Conditional Generation of Periodic Signals with Fourier-Based Decoder

Jiyoung Lee
KAIST AI
Daejeon, South Korea
jiyounglee0523@kaist.ac.kr

Wonjae Kim
NAVER AI LAB
Seongnam-si, South Korea
wonjae.kim@navercorp.com

Daehoon Gwak
KAIST AI
Daejeon, South Korea
daehoon.gwak@kaist.ac.kr

Edward Choi
KAIST AI
Daejeon, South Korea
edwardchoi@kaist.ac.kr

Abstract

Periodic signals play an important role in daily lives. Although conventional sequential models have shown remarkable success in various fields, they still come short in modeling periodicity; they either collapse, diverge or ignore details. In this paper, we introduce a novel framework inspired by Fourier series to generate periodic signals. We first decompose the given signals into multiple sines and cosines and then conditionally generate periodic signals with the output components. We have shown our model efficacy on three tasks: reconstruction, imputation and conditional generation. Our model outperforms baselines in all tasks and shows more stable and refined results.

1 Introduction

Periodic signals exist in daily lives. In biomedical domain, electrocardiogram (ECG) [18] and body temperature [16, 25] are critical periodic signals in examining patients health status. ECG, in particular, is an important measure to diagnose patient heart diseases [14] such as Myocardial Infarction, AV Block, and Ventricular Tachycardia. Despite its significance, most publicly open-sourced ECG datasets [20, 2] are small in size and they all tend to have a severe data imbalance problem [27]–normal or common disease ECG records make up the majority while rare diseases barely exist. Therefore, generating ECG records conditioned on diagnosis can be beneficial in solving data imbalance issue and further developing ECG deep learning models.

Sequential models, such as recurrent neural networks (RNNs) [15, 6], transformer decoder [31], neural ordinary differential equations (NODEs) [4], and neural processes (NPs) [12, 13, 17] have shown excellent results in various fields [7, 3, 22]. However, as shown in section 4, there is room for improvement when it comes to generating periodic signals. These models either collapse, diverge or ignore subtle periodic details. In theory, discrete Fourier series (DFS) or discrete Fourier transform (DFT) can handle sampled periodic signals. In practice, however, DFS and DFT perform poorly when signals are obtained in irregular timesteps or contain noise [28, 33], and are difficult to serve as (conditional) generative models.

In this paper, we introduce a novel architecture that directly utilizes Fourier series for generating periodic signals. By the definition of Fourier series, periodic signals are composed of sinusoids. Built upon conditional variational autoencoder [29], we first obtain latent representation vector of the signal with the encoder. The decoder then outputs Fourier coefficients, which represent how much each sinusoid contributes to compose the original signal, using the sampled latent vector. We further

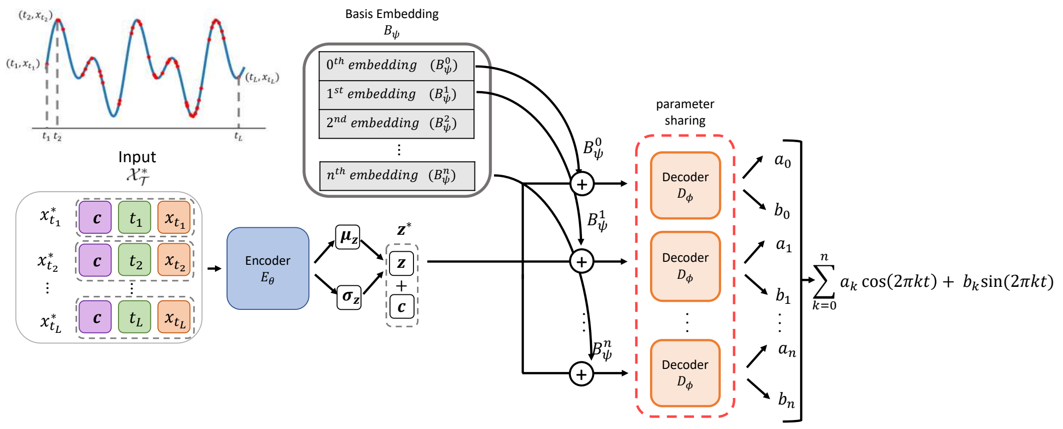
Basis Embedding B_ψ
0^{th} embedding (B_ψ^0)
1^{st} embedding (B_ψ^1)
2^{nd} embedding (B_ψ^2)
n^{th} embedding (B_ψ^n)
Input $\mathcal{X}_\mathcal{T}^*$
Encoder E_θ
μ_z
σ_z
z^*
parameter sharing
Decoder D_ϕ
a_0 b_0 a_1 b_1 a_n b_n
$\sum_{k=0}^{n} a_k \cos(2\pi kt) + b_k \sin(2\pi kt)$

Figure 1: Model architecture. Input $x_{t_i}^*$ is a concatenation of input signal x_{t_i}, time t_i, and label $\mathbf{c}$. We sample $\mathbf{z}$ from $\mathcal{N}(\mu_\mathbf{z}, \sigma_\mathbf{z})$, and concatenate with label $\mathbf{c}$, producing $\mathbf{z}^*$. Then $\mathbf{z}^*$ is added with the basis embeddings B_ψ^k and passed through the decoder D_ϕ, which produces a_k and b_k, namely the *k-th* approximate Fourier coefficients. With a set of Fourier coefficients $\mathcal{A} = \{a_0, \ldots, a_n\}$ and $\mathcal{B} = \{b_0, \ldots, b_n\}$, we build an approximated Fourier series.

generate the label conditioned periodic signals by adding the class information to the sampled latent vector. We evaluate our model on three tasks: reconstruction, imputation, and conditional generation to demonstrate its usefulness. Our model outperforms baseline models in all tasks, suggesting a new direction for modeling periodic data.

2 Related Work

Modeling Periodic Signals Previous research that tried to model periodic signals replaces non-linear activation functions by sinusoids such as $sin(x)$, $cos(x)$ or the linear combination of the two [28, 34, 21, 35]. However, due to the activations' *non-monotonicity*, the periodic activation functions induce numerous local minima, causing troubles in model optimization [21].

Fourier Neural Networks (FNNs) FNNs are neural networks that resemble Fourier series. Gallant and White [11] suggest *cosine squasher* as an activation function. Silvescu [28] use $cos(x)$ as an activation function to mimic Fourier series. Liu [19] proposes a combination of $cos(x)$ and $sin(x)$ as activation function, and showed comparable results on various datasets empirically. FNN had a superior performance in practical tasks such as aircraft engine fault diagnostics [30], and control of a class of uncertain nonlinear systems [36–38].

Conditional Time-Series Generation Previous conditional time-series generative work employ convolutional neural networks (CNNs) [1, 5] and RNNs [10]. They often make use of generative adversarial networks (GANs) to achieve a realistic-looking data [23]. Esteban et al. [10] achieve conditional generation by setting RNNs in both generator and discriminator and concatenating a class label for every time steps.

3 Proposed Methods

3.1 Problem Definition

In this section, we will clarify notations to be used throughout the paper and define our task. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_L\}$ be a set of L ordered timesteps with $t_i \in \mathbb{R}$, and $\mathcal{X}_\mathcal{T} = \{x_{t_1}, x_{t_2}, \ldots, x_{t_L}\}$ be a corresponding input sequence, where $x_{t_i} \in \mathbb{R}$. We note that the intervals among $\mathcal{T}$ can be irregular. Additionally, the input sequence $\mathcal{X}_\mathcal{T}$ has its own one-hot vector label $\mathbf{c} \in \mathbb{R}^{n_c}$, where n_c is the number of unique labels. Given a label $\mathbf{c}$, the goal of conditional generation is to generate $\mathcal{X}_\mathcal{S}$ with another ordered timesteps $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$ that maximizes $P(\mathcal{X}_\mathcal{S} \mid \mathbf{c})$.

3.2 Background: Fourier Series

Fourier series approximates any periodic function $f(t)$ as an infinite sum of sines and cosines with increasing frequency as written in eq. (1). Here, P is the period of function $f(t)$.

$$f(t) = \sum_{k=0}^{\infty} A_k cos\left(\frac{2\pi kt}{P}\right) + B_k sin\left(\frac{2\pi kt}{P}\right) \tag{1}$$

$A_k \in \mathbb{R}, B_k \in \mathbb{R}$ are *Fourier coefficients*. They represent how much each sine and cosine is contributing to formulate the given function. Hereafter, we will name the set of sines and cosines sharing the same frequency as *basis* and we will assume all signals have a period of 1. Originally, Fourier coefficients are acquired by integrating the original function $f(t)$ with corresponding sines or cosines. However, this calculation is infeasible when the data points are discrete and irregular.

For the most of the real signals, finite numbers of basis are sufficient to model a function $f(t)$. As such, we will predefine the number of basis, denoted as n. In our model, given an input sequence $\mathcal{X}_\mathcal{T}$, we approximate the true value of A_i and B_i. We denote the approximations as $\mathcal{A} = \{a_0, \dots, a_n\}$ and $\mathcal{B} = \{b_1, \dots b_n\}$: a_i and b_i approximate A_i and B_i, respectively.

3.3 Model Architecture

Our model is built upon conditional variational autoencoder [29]. The overall architecture is illustrated in fig. 1. We construct the input sequence $\mathcal{X}^*_\mathcal{T} = [\mathbf{x}^*_{t_1}, \mathbf{x}^*_{t_2}, \dots, \mathbf{x}^*_{t_L}]$ where $\mathbf{x}^*_{t_i}$ is a concatenation of $x_{t_i} \in \mathcal{X}_\mathcal{T}$, $t_i \in \mathcal{T}$, and label $\mathbf{c}$ as follows:

$$\mathbf{x}^*_{t_i} = [x_{t_i}; t_i; \mathbf{c}] \tag{2}$$

Encoder E_θ produces $\boldsymbol{\mu}_z \in \mathbb{R}^z$ and $\boldsymbol{\sigma}_z \in \mathbb{R}^z$ from the input sequence $\mathcal{X}^*_\mathcal{T}$ and we sample the latent variable $\mathbf{z} \in \mathbb{R}^z$ from $q(\mathbf{z}|\mathcal{X}^*_\mathcal{T}) := \mathcal{N}(\boldsymbol{\mu}_z, diag(\boldsymbol{\sigma}_z))$. E_θ can be any type of model (*e.g.,* 1-D CNNs, RNNs, transformer encoder) capable of producing an informative representation of the input sequence.

Decoder $\mathbf{z} \in \mathbb{R}^z$ is sampled from $q(\mathbf{z}|\mathcal{X}^*_\mathcal{T})$ and concatenated with $\mathbf{c} \in \mathbb{R}^{n_c}$, producing $\mathbf{z}^* \in \mathbb{R}^{z+n_c}$ as $\mathbf{z}^* = [\mathbf{z}; \mathbf{c}]$. We have a basis embedding lookup table denoted as $B_\psi \in \mathbb{R}^{n\times(z+n_c)}$, where n is the predefined number of bases. The *k-th* row of B_ψ, B^k_ψ, represents the embedding vector for the *k-th* frequency basis. In order to obtain the *k-th* Fourier coefficient, a_k and b_k, we add B^k_ψ to the latent variable $\mathbf{z}^*$. Note that this addition is done in parallel, so we can compute it efficiently even when n is large. The decoder D_ϕ produces a_k and b_k as follows:

$$a_k, b_k = D_\phi(\mathbf{z}^* + B^k_\psi) \tag{3}$$

With $\mathcal{A} = \{a_0, \dots, a_n\}$ and $\mathcal{B} = \{b_1, \dots b_n\}$, we build an approximated Fourier series in eq. (4).

$$\hat{f}(t) = \sum_{k=0}^{n} \big(a_k cos(2\pi kt) + b_k sin(2\pi kt)\big) \tag{4}$$

Finally, we construct a sequence of predictions as $\widehat{\mathcal{X}}_\mathcal{T} = [\hat{x}_{t_1}, \hat{x}_{t_2}, \dots, \hat{x}_{t_L}]$ where $\hat{x}_{t_i} = \hat{f}(t_i)$. We optimize our model by the summation of reconstruction loss between the true observations $\mathcal{X}_\mathcal{T}$ and the predicted observations $\widehat{\mathcal{X}}_\mathcal{T}$, and the Kullback–Leibler divergence loss (D_{KL}) between the posterior distribution $q(\mathbf{z}|\mathcal{X}^*_\mathcal{T})$ and the prior distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The overall loss is:

$$\mathcal{L} = \sum_{i=1}^{L} \|x_{t_i} - \hat{x}_{t_i}\|^2_2 + \beta D_{KL}(q(\mathbf{z}|\mathcal{X}^*_\mathcal{T})||p(\mathbf{z})), \tag{5}$$

where $\beta > 0$ is a hyperparameter. After training, to generate $\mathcal{X}_\mathcal{S}$ conditioned on label $\mathbf{c}$, we sample $\mathbf{z}$ from the prior distribution $p(\mathbf{z})$ and concatenate with $\mathbf{c}$. Then we can generate $\mathcal{X}_\mathcal{S}$ by decoding the concatenated vector. Currently, our model assumes all input signals have the period of 1. Extension of this model to dynamically deal with signals of varying periods is our primary future research interest.

4 Experiments

4.1 Experimental Setup

We conduct experiments with two periodic datasets: toy sinusoid dataset and Physionet2021 [26]. Toy dataset is a simple mixture of three sine and cosine functions: $\sum_{i=1}^{3} m_{2i-1} cos(2\pi d_{2i-1} t) +$

$m_{2i} sin(2\pi d_{2i})$. We put four class conditions for the toy dataset based on amplitudes $\mathcal{M} = \{m_1, \dots, m_6\}$ and frequencies $\mathcal{D} = \{d_1, \dots, d_6\}$, resulting in four amplitude-frequency class labels: 'Low Amp. & Low Freq.', 'Low Amp. & High Freq.', 'High Amp. & Low Freq.', and 'High Amp. & High Freq.'. 'Low Amp.' classes sample m from a uniform distribution $U(1, 4)$ whereas 'High Amp.' classes sample m from $U(6, 9)$. 'Low Freq.' classes sample d from $U(1, 4)$, whereas 'High Freq.' classes sample d from $U(8, 11)$. Each signal has a total of 500 timesteps.

Physionet2021 [26] contains 12-lead ECG recordings collected from six separate datasets. We cropped each record into one second consisting of 500 timesteps and extracted samples with three diagnoses, namely 'Right Bundle Branch Block' (RBBB), 'Left Bundle Branch Block' (LBBB), and 'Atrial Fibrillation' (AF) from the V6 lead. These diagnoses are selected because they can be examined within one second record [9]. After preprocessing, there were total 36,110 cropped ECG samples. We split the data into train, validation and test sets with the ratio of 8:1:1. More details on preprocessing are in appendix A.

We sampled 20% of the timesteps during training to insure the irregularity of time. [1] For all experiments, we employ 5-layer 1D CNN as an encoder E_θ and 6-layer MLP as a decoder D_ϕ. We evaluate our model on three tasks: (1) reconstruction (for the sampled 20%), (2) imputation for missing timesteps (for the non-sampled 80%) and (3) conditional generation. For the tasks (1) and (2), we use sampled time points and make the model perform both reconstruction and imputation in parallel by generating $\widehat{\mathcal{X}}_\mathcal{T}$ for all timesteps. For the task (3), we sample $\mathbf{z}$ from the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and pass through the decoder D_ϕ. With the same 5-layer 1D CNN encoder, we compare our model with four different baseline decoders: Gated Recurrent Units (GRU), Transformer decoder, Neural ODE (NODE) and Neural Processes (NP). Further model implementation details are explained in appendix B.

4.2 Experiment Results of Toy Dataset

We report reconstruction and imputation results in table 1. Our model shows the lowest MSE on both reconstruction and imputation compared to other baseline models. We illustrate the reconstruction results in fig. 5 in the appendix C. Based on fig. 5, we found both GRU and Transformer to perform very poorly, though GRU was able to capture partial periodicity. NODE and NP showed better performance on low frequency samples, but they were not able to reproduce subtle details for high frequency samples. In contrast, our model showed superior performance across all amplitude-frequency classes.

Table 1: Reconstruction and imputation MSE on Toy Dataset

	Reconstruction	Imputation
GRU	72.454	74.711
Transformer	183.203	172.338
NODE	15.905	19.896
NP	3.813	4.615
Fourier (Ours)	**0.649**	**0.686**

We conditionally generate 2,000 samples from the sampled $\mathbf{z}$ for each amplitude-frequency class, and for each decoder. As visualized in fig. 2, NODE produced flattened samples and NP generated non-periodic signals with a large amplitude regardless of the class conditions. GRU and Transformer were able to produce periodic signals, but all 2,000 samples were nearly identical with minimal sample diversity. In contrast, our model was able to generate diverse periodic signals.

In order to verify whether the generated samples correctly reflect the class conditions, we conduct Fourier series analysis, which decomposes a given signal into multiple sines and cosines as expressed in eq. (1). From the analysis, we can calculate which frequency is used to compose the signal and its corresponding coefficients (*i.e.* amplitude). We plot histograms on both amplitude and frequency for the toy dataset, and compare the baselines with our model. The results are shown in fig. 3 for 'Low Amplitude & High Frequency' and 'High Amplitude & High Frequency', the rest two class conditions are described in fig. 6 in the appendix C. In all baseline models, their amplitude and frequency histograms are similarly shaped across the two classes, which implies that those models fail to reflect the class conditions when generating samples. Our model overlaps with the dataset

[1] We also conduct experiments without sampling to compare model performance in two scenarios (sparse irregular timeseries VS dense regular timeseries). The results are reported in appendix D, where our model outperforms the baselines. However, our model is particularly more powerful when input sequences are irregular, indicating its usefulness in handling real signals where irregularities exist due to missing timesteps [32].

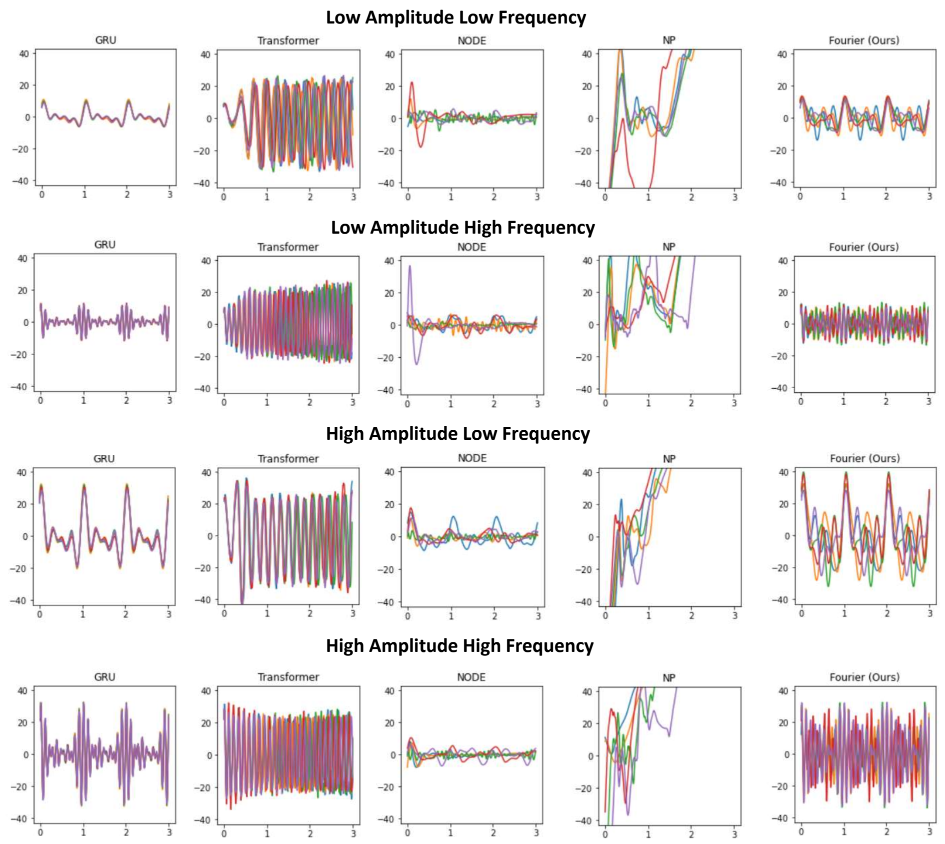

Figure 2: Conditionally generated samples in toy dataset. We draw five generated samples for each model plotted in different colors. Our model is able to generate diverse samples while assimilating class conditions whereas baseline models either collapse, diverge, or have a low sample diversity.

distribution most precisely compared to the other baselines, suggesting that the periodic signals generated by our model are properly conditioned on the class.

4.3 Experiment Results of Electrocardiogram

We report reconstruction and imputation results in table 2. Our model shows the lowest MSE in both reconstruction and imputation. The results are visualized by fig. 7 in appendix C. Based on fig. 7, GRU and Transformer cannot capture the peak of given ECG records. Although NODE and NP could grasp the peak point, they disregard details such as subtle waves in the isoelectric line, the straight line on the ECG. Our model can both capture the peak and the subtle fluctuations clearly.

Table 2: Reconstruction and imputation MSE on Physionet2021

	Reconstruction	Imputation
GRU	77.393	75.327
Transformer	68.823	35.692
NODEs	4.937	3.262
NP	2.630	1.840
Fourier (Ours)	**2.164**	**1.519**

We conditionally generate 3,000 samples for each diagnosis and decoder. As visualized in fig. 4, our model generates samples that are highly similar to the real dataset. According to Rawshani [24], RBBB diagnostic characteristics include a deep and broad depression after the peak while LBBB has a wide peak and a shallow depression after the peak. AF has a f-wave, a fibrillatory wave in the isoelectric line, and does not have a P-wave, a little uprising before the peak.

As shown in fig. 4, our model captures the necessary characteristics of each diagnosis, and in the figure, its significance is highlighted in red. GRU generates similar samples regardless of the given

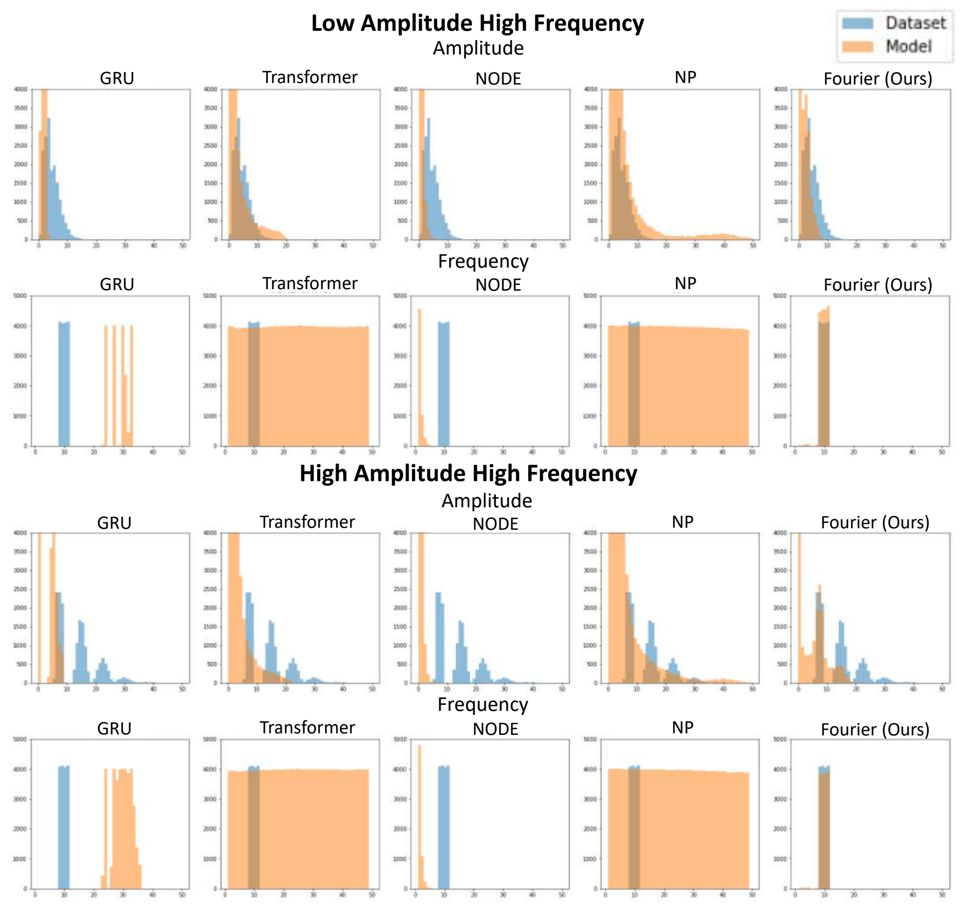

Figure 3: Fourier series analysis on conditionally generated samples. We plot histograms on amplitude and frequency for each model from Fourier analysis. Blue color represents the original dataset histogram and orange color represents each model. Note that the Transformer and NP cover more space than original dataset meaning that those models use more sinusoid to generate a single sample than the original dataset.

diagnosis, whereas Transformer draws flat lines after a certain time point. NODE and NP tend to generate smooth ECG signals ignoring all fluctuations in ECG. Also, they could not synthesize necessary characteristics of the given diagnosis.

We quantitatively evaluate the generated samples by using a pre-trained ECG classifier. The classifier is trained beforehand on the real dataset to classify the three diagnosis. We use total of five classifiers trained with a different parameter initialization seed. We run the classifier on our generated samples in order to confirm whether our samples are classified to their given diagnosis. We report our results in table 3.

Our model outperforms all other baselines for diagnosis-averaged overall scores, showing notable performance in AF. We speculate the reason for this improved performance is that our model is the only model that can synthesize f-wave, a main feature of AF, while other models fail to generate such fine oscillations.

5 Conclusion

In this work, we introduce a new Fourier-based model architecture that can generate periodic signals. Conventional sequential models collapse, diverge, or ignore subtle details even if they modeled the periodicity successfully. In contrast, our model outperforms all baseline models in all tasks across

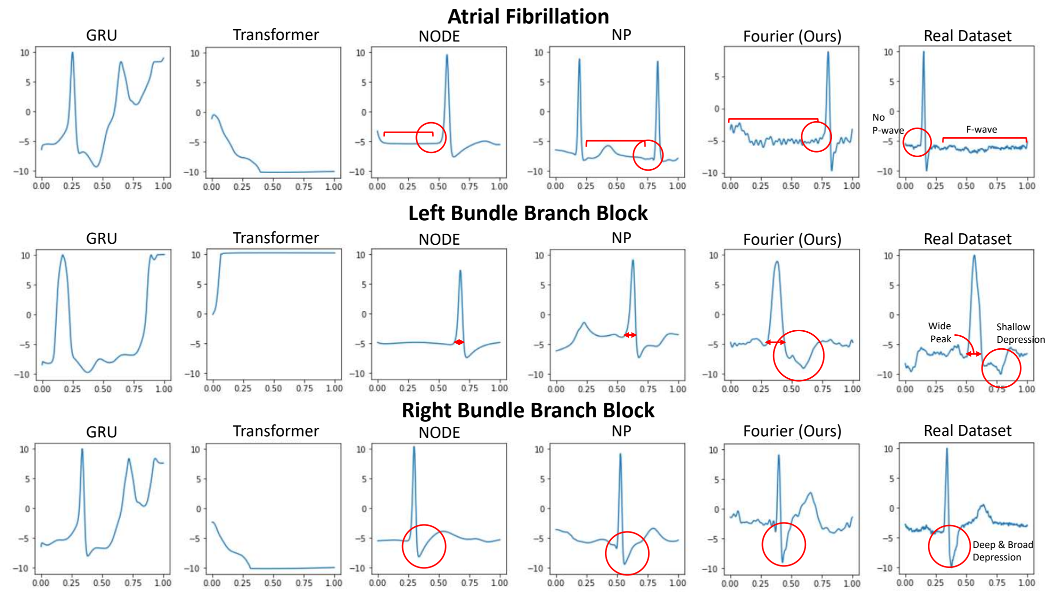

Figure 4: Conditionally generated ECG samples for each diagnosis. Real dataset samples are from Physionet2021. Main characteristics for each diagnosis and corresponding spots in generated samples are marked in red.

Table 3: Averaged and standard deviation of classifier performance on generated ECG samples

	Overall				AF			LBBB			RBBB		
	Acc	Recall	Precision	F1 Score	Recall	Precision	F1 Score	Recall	Precision	F1 Score	Recall	Precision	F1 Score
GRU	0.448 (0.017)	0.448 (0.017)	0.419 (0.023)	0.408 (0.022)	0.143 (0.035)	0.260 (0.050)	0.184 (0.040)	0.816 (0.015)	0.475 (0.016)	0.600 (0.009)	0.383 (0.056)	0.522 (0.020)	0.441 (0.044)
Transformer	0.338 (0.022)	0.338 (0.022)	0.283 (0.028)	0.239 (0.048)	0.263 (0.279)	0.481 (0.102)	0.265 (0.175)	0.002 (0.003)	0.050 (0.103)	0.004 (0.007)	0.748 (0.266)	0.317 (0.016)	0.434 (0.082)
NODE	0.488 (0.008)	0.488 (0.008)	0.496 (0.050)	0.465 (0.010)	0.240 (0.034)	0.443 (0.021)	0.309 (0.023)	0.459 (0.038)	0.587 (0.006)	0.514 (0.022)	0.765 (0.024)	0.457 (0.011)	0.572 (0.004)
NP	0.722 (0.004)	0.722 (0.004)	0.742 (0.003)	0.719 (0.004)	0.546 (0.015)	**0.748** (0.009)	0.631 (0.007)	**0.757** (0.013)	0.853 (0.008)	0.802 (0.005)	**0.863** (0.011)	0.625 (0.010)	**0.725** (0.004)
Fourier (Ours)	**0.730** (0.005)	**0.730** (0.005)	**0.746** (0.004)	**0.734** (0.004)	**0.710** (0.056)	0.675 (0.024)	**0.691** (0.014)	0.746 (0.019)	**0.906** (0.017)	**0.818** (0.005)	0.735 (0.040)	**0.657** (0.025)	0.693 (0.008)
Real Dataset	0.816 (0.005)	0.796 (0.004)	0.816 (0.005)	0.805 (0.004)	0.737 (0.016)	0.756 (0.014)	0.746 (0.008)	0.777 (0.002)	0.850 (0.007)	0.812 (0.004)	0.873 (0.011)	0.841 (0.009)	0.857 (0.006)

two datasets, showing stable and refined results. We plan to extend out work to more diverse signals with various periods, since we only examined the samples with a period of 1. We believe the periodic structure inherited by the Fourier series will enable new possibilities to model various periodic signals even beyond bio-signals in the future.

References

[1] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

[2] R Bousseljot, D Kreiseler, and A Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. 1995.

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[4] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

[5] Yepeng Cheng, Zuren Liu, and Yasuhiko Morimoto. Attention-based seriesnet: An attention-based hybrid neural network model for conditional time series forecasting. *Information*, 11(6):305, 2020.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[8] Ivaylo I Christov. Real time electrocardiogram qrs detection using combined adaptive threshold. *Biomedical engineering online*, 3(1):1–9, 2004.

[9] David Da Costa, William J Brady, and June Edhouse. Bradycardias and atrioventricular conduction block. *Bmj*, 324(7336):535–538, 2002.

[10] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[11] A Ronald Gallant and Halbert White. There exists a neural network that does not make avoidable mistakes. In *ICNN*, pages 657–664, 1988.

[12] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2018.

[13] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[14] Bo Hedén, Mattias Ohlsson, Holger Holst, Mattias Mjöman, Ralf Rittner, Olle Pahlm, Carsten Peterson, and Lars Edenbrandt. Detection of frequently overlooked electrocardiographic lead reversals using artificial neural networks. *The American journal of cardiology*, 78(5):600–604, 1996.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] Norio Ishida, Maki Kaneko, and Ravi Allada. Biological clocks. *Proceedings of the National Academy of Sciences*, 96(16):8819–8820, 1999.

[17] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

[18] Yakup Kutlu, Gokhan Altan, and Novruz Allahverdi. Arrhythmia classification using waveform ecg signals. In *Int. Conf. Advanced Technology & Sciences, Konya, Turkey*, 2016.

[19] Shuang Liu. Fourier neural network for machine learning. In *2013 International Conference on Machine Learning and Cybernetics*, volume 1, pages 285–290. IEEE, 2013.

[20] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[21] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Taming the waves: sine as activation function in deep neural networks. 2016.

[22] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[23] Giorgia Ramponi, Pavlos Protopapas, Marco Brambilla, and Ryan Janssen. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv preprint arXiv:1811.08295*, 2018.

[24] Araz Rawshani. Clinical ecg interpretation, Dec 2019. URL `https://ecgwaves.com/product/clinical-ecg-interpretation/`.

[25] Roberto Refinetti and Michael Menaker. The circadian rhythm of body temperature. *Physiology & behavior*, 51(3):613–637, 1992.

[26] Matthew A Reyna, Nadi Sadr, Erick A Perez Alday, Annie Gu, Amit J Shah, Chad Robichaux, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, Sardar Ansari, et al. Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021. *Computing in Cardiology*, 48:1–4, 2021.

[27] Abdelrahman M Shaker, Manal Tantawi, Howida A Shedeed, and Mohamed F Tolba. Generalization of convolutional neural networks for ecg classification using generative adversarial networks. *IEEE Access*, 8: 35592–35605, 2020.

[28] Adrian Silvescu. Fourier neural networks. In *Association for the Advancement of Artificial Intelligence*, 2000.

[29] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[30] HS Tan. Fourier neural networks and generalized single hidden layer networks in aircraft engine fault diagnostics. 2006.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[32] Fei Yang, Jiazhi Du, Jiying Lang, Weigang Lu, Lei Liu, Changlong Jin, and Qinma Kang. Missing value estimation methods research for arrhythmia classification using the modified kernel difference-weighted knn algorithms. *BioMed research international*, 2020, 2020.

[33] Haolong Zhang, Haoye Lu, and Amiya Nayak. Periodic time series data analysis by deep learning methodology. *IEEE Access*, 8:223078–223088, 2020.

[34] Abylay Zhumekenov, Malika Uteuliyeva, Olzhas Kabdolov, Rustem Takhanov, Zhenisbek Assylbekov, and Alejandro J Castro. Fourier neural networks: a comparative study. *arXiv preprint arXiv:1902.03011*, 2019.

[35] Liu Ziyin, Tilman Hartwig, and Masahito Ueda. Neural networks fail to learn periodic functions and how to fix it. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1583–1594. Curran Associates, Inc., 2020.

[36] Wei Zuo and Lilong Cai. Tracking control of nonlinear systems using fourier neural network. In *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pages 670–675. IEEE, 2005.

[37] Wei Zuo and Lilong Cai. Adaptive-fourier-neural-network-based control for a class of uncertain nonlinear systems. *IEEE transactions on neural networks*, 19(10):1689–1701, 2008.

[38] Wei Zuo, Yang Zhu, and Lilong Cai. Fourier-neural-network-based learning control for a class of nonlinear systems with flexible components. *IEEE transactions on neural networks*, 20(1):139–151, 2008.

A Dataset

A.1 Toy Dataset

The number of samples in each class is 20,000 in train set and 5,000 for validation and test sets. We have total 80,000 samples in train set and 20,000 for validation and test sets. Each sample lasts for three seconds and has total of 500 time steps. We added Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, diag(0.3))$ in all samples.

A.2 Physionet 2021

We filtered out those which sampling rate is not 500Hz and those that last less than 10 seconds, or longer than 20 seconds. We utilized data between 5 to 10 seconds because the majority of samples include extreme noise up to 5 second and stabilize afterward. With using ECG R-peak detectors[2] [8], we maintain samples that have one or two detected R peaks within one second. This process filters out extremely noisy samples, such as samples with all zero values or samples that do not show any dominant QRS peaks but rather a random noise. The total number of diagnosis label in final preprocessed dataset is shown in Table 4.

Table 4: Number of diagnosis labels in final dataset

	# of Samples
RBBB	19,425
LBBB	5,270
AF	11,415
Total	36,110

B Experiment Details

B.1 Model Architecture

Table 5: Encoder architecture

Layer	Layer Information
Layer 1	Conv(# of output channels=256, Kernel=3, Stride=1), MaxPool(2), SiLU
Layer 2	Conv(# of output channels=256, Kernel=3, Stride=1), MaxPool(2), SiLU
Layer 3	Conv(# of output channels=256, Kernel=3, Stride=1), MaxPool(2), SiLU
Layer 4	Conv(# of output channels=256, Kernel=3, Stride=1), MaxPool(2), SiLU
Layer 5	Conv(# of output channels=128, Kernel=3, Stride=1)

Table 6: Decoder architecture

	Latent Dimension	Hidden Dimension	# of Layers	# of Multi-Head Attention
Fourier (Ours)	128	256	6	-
NP	128	256	6	-
NODE	128	256	6	-
Transformer	128	256	2	4
GRU	128	256	2	-

B.2 Hyperparameters

We share a fixed hyperparameter set for Toy Datset and Physionet2021 as follows.

[2]https://github.com/berndporr/py-ecg-detectors

- dropout = 0.1
- lr = 1e-4
- batch size = 512
- $\beta = 1$ for Toy Dataset, $\beta = 30$ for Physionet2021

B.3 Experiment environments

We train our models on NVIDIA GeForce RTX 3090. Also, CUDA version is 11.1 and torch version is 1.8.1.

C Additional Experiment Results

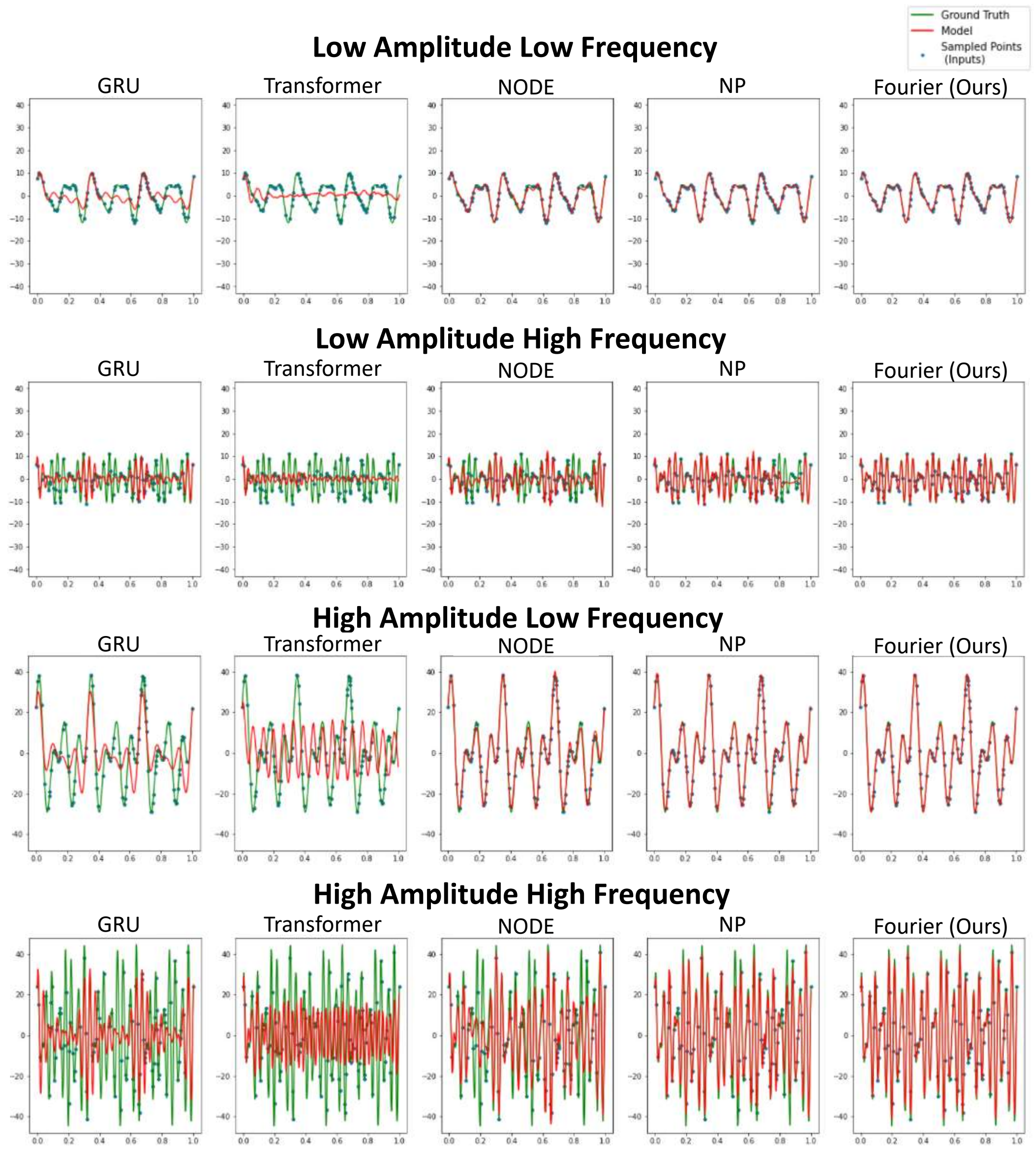

Figure 5: Illustrated examples of reconstruction and imputation on toy dataset. Green line indicates the ground truth and blue points are sampled inputs. Red line is the model's prediction.

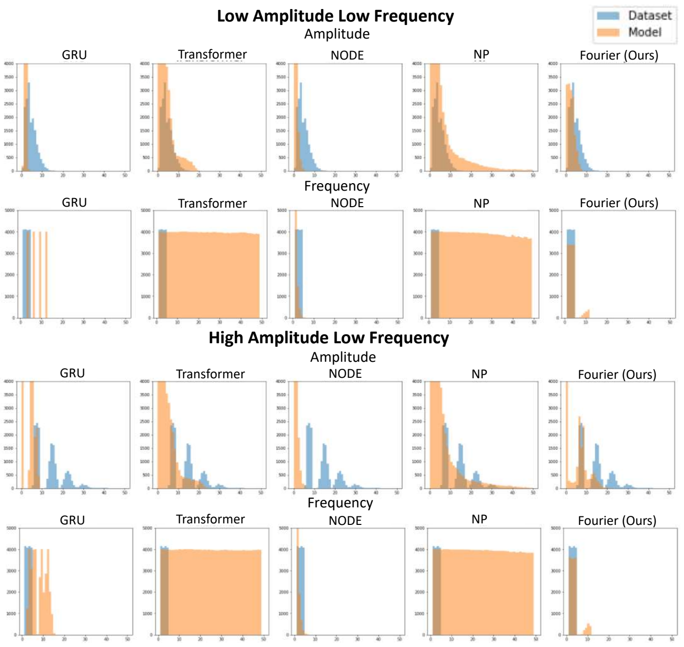

Figure 6: Fourier anaylsis in 'Low Amplitude & Low Frequency' and 'High Amplitude & Low Frequency'

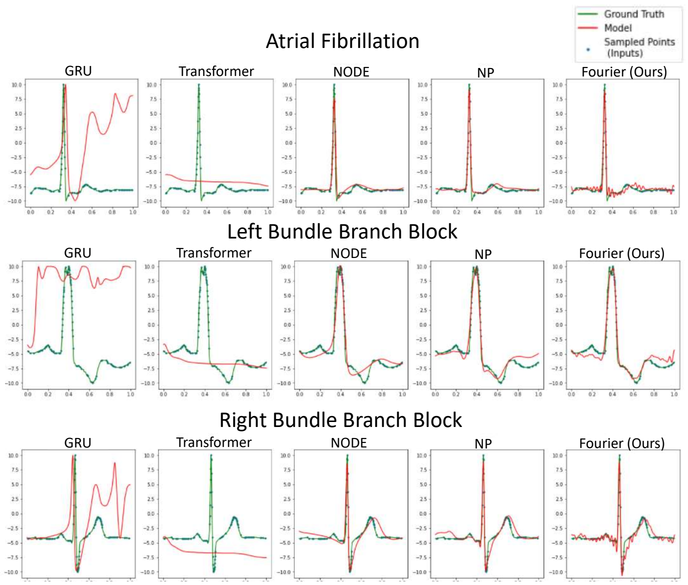

Figure 7: Illustrated examples of reconstruction and imputation on Physionet2021. Green line is the ground truth and blue points are the sampled inputs. Red line is the model's prediction.

D Experiment Result without Sampling

This section describes the results from the experiments without sampling. That is, the model receives all available input sequences.

D.1 Experiment Results of Toy Dataset

Generated samples on each amplitude-frequency classes are visualized in fig. 8. The overall characteristics are similar to fig. 2 except that GRU can model more diverse samples. we report histograms on amplitude and frequency based on Fourier series analysis in fig. 9. The overall distributions are alike to fig. 3 but GRU covers more wide space in frequency.

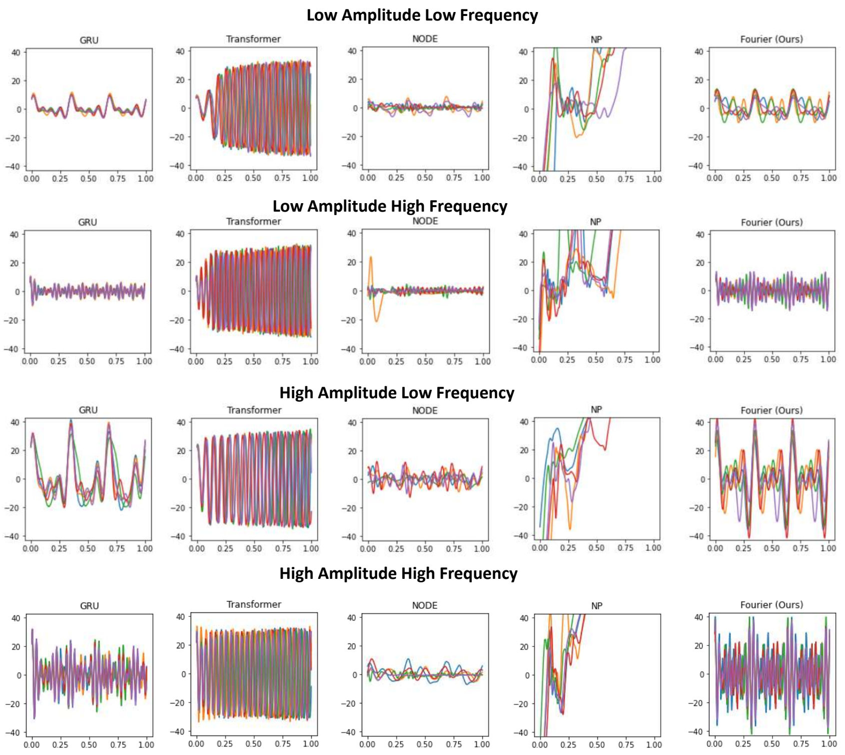

Figure 8: Conditionally generated samples in toy dataset without sampling. We draw five generated samples for each model plotted in different colors.

D.2 Experiment Results of Electrocardiogram

Table 7: Averaged and standard deviation of classifier performance on generated ECG samples

	Overall				AF			LBBB			RBBB		
	Acc	Recall	Precision	F1 Score	Recall	Precision	F1 Score	Recall	Precision	F1 Score	Recall	Precision	F1 Score
GRU	0.333 (0.000)	0.333 (0.000)	0.110 (0.000)	0.167 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	**1.000** (0.000)	0.330 (0.000)	0.500 (0.000)
Transformer	0.557 (0.152)	0.557 (0.152)	0.552 (0.128)	0.446 (0.158)	0.000 (0.000)	0.200 (0.447)	0.000 (0.001)	0.670 (0.456)	**1.000** (0.001)	0.714 (0.409)	**1.000** (0.000)	0.455 (0.070)	0.623 (0.069)
NODE	0.531 (0.010)	0.531 (0.010)	0.574 (0.006)	0.515 (0.013)	0.306 (0.034)	0.476 (0.020)	0.371 (0.023)	0.454 (0.021)	0.773 (0.009)	0.572 (0.015)	0.832 (0.028)	0.471 (0.011)	0.601 (0.007)
NP	0.709 (0.012)	0.709 (0.012)	**0.751** (0.004)	0.710 (0.013)	0.548 (0.046)	**0.734** (0.019)	0.626 (0.024)	0.698 (0.013)	0.932 (0.011)	0.798 (0.006)	0.883 (0.022)	0.588 (0.022)	**0.734** (0.068)
Fourier (Ours)	**0.711** (0.011)	**0.711** (0.011)	0.741 (0.006)	**0.716** (0.011)	**0.779** (0.053)	0.625 (0.032)	**0.692** (0.005)	**0.714** (0.035)	0.954 (0.011)	**0.816** (0.020)	0.642 (0.052)	**0.645** (0.030)	0.642 (0.016)
Real Dataset	0.816 (0.005)	0.796 (0.004)	0.816 (0.005)	0.805 (0.004)	0.737 (0.016)	0.756 (0.014)	0.746 (0.008)	0.777 (0.002)	0.850 (0.007)	0.812 (0.004)	0.873 (0.011)	0.841 (0.009)	0.857 (0.006)

Generated ECG samples from each model are visualized in fig. 10. GRU produces the same samples across all diagnoses. We report the mean and standard deviation of classifier performance on generated ECG samples in table 7. Similar to table 3, our model shows decent performance in diagnosis-averaged overall score and impressive performance in AF. Since GRU generates the same samples regardless of diagnosis, recall in RBBB is 1 and all scores in other diagnoses scores are 0. Likewise, Transformer synthesize AF samples as RBBB, causing recall in RBBB 1 and scores in AF nearly 0.

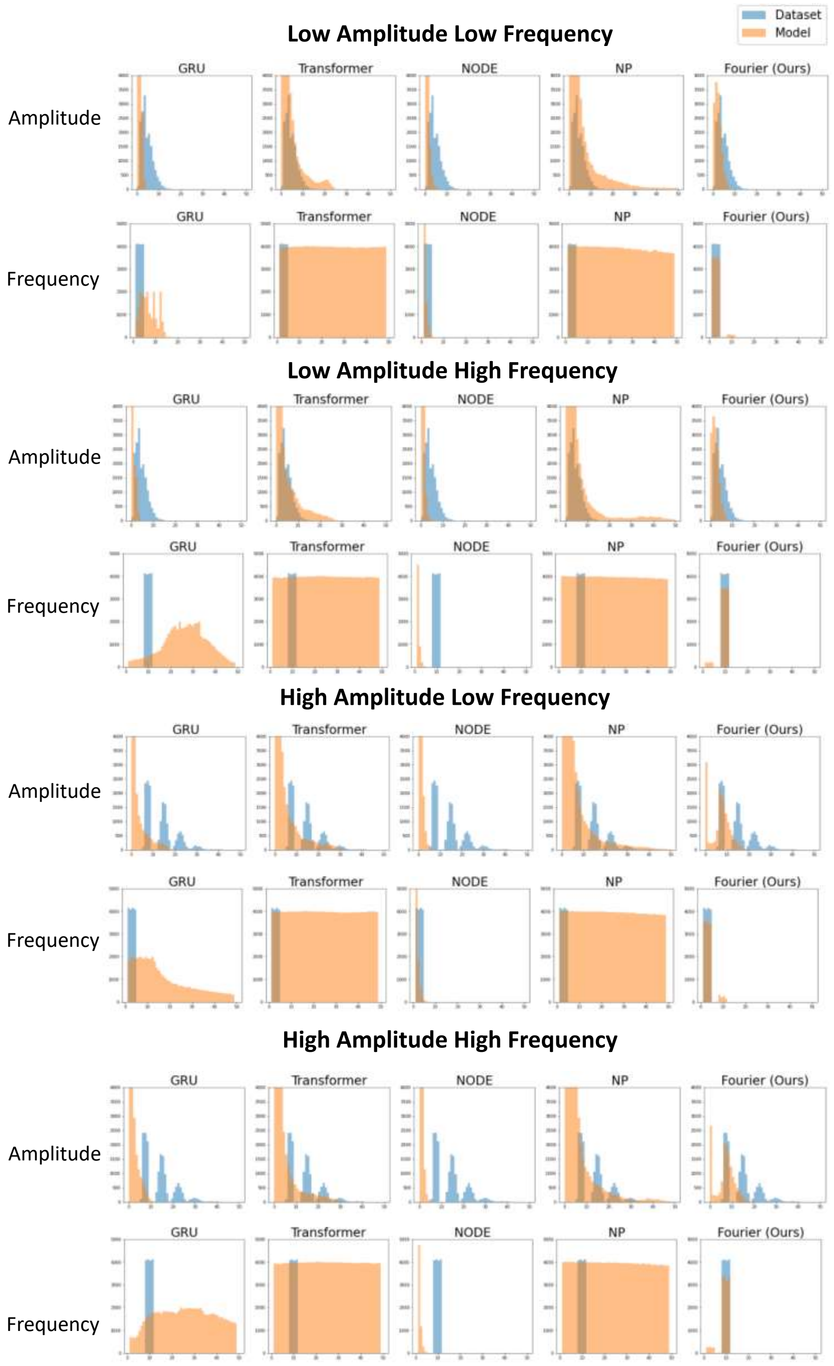

Figure 9: Fourier series analysis on conditionally generated samples without sampling. Blue color represents the original dataset whereas orange color represents each model.

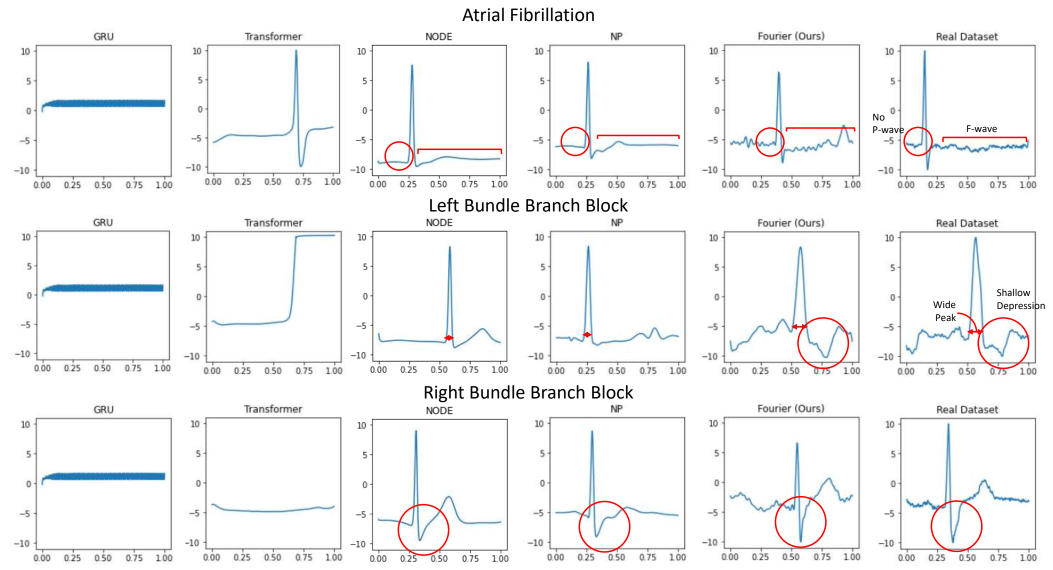

Figure 10: Conditionally generated ECG samples for each diagnosis. Main characteristics for each diagnosis and corresponding spots in generated samples are marked in red.