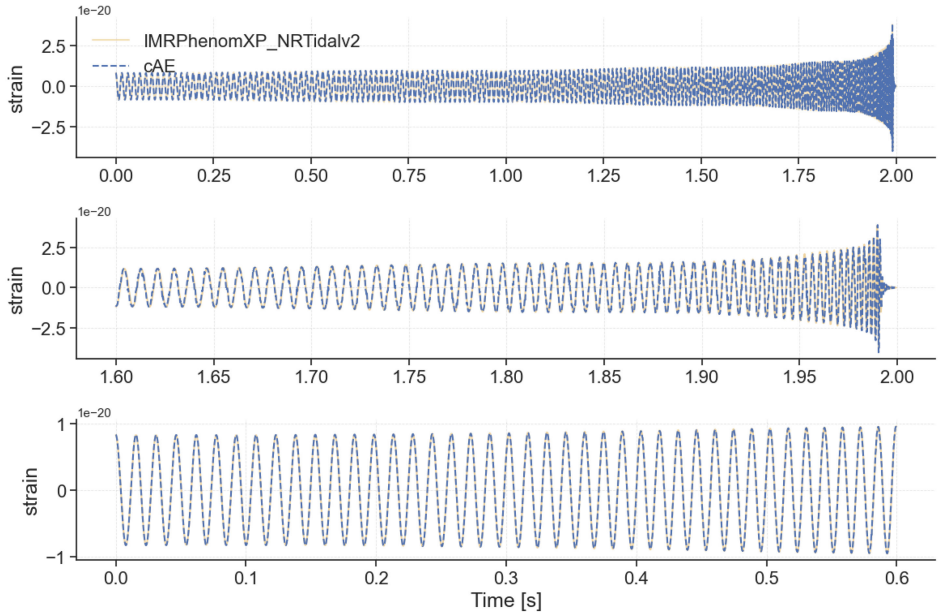

FIG. 17. The top subplot shows the full waveform over 0–2 s, with mismatch = 1.2×10^{-3} and 250 cycles. The middle subplot enlarges into $t = 1.5998$–1.9995 s. The bottom subplot enlarges into $t = 0$–0.5996 s. The test waveform parameters are $m_1 = 2.66M_\odot$, $m_2 = 1.81M_\odot$, $\Lambda_1 = 176.83$, $\Lambda_2 = 258.68$, $\text{spin}_{1x} = 0.19$, $\text{spin}_{1y} = 0.34$, $\text{spin}_{1z} = 0.08$, $\text{spin}_{2x} = -0.02$, $\text{spin}_{2y} = 0.21$, and $\text{spin}_{2z} = -0.18$.

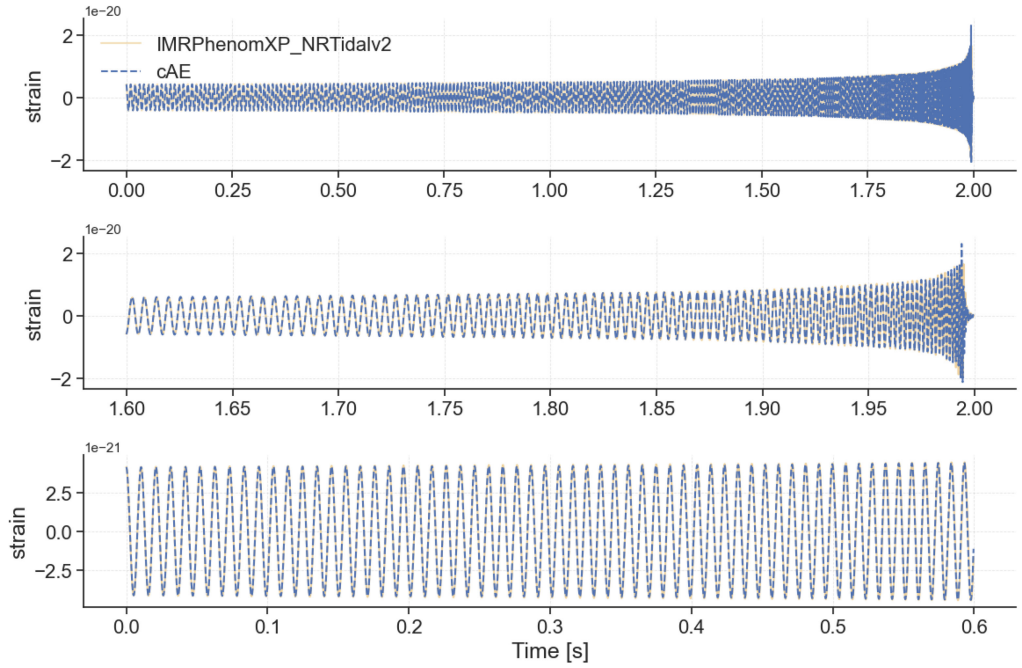

FIG. 18. The top subplot shows the full waveform over 0–2 s, with mismatch = 3.4×10^{-3} and 300 cycles. The middle subplot enlarges into $t = 1.5998$–1.9995 s. The bottom subplot enlarges into $t = 0$–0.5996 s. The test waveform parameters are $m_1 = 1.69M_\odot$, $m_2 = 1.66M_\odot$, $\Lambda_1 = 179.26$, $\Lambda_2 = 375.52$, $\text{spin}_{1x} = -0.08$, $\text{spin}_{1y} = 0.20$, $\text{spin}_{1z} = -0.36$, $\text{spin}_{2x} = 0.02$, $\text{spin}_{2y} = 0.36$, and $\text{spin}_{2z} = -0.04$.

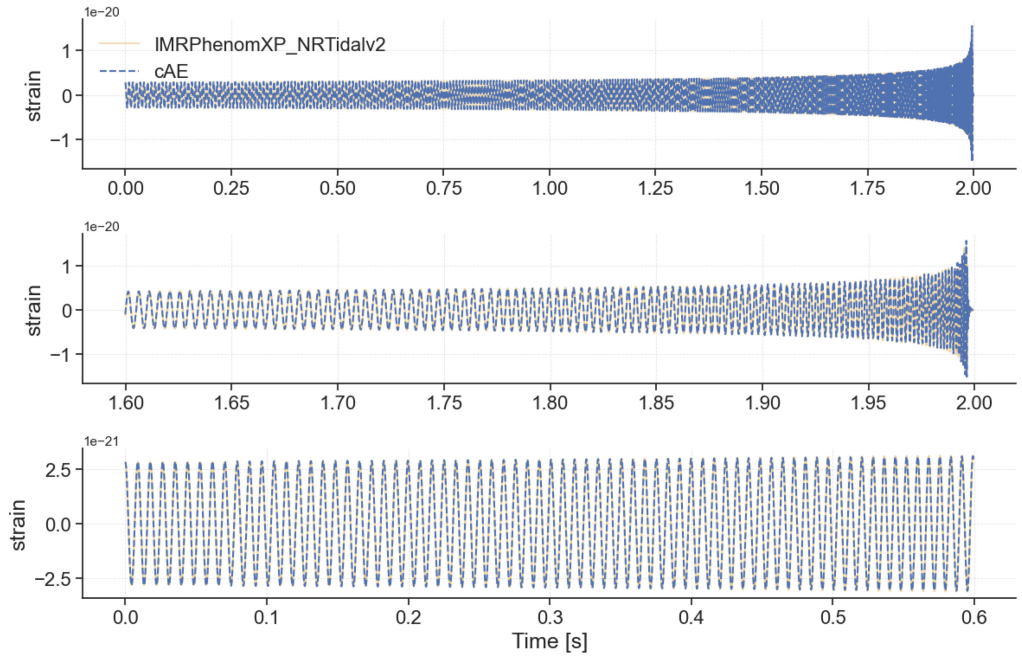

FIG. 19. The top subplot shows the full waveform over 0–2 s, with mismatch = 5.7×10^{-3} and 350 cycles. The middle subplot enlarges into $t = 1.5998$–1.9995 s. The bottom subplot enlarges into $t = 0$–0.5996 s. The test waveform parameters are $m_1 = 1.31M_\odot$, $m_2 = 1.24M_\odot$, $\Lambda_1 = 135.05$, $\Lambda_2 = 417.58$, $\text{spin}_{1x} = 0.24$, $\text{spin}_{1y} = 0.03$, $\text{spin}_{1z} = 0.06$, $\text{spin}_{2x} = -0.03$, $\text{spin}_{2y} = -0.30$, and $\text{spin}_{2z} = 0.32$.

[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.*, Observation of gravitational waves from a binary black hole merger, Phys. Rev. Lett. **116**, 061102 (2016).

[2] B. P. Abbott, R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya *et al.*, Gw170817: Observation of gravitational waves from a binary neutron star inspiral, Phys. Rev. Lett. **119**, 161101 (2017).

[3] B. P. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt *et al.*, Gw190425: Observation of a compact binary coalescence with total mass 3.4 $m_{\odot}$, Astrophys. J. **892**, L3 (2020).

[4] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. Adhikari, V. Adya, C. Affeldt *et al.*, Observation of gravitational waves from two neutron star–black hole coalescences, Astrophys. J. Lett. **915**, L5 (2021).

[5] D. Radice, A. Perego, K. Hotokezaka, S. A. Fromm, S. Bernuzzi, and L. F. Roberts, Binary neutron star mergers: Mass ejection, electromagnetic counterparts, and nucleosynthesis, Astrophys. J. **869**, 130 (2018).

[6] L. Sagunski, J. Zhang, M. C. Johnson, L. Lehner, M. Sakellariadou, S. L. Liebling, C. Palenzuela, and D. Neilsen, Neutron star mergers as a probe of modifications of general relativity with finite-range scalar forces, Phys. Rev. D **97**, 064016 (2018).

[7] E. E. Flanagan and T. Hinderer, Constraining neutron-star tidal love numbers with gravitational-wave detectors, Phys. Rev. D **77**, 021502 (2008).

[8] T. Hinderer, Tidal Love numbers of neutron stars, Astrophys. J. **677**, 1216 (2008).

[9] J. Haislip, V. Kouprianov, D. Reichart, L. Tartaglia, D. Sand, S. Valenti, S. Yang, I. Arcavi, G. Hosseinzadeh, D. Howell, C. McCully, D. Poznanski, and S. Vasylyev, A gravitational-wave standard siren measurement of the Hubble constant, Nature (London) **551**, 85 (2017).

[10] K. Hotokezaka, P. Beniamini, and T. Piran, Neutron star mergers as sites of r-process nucleosynthesis and short gamma-ray bursts, Int. J. Mod. Phys. D **27**, 1842005 (2018).

[11] S. Bernuzzi, A. Nagar, M. Thierfelder, and B. Brügmann, Tidal effects in binary neutron star coalescence, Phys. Rev. D **86**, 044030 (2012).

[12] M. Hannam, P. Schmidt, A. Bohé, L. Haegel, S. Husa, F. Ohme, G. Pratten, and M. Pürrer, Simple model of complete precessing black-hole-binary gravitational waveforms, Phys. Rev. Lett. **113**, 151101 (2014).

[13] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond *et al.*, Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors, Phys. Rev. D **95**, 044028 (2017).

[14] J. Centrella, J. G. Baker, B. J. Kelly, and J. R. Van Meter, Black-hole binaries, gravitational waves, and numerical relativity, Rev. Mod. Phys. **82**, 3069 (2010).

[15] T. W. Baumgarte and S. L. Shapiro, Numerical relativity and compact binaries, Phys. Rep. **376**, 41 (2003).

[16] F. Löffler, J. Faber, E. Bentivegna, T. Bode, P. Diener, R. Haas, I. Hinder, B. C. Mundim, C. D. Ott, E. Schnetter *et al.*, The Einstein toolkit: A community computational infrastructure for relativistic astrophysics, Classical Quantum Gravity **29**, 115001 (2012).

[17] L. Blanchet, Gravitational radiation from post-Newtonian sources and inspiralling compact binaries, Living Rev. Relativity **17**, 2 (2014).

[18] A. Einstein, L. Infeld, and B. Hoffmann, The gravitational equations and the problem of motion, Ann. Math. **39**, 65 (1938).

[19] A. Buonanno and T. Damour, Effective one-body approach to general relativistic two-body dynamics, Phys. Rev. D **59**, 084006 (1999).

[20] T. Damour, A. Nagar, E. N. Dorband, D. Pollney, and L. Rezzolla, Faithful effective-one-body waveforms of equal-mass coalescing black-hole binaries, Phys. Rev. D **77**, 084017 (2008).

[21] A. Buonanno and T. Damour, Transition from inspiral to plunge in binary black hole coalescences, Phys. Rev. D **62**, 064015 (2000).

[22] S. Khan, K. Chatziioannou, M. Hannam, and F. Ohme, Phenomenological model for the gravitational-wave signal from precessing binary black holes with two-spin effects, Phys. Rev. D **100**, 024059 (2019).

[23] P. Ajith, S. Babak, Y. Chen, M. Hewitson, B. Krishnan, J. Whelan, B. Bruegmann, P. Diener, J. Gonzalez, M. Hannam *et al.*, A phenomenological template family for black-hole coalescence waveforms, Classical Quantum Gravity **24**, S689 (2007).

[24] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. J. Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era, Phys. Rev. D **93**, 044007 (2016).

[25] Z. Doctor, B. Farr, D. Holz, and M. Purrer, Statistical gravitational waveform models: What to simulate next?, Phys. Rev. D **96** (2017).

[26] S. E. Field, C. R. Galley, J. S. Hesthaven, J. Kaye, and M. Tiglio, Fast prediction and evaluation of gravitational waveforms using surrogate models, Phys. Rev. X **4**, 031006 (2014).

[27] A. Nagar and P. Rettegno, Efficient effective one body time-domain gravitational waveforms, Phys. Rev. D **99**, 021501(R) (2018).

[28] T. Grimbergen, S. Schmidt, C. Kalaghatgi, and C. van den Broeck, Generating higher order modes from binary black hole mergers with machine learning, Phys. Rev. D **109**, 104065 (2024).

[29] O. G. Freitas, A. Theodoropoulos, N. Villanueva, T. Fernandes, S. Nunes, J. A. Font, A. Onofre, A. Torres-Forné, and José D. Martin-Guerrero, A deep learning powered numerical relativity surrogate for binary black hole waveforms, Phys. Rev. D **112**, 043026 (2025).

[30] C.-H. Liao and F.-L. Lin, Deep generative models of gravitational waveforms via conditional autoencoder, Phys. Rev. D **103**, 124051 (2021).

[31] R. Shi, Y. Zhou, T. Zhao, Z. Wang, Z. Ren, and Z. Cao, Rapid eccentric spin-aligned binary black hole waveform generation based on deep learning, Phys. Rev. D **111**, 044016 (2025).

[32] S. He, H. Wang, H. Li, and J. Zhao, Principle of machine learning and its potential application in climate prediction, J. Autonom. Intell. **4**, 13 (2021).
[33] D. George and E. A. Huerta, Deep learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data, Phys. Lett. B **778**, 64 (2018).
[34] S. Schmidt, M. Breschi, R. Gamba, G. Pagano, P. Rettegno, G. Riemenschneider, S. Bernuzzi, A. Nagar, and W. Del Pozzo, Machine learning gravitational waves from binary black hole mergers, Phys. Rev. D **103**, 043020 (2021).
[35] M. Dax, S. R. Green, J. Gair, N. Gupte, M. Pürrer, V. Raymond, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, Real-time inference for binary neutron star mergers using machine learning, Nature (London) **639**, 49 (2025).
[36] T. Whittaker, W. E. East, S. R. Green, L. Lehner, and H. Yang, Using machine learning to parametrize postmerger signals from binary neutron stars, Phys. Rev. D **105**, 124021 (2022).
[37] A. J. Chua, M. L. Katz, N. Warburton, and S. A. Hughes, Rapid generation of fully relativistic extreme-mass-ratio-inspiral waveform templates for LISA data analysis, Phys. Rev. Lett. **126**, 051102 (2021).
[38] T. Dietrich, A. Samajdar, S. Khan, N. K. Johnson-McDaniel, R. Dudi, and W. Tichy, Improving the NRTidal model for binary neutron star systems, Phys. Rev. D **100**, 044003 (2019).
[39] M. Colleoni, F. A. Ramis Vidal, N. K. Johnson-McDaniel, T. Dietrich, M. Haney, and G. Pratten, New gravitational waveform model for precessing binary neutron stars with double-spin effects, Phys. Rev. D **111**, 064025 (2025).
[40] A. Nitz *et al.*, gwastro/pycbc: Release 2.2.0 of PYCBC (2023).
[41] S. Pal and R. K. Nayak, Tidal reconstruction of neutron star mergers from their late inspiral, Astrophys. J. **980**, 76 (2025).
[42] L. Rahmad Ramadhan and Y. Anne Mudya, A comparative study of z-score and min-max normalization for rainfall classification in pekanbaru, J. Data Sci. **2024**, 1 (2024).
[43] N. Singh and P. Singh, Exploring the effect of normalization on medical data classification, in *Proceedings of the 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)* (IEEE, New York, 2021), pp. 1–5.
[44] G. E. Hinton and R. Zemel, Autoencoders, minimum description length and helmholtz free energy, Adv. Neural Inf. Process. Syst. **6**, 3 (1993).
[45] A. Maćkiewicz and W. Ratajczak, Principal components analysis (PCA), Comput. Geosci. **19**, 303 (1993).
[46] D. Cacciarelli and M. Kulahci, Hidden dimensions of the data: PCA vs autoencoders, Qual. Eng. **35**, 741 (2023).
[47] E. J. Bloomer, A principal component analysis of gravitational-wave signals from extreme-mass-ratio sources, Ph.D. thesis, University of Glasgow (2010).
[48] M. P. Libório, O. Da Silva Martinuci, A. M. C. Machado, T. M. Machado-Coelho, S. Laudares, and P. Bernardes, Principal component analysis applied to multidimensional social indicators longitudinal studies: Limitations and possibilities, GeoJournal **87**, 1453 (2022).
[49] S. Ladjal, A. Newson, and C.-H. Pham, A PCA-like autoencoder, arXiv:1904.01277.
[50] P. Nousi, S.-C. Fragkouli, N. Passalis, P. Iosif, T. Apostolatos, G. Pappas, N. Stergioulas, and A. Tefas, Autoencoder-driven spiral representation learning for gravitational wave surrogate modelling, Neurocomputing;Variable Star Bulletin **491**, 67 (2022).
[51] D. P. Kingma, M. Welling *et al.*, An introduction to variational autoencoders, Found. Trends Mach. Learn. **12**, 307 (2019).
[52] A. Asperti and M. Trentin, Balancing reconstruction error and Kullback-Leibler divergence in variational autoencoders, IEEE Access **8**, 199440 (2020).
[53] V. Prokhorov, E. Shareghi, Y. Li, M. T. Pilehvar, and N. Collier, On the importance of the Kullback-Leibler divergence term in variational autoencoders for text generation, arXiv:1909.13668.
[54] C. Zhang, R. Barbano, and B. Jin, Conditional variational autoencoder for learned image reconstruction, Computation **9**, 114 (2021).
[55] B. J. Owen, Search templates for gravitational waves from inspiraling binaries: Choice of template spacing, Phys. Rev. D **53**, 6749 (1996).
[56] B. J. Owen and B. S. Sathyaprakash, Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement, Phys. Rev. D **60**, 022002 (1999).
[57] LIGO Scientific Collaboration, Noise curves used for simulations in the update of the observing scenarios paper, LIGO Document Control Center (DCC), T2000012-v2 (2022), available online: https://dcc.ligo.org/LIGO-T2000012/public.
[58] R. Shi, Y. Zhou, T. Zhao, Z. Wang, Z. Ren, and Z. Cao, Rapid eccentric spin-aligned binary black hole waveform generation based on deep learning, Phys. Rev. D **111**, 044016 (2025).
[59] L. Marple, Computing the discrete-time "analytic" signal via FFT, IEEE Trans. Signal Process. **47**, 2600 (1999).
[60] D. Foreman-Mackey, corner.py: Scatterplot matrices in python, J. Open Source Softwaare **1**, 24 (2016).
[61] P. Chaturvedi, A. Khan, M. Tian, E. Huerta, and H. Zheng, Inference-optimized ai and high performance computing for gravitational wave detection at scale, Front. Artifi. Intell. **5**, 828672 (2022).
[62] Z. Lijun, L. Yu, B. Lu, L. Fei, and W. Yawei, Using tensorrt for deep learning and inference applications, J. Opt. A **41**, 337 (2020).
[63] N. Alizadeh and F. Castor, Green AI: A preliminary empirical study on energy consumption in dl models across different runtime infrastructures, in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI* (2024), pp. 134–139.
[64] S. Mo and Y. Tian, Scaling diffusion mamba with bidirectional SSMs for efficient image and video generation, arXiv:2405.15881.
[65] Z. Fei, M. Fan, C. Yu, D. Li, Y. Zhang, and J. Huang, Dimba: Transformer-mamba diffusion models, arXiv:2406.01159.
[66] https://github.com/thisshy/BNS_Waveform_Generator_Based_on_CAE.