A. Proof of Proposition 2

Proposition 2. *When the optimally learned encoder and decoder achieve the same joint distribution over* $(\mathbf{x}, \mathbf{t})$ *and* $\mathbf{z}$ *by optimizing* (5)*, for any* $(\mathbf{x}, \mathbf{t})$ *with non-zero probability, if* $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})$ *we have* $g_\theta(\mathbf{z}, \mathbf{t}) = \mathbf{x}$ *almost surely.*

Proof. The joint distribution induced by the encoder is

$$p_{\text{enc}}(\mathbf{x}, \mathbf{t}, \mathbf{z}) = p_{\mathcal{D}}(\mathbf{x}, \mathbf{t}) q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t}).$$

The joint distribution induced by the decoder is

$$p_{\text{dec}}(\mathbf{x}, \mathbf{t}, \mathbf{z}) = p_{\mathcal{I}}(\mathbf{t}) p_z(\mathbf{z}) \delta(\mathbf{x} - g_\theta(\mathbf{z}, \mathbf{t})).$$

When the optimality is achieved so that $p_{\text{enc}} = p_{\text{dec}}$, for $p_{\mathcal{D}}(\mathbf{x}, \mathbf{t}) > 0$ we have

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t}) = \frac{p_{\mathcal{I}}(\mathbf{t}) p_z(\mathbf{z})}{p_{\mathcal{D}}(\mathbf{x}, \mathbf{t})} \delta(\mathbf{x} - g_\theta(\mathbf{z}, \mathbf{t})).$$

Therefore, given $(\mathbf{x}, \mathbf{t})$ such that $p_{\mathcal{D}}(\mathbf{x}, \mathbf{t}) > 0$, for $Z \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})$ we have

$$\begin{aligned} \Pr[\mathbf{x} = g_\theta(Z, \mathbf{t})] &= \int \mathbf{1}\{\mathbf{x} = g_\theta(\mathbf{z}, \mathbf{t})\} q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t}) d\mathbf{z} \\ &= \int q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t}) d\mathbf{z} \\ &= 1. \end{aligned}$$

□

B. Autoencoding Regularization in P-BiGAN

In Section 3.3 we discussed regularizing P-BiGAN with an autoencoding loss using the augmented objective (6). Here we demonstrate the effect of introducing this autoencoding loss in P-BiGAN by comparing the augmented model with the non-regularized counterpart, which is equivalent to the model with the autoencoding coefficient $\lambda = 0$.

Figure 5 compares P-BiGAN with the default strictly-positive λ and the one without autoencoding regularization using $\lambda = 0$ on the MNIST and CelebA imputation experiments. Similarly, Table 3 compares P-BiGAN with the default $\lambda = 1$ and the one without the autoencoding term on the MIMIC-III experiment. It shows that autoencoding regularization improves the performance in almost all the cases. Nonetheless, even without autoencoding regularization P-BiGAN still gives reasonable imputation and classification results. This provides empirical evidence to support the invertibility property stated in Proposition 2.

C. Synthetic Multivariate Time Series

In this section, we equip P-VAE and P-BiGAN with the continuous decoder and encoder described in Section 4 and demonstrate how they work on a synthetic time series dataset using the same architecture described in Section 6.2. We generate a dataset containing 10,000 time series each with three channels over $t \in [0, 1]$ according to the following generative process:

$$\begin{aligned} a &\sim \mathcal{N}(0, 10^2) \\ b &\sim \text{uniform}(0, 10) \\ f_1(t) &= .8 \sin(20(t + a)) + \sin(20(t + a))) \\ f_2(t) &= -.5 \sin(20(t + a + 20) + \sin(20(t + a + 20))) \\ f_3(t) &= \sin(12(t + b)) \end{aligned}$$

where an independent Gaussian noise $\mathcal{N}(0, 0.01^2)$ is added to each channel.

The observation time points for each channel are drawn independently from a homogeneous Poisson process with rate $\lambda = 30$ sampled continuously within $[d, d+0.25]$ where $d \sim \text{uniform}(0, 0.75)$. This results in 7.4 observations in each channel on average. The first row of Figure 7 shows some examples from the generated synthetic dataset.

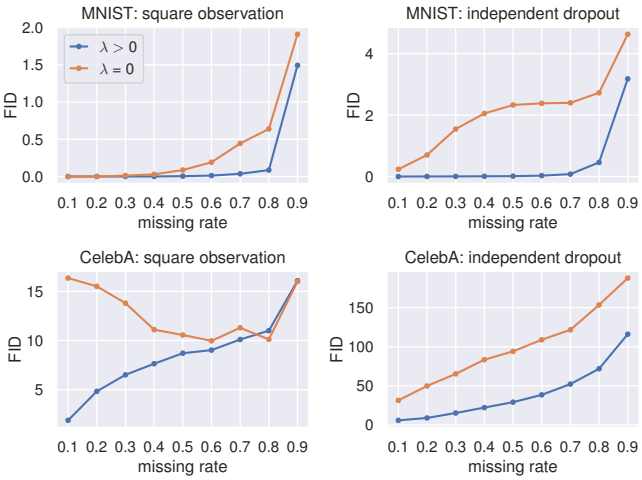

Figure 5. Comparing the effect of autoencoding regularization on the imputation FIDs of P-BiGAN on MNIST and CelebA (no autoencoding regularization when $\lambda = 0$). The high FIDs of the cases of low missing rates on CelebA with square observation are due to the inconsistency between the observed region and the imputed part. Figure 6 shows the FIDs of the generated images under the same settings, from which we can see that the decoder of P-BiGAN performs roughly the same regardless of the autoencoding regularization.

Table 3. Comparing P-BiGAN with autoencoding regularization ($\lambda = 1$) and without it ($\lambda = 0$) on MIMIC-III classification.

AE λ	AUC (%)
$\lambda = 0$	83.56 ± 0.49
$\lambda = 1$	86.02 ± 0.38

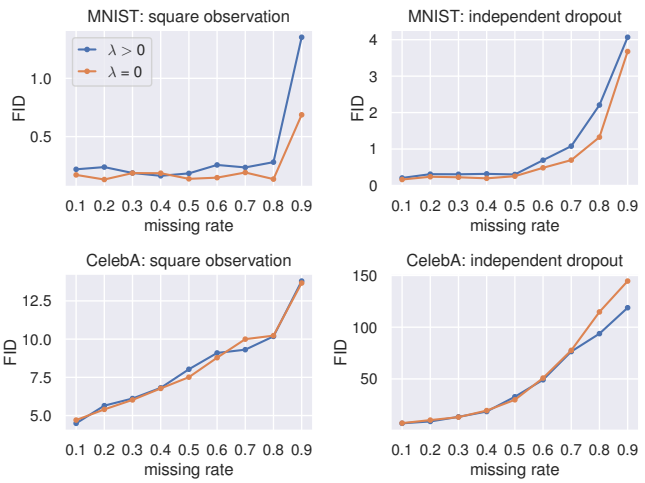

Figure 6. Comparing the effect of autoencoding regularization on the generation FIDs of P-BiGAN on MNIST and CelebA (no autoencoding loss when $\lambda = 0$).

Figure 7 and 8 shows that both P-VAE and P-BiGAN are able to learn the generative distribution reasonably given the sparsely and irregularly-sampled observations. They are both able to learn the periodic dynamics and infer the latent functions according to sparse observations. Moreover, both models also learn that the first two channels are correlated due to the shared random offset a in the generative process, and the shifting of the third channel is uncorrelated to the first two channels as shown in Figure 8.

From the plots, we can see that P-VAE tends to generate smoother curves, while P-BiGAN captures the detailed fluctuation caused by the added Gaussian noise. This is similar to the results on image modeling shown in Section 6.1: GAN-based models capture the local details better but the results can be noisy when the spatial signals are weak. On the contrary, VAE-based models learn the big picture better but the results are usually smoother.

D. On the Independence Assumption

Throughout this paper, we assume the complete temporal process f and the observation indices $\mathbf{t}$ are independent, which corresponds to the missing completely at random (MCAR) case categorized by Little & Rubin (2014). We point out that P-VAE is still unbiased if the data are missing at random (MAR) according to Little & Rubin (2014, Chapter 6).

We note that the introduction of the independence assumption is mainly for better modeling scalability and stability. For the most general situation that corresponds to the not missing at random (NMAR) case, we will need to model $p_{\mathcal{I}}(\mathbf{t}|\mathbf{z})$ explicitly in both P-VAE and P-BiGAN. For time series, we can use the variational RNN (Chung et al., 2015) to model the temporal point process $p_{\mathcal{I}}(\mathbf{t}|\mathbf{z})$.

Our preliminary results show that incorporating $p_{\mathcal{I}}(\mathbf{t}|\mathbf{z})$ makes learning the data distribution harder, especially for P-BiGAN as the discriminator is sensitive to the discrepancy between the learned temporal point process and the empirical samples of observation times. Specifically, modeling the dependency of the temporal point process reduces bias while significantly increasing variance such that the overall model ends up performing worse. The same phenomenon was also reported in the Latent ODE work—Rubanova et al. (2019) jointly model a Poisson process using a Neural ODE, which also leads to worse classification results.

Moreover, learning the temporal point process using variational RNN is quite slow due to the sequential nature of RNNs. Meanwhile, it is unclear how to model such distribution efficiently given the number of observations may be varied from case to case. Therefore, studying how to effectively and efficiently learn the temporal point process and incorporate it in the missing data setting for time series is of interest in the future.

E. Details of Experiments

E.1. Data Preparation and Preprocessing

MNIST can be downloaded from:
`http://yann.lecun.com/exdb/mnist/`

CelebA can be downloaded from:
`http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`

For both MNIST and CelebA, the range of pixel values of the image is rescaled to $[0, 1]$.

MIMIC-III can be downloaded following the instructions from its website:
`https://mimic.physionet.org/gettingstarted/access/`

We follow the GitHub repository below to preprocess the MIMIC-III dataset:
`https://github.com/mlds-lab/interp-net`

For MIMIC-III, we normalize the timestamps within 48 hours to the interval $[0, 1]$. The observed values of the time series are also normalized to $[0, 1]$ according to the minimum and maximum value of each channel across the entire training set.

E.2. Reference Implementations

We use the following reference implementation for the baseline models in our experiments.

MisGAN:

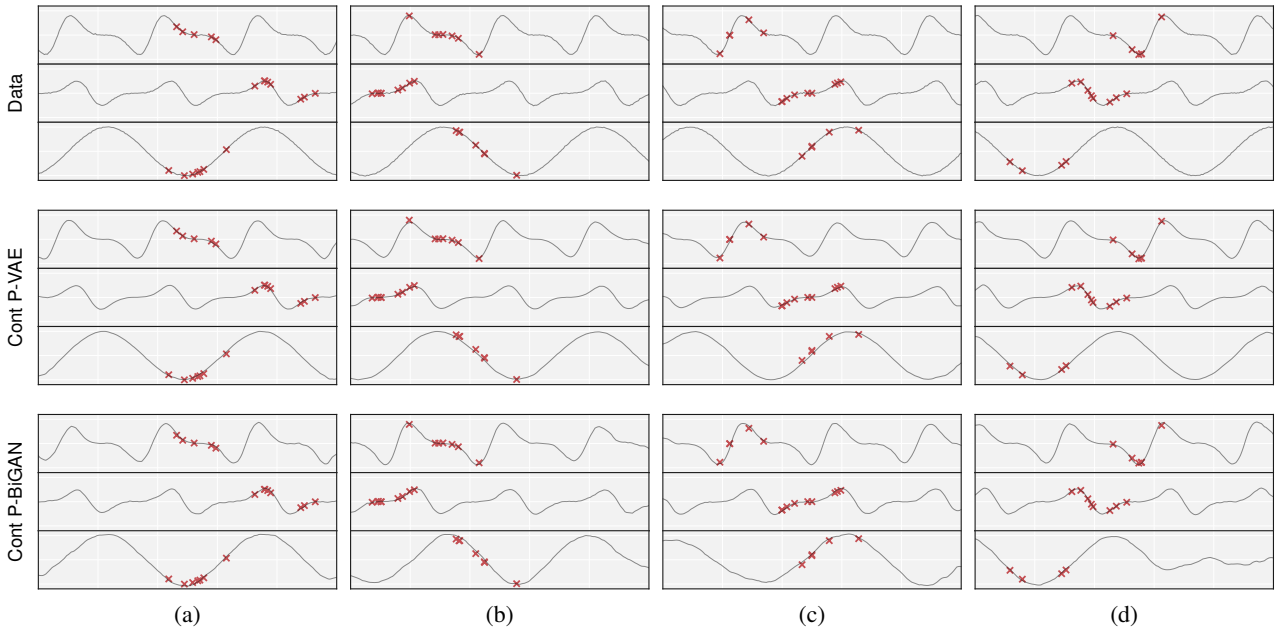

Figure 7. Imputation results of Cont P-VAE and Cont P-BiGAN on a 3-channel synthetic time series. The first row shows four random samples from the training data. Each sample has three channels displayed as a group and the observations in each channel are shown as the red markers, which are drawn from the latent temporal function plotted as the gray trajectory. The second and the third rows show the inferred latent trajectory of each channel, conditioned on the same observations shown in the first row by Cont P-VAE and Cont P-BiGAN respectively. We can see that in general Cont P-VAE produces visually better completion results that are consistent with the overall structure of the training samples. On the other hand, the inferred trajectories of P-BiGAN are less smooth (zoom-in to see the details), and it seems that P-BiGAN captures more easily the Gaussian noise added in the training data. However, P-BiGAN generally produces relatively poor imputation results that do not have the consistent overall structure such as the right tail in channel 3 of case (c) and the right tail in channel 3 of case (d). This is similar to the case of high missing rate with independent dropout missingness in Section 6.1, as the time series are very sparsely observed (7.4 observations in each channel on average). Note that if we trained both model on a more densely sampled time series, such as the one with times drawn from a homogeneous Poisson process with rate $\lambda = 200$, the two models will behave similarly.

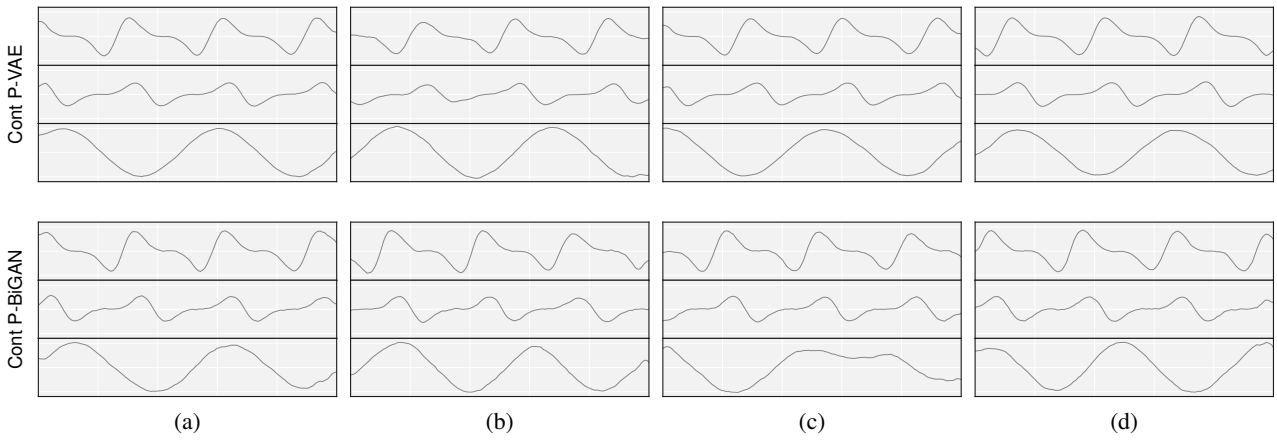

Figure 8. Randomly generated samples by Cont P-VAE (first row) and Cont P-BiGAN (second row) trained on the synthetic time series shown in Figure 7. Similar to the imputation results, Cont P-VAE produces smoother trajectories that are consistent with the ground truth generative process. On the contrary, occasionally there are artifacts in the samples generated by Cont P-BiGAN such as the trajectory of the third channel in case (c).

https://github.com/steveli/misgan

GRU-D:
https://github.com/fteufel/PyTorch-GRU-D

Latent ODE:
https://github.com/YuliaRubanova/latent_ode

M-RNN:
https://github.com/jsyoon0823/MRNN

The continuous convolutional layer described in Section 4.2 is built upon the spline-based convolution operator:
https://github.com/rusty1s/pytorch_spline_conv

E.3. Hyperparameters

Most of the hyperparameters of our models used in the experiments are manually chosen as described in Section 6 without further tuning and are specified in the provided implementation. The only hyperparameter we tune is the strength of the autoencoding loss of P-BiGAN, the coefficient λ in objective (6), for the CelebA experiments. We vary λ from $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and choose the one that yields the best FID. We found that tuning this hyperparameter makes a significant difference for different missing patterns. For block observation, smaller λ yields better results; while for independent dropout, larger λ yields better results.

E.4. Computing Infrastructure

All of our experiments are computed using the NVIDIA GeForce GTX 1080 Ti GPUs.

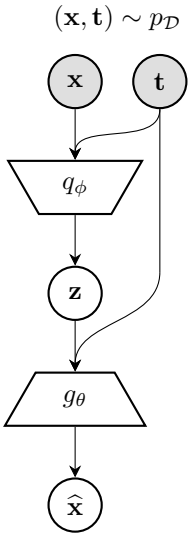

Figure 9. The structure of P-VAE. q_ϕ is the encoder and g_θ is the decoder.

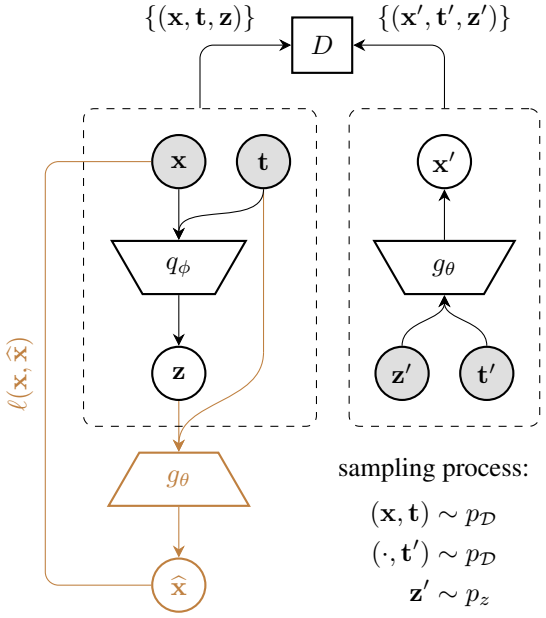

Figure 10. P-BiGAN with autoencoding regularization. q_ϕ is the stochastic encoder. g_θ is the deterministic decoder; the two g_θ share the same parameters. D is the discriminator that takes as input a collection of tuples $(\mathbf{x}, \mathbf{t}, \mathbf{z})$ and $(\mathbf{x}', \mathbf{t}', \mathbf{z}')$. $\ell(\mathbf{x}, \widehat{\mathbf{x}})$ is the autoencoding loss. $p_{\mathcal{D}}$ denotes the empirical distribution of the training dataset $\mathcal{D}$ and p_z is the prior distribution of the latent code $\mathbf{z}$. The part in brown is for additional autoencoding regularization.