GHOSTS: Generation of synthetic hospital time series for clinical machine learning research

Rustam Zhumagambetov[*]
Mathematical Modelling and Data Analysis Department
Physikalisch-Technische Bundesanstalt Braunschweig und Berlin
Berlin, Germany 10587
rustam.zhumagambetov@ptb.de

Niklas Giesa
Institute of Medical Informatics
Charité – Universitätsmedizin Berlin
Berlin, Germany 10117

Sebastian D. Boie
Institute of Medical Informatics
Charité – Universitätsmedizin Berlin
Berlin, Germany 10117

Stefan Haufe
Mathematical Modelling and Data Analysis Department
Physikalisch-Technische Bundesanstalt Braunschweig und Berlin
Berlin, Germany 10587

Abstract

Machine learning (ML) holds great promise to support, improve, and automatize clinical decision-making in hospitals. Data protection regulations, however, hinder abundantly available routine data from being shared across sites for model training. Generative models can overcome this limitation by learning to synthesize hospital data from a target population while ensuring data privacy. Clinical time series acquired during intensive care are, however, difficult to model using established techniques, especially due to uneven sampling intervals. Here we introduce GHOSTS (Generator of Hospital Time Series), a novel generator of synthetic patient trajectories that is capable of generating heterogeneous hospital data including realistic time series with uneven sampling intervals. We further design a suite of novel benchmarks, GHOSTS-Bench. We train GHOSTS on a large cohort of patient data from the MIMIC-IV critical care dataset and measure the quality of the generated data in terms of how faithfully the distributions of individual features in the real data are approximated, how well spatio-temporal dynamics in the multivariate time series are preserved, and how well ML models trained on the generated data can solve a clinical prediction task on the real data. We observe that GHOSTS outperforms a state-of-the-art approach, DoppelGANger, with respect to these criteria.

1 Introduction

Intensive care units (ICUs) are among the most highly digitalized wards in hospitals, continuously aggregating information about a patient's vital signs, laboratory measurements, treatments, and diagnostic labels, among other data. Combined with demographic information and patient data, these data form a patient's electronic health record (EHR). Modern machine learning (ML) models

[*]Corresponding author

Preprint. Under review.

trained on routinely collected EHR data have recently achieved remarkable success in predicting severe outcomes such as acute kidney injury (Tomašev et al., 2019), different complications during perioperative care (Meyer et al., 2018; Giesa et al., 2024), and death (Lichtner et al., 2021; Nistal-Nuño, 2022).

Sharing medical data across institutions is critical for the development of powerful ML models. This typically requires that the data are sufficiently "de-identified", which is often a non-trivial task. Technical solutions to prevent the identification of individual patients include techniques to remove identifying properties in imaging data (Kim et al., 2021), statistics-based approaches based on the concepts of k-anonymity, l-diversity and T-closeness (Samarati and Sweeney, 1998; Machanavajjhala et al., 2007; Li et al., 2007) for tabular data, as well as federated learning (Kaissis et al., 2020). Another promising approach is the creation of synthetic data that closely approximate the distributions of the intended target population while at the same time preventing de-identification. Synthetic data can be generated either in a model-driven or a data-driven way. However, while for certain bodily functions, organs, and systems, detailed biophysical models exist (Maglio et al., 2021), no single biophysical model can describe the complete data that is collected from patients in intensive care units (ICU). On the other hand, generative ML models are able to learn complex distribution from training data. Given a trained model, one can easily sample large amounts of synthetic data from that distribution. Generative models have demonstrated excellent performance in the image processing domain. However, only few models have been proposed to synthesize EHR data collected in ICUs. A potential reason is the heterogeneity of ICU data, which is difficult to model. While some of the collected variables are static (such as demographic and diagnostic labels), the majority of vital signs, laboratory values, and procedures/medications are assessed continuously with distinct mutually inconsistent uneven sampling rates. Another distinction is between continuous-valued features such as blood pressure on one hand and discrete features such as ICD (International Statistical Classification of Diseases and Related Health Problems, World Health Organization (WHO), 2022) codes for clinical diagnoses and procedures. To the authors' knowledge, the only generative model that is currently capable of jointly synthesizing continuous-valued and discrete static and time series data is the DoppelGANger model (Lin et al., 2020). However, being designed for the generation of longitudinal networking data, DoppelGANger does not have a built-in mechanism to faithfully reproduce the spectral characteristics of ICU time series, which includes artifacts introduced by uneven sampling.

In order to aid the development of ML models for clinical prediction taks, synthetic data should fulfill various requirements (Lin et al., 2020). Here, we focus on the following *key quality aspects*.

Faithfulness: The generated data should match the distribution of the training data.

Diversity: Generated data should reflect the full range of patients in the studied cohort, that is, faithfully cover the dispersion and the possible presence of multiple modes and clusters of the training data distribution. While being implied by faithfulness, this property is often violated in practice, where generated data are spread too tightly around a few cluster centers or even individual training samples as in mode collapse.

Utility for downstream prediction tasks: discriminative predictive models trained on the synthetic data should perform well on corresponding real data. While this is again implied by faithfulness, it emphasizes that clinically relevant differences in the data, which could be subtle in relation to the overall variability in the training data, should be particularly well preserved.

The degree to which generated data meet these requirements can be quantitatively assessed using appropriate metrics. However, there are currently no established metrics to measure the quality of synthetic ICU time series data. Addressing these research gaps, the present work makes the following *contributions*.

1. We propose a novel generative model architecture, Generator of Hospital Time Series (GHOSTS), that can generate realistic heterogeneous EHR data consisting of unevenly sampled time series and static attributes that can be either categorical, discrete, or continuous-valued. We train GHOSTS on a large cohort of patients from the public MIMIC IV (Johnson et al., 2023) database.

2. We devise experiments and quantitative metrics to assess the quality of generative models of EHR time series data with respect to faithfulness, diversity, and utility. We use these

to benchmark the synthetic MIMIC data generated by GHOSTS in comparison to data generated by the DoppelGANger method.

The remainder of the paper is structured as follows. In Section 2 we present the technical details of the GHOSTS, the training of GHOSTS to generate synthetic MIMIC data, and the quantitative metrics to measure faithfulness, diversity, and utility. In Section 3, we present the experimental details and results of a comprehensive evaluation of the GHOSTS and DoppelGANger models. The paper ends with a discussion of the results and concluding remarks in Section 4 and Section 5.

2 Methods

2.1 Data preparation

MIMIC IV (Johnson et al., 2023) is a large database of deidentified data of more than 300,000 patients (Johnson et al., 2023). To extract EHR data from the MIMIC IV tables, we used routines from the MIMIC IV companion repository[2]. We removed outliers by only keeping values within the physiological range as defined in Harutyunyan et al. (2019). We extracted the following $|F| = 10$ ICU time series features: $F = \{$mean blood pressure, diastolic blood pressure, systolic blood pressure, heart rate, respiratory rate, SpO2, temperature, sodium, and glucose$\}$. These features were selected due to their high importance for clinical decision making in ICUs.

An ICU stay is defined as the time between the first and last available measurement from the heart rate monitor. We extracted signals within the first 48 hours of the ICU stay. We excluded stays that had any of the ICU time series features completely missing, leaving 72,428 ICU stays. In addition, we also extracted static patient attributes including patient age and gender, and the sequential organ failure assessment (SOFA) score (Vincent et al., 1996). The prediction of SOFA scores defines the downstream task we use to measure the utility of the generated synthetic data. The SOFA score was extracted using SQL queries provided as part of MIMIC IV (Johnson et al., 2023) within a 54-hour window starting 6 hours before the ICU entry and ending at the 48-hour mark. Patients with SOFA scores below 5 are marked as SOFA$^-$, while patients with SOFA scores above 8 are marked as SOFA$^+$, and patients with SOFA scores in between are marked as SOFA^o. The numbers of ICU stays marked SOFA$^-$ and SOFA$^+$ was 39,845 and 11,199, respectively, and the number of ICU stays with in-between scores was 21,384. To preserve class distribution balance 11,000 stays were sampled from each class.

To prepare data in a suitable format for training ML models, it is necessary to harmonize the different irregular sampling rates of the individual measurements. To this end, the 48 hours worth of data were upsampled by assigning each 10-min interval the value of the closest temporally preceding measurement, and interpolated to a regular sampling interval of 10 minutes using forward fill. This resulted in $|T| = 288$ time points; where $T = \{t_i\}$ denotes the sequence of time points. Next, we applied forward fill followed by median imputation on the whole series to fill in the missing values.

ICU stays were randomly assigned into a training and a test split, where we stratified patients by their SOFA score class (SOFA$^{-/+}$) to achieve a balanced split. We extracted $N_{\text{train}} = 21,000$ ICU stays for the training split and $N_{\text{test}} = 1,000$ disjoint stays for the testing split. Each ICU stay comprises an $|F| \times |T|$ matrix $\mathbf{d}_{\text{time}} = (d_{f,t})$ of time series data, where $f \in F$ denotes feature and $t \in T$ is a time index. In addition, each ICU stay is associated with static patient attributes $\mathbf{d}_{\text{attr}} = (s, g, a)$, where $s \in \{\text{SOFA}^-, \text{SOFA}^o, \text{SOFA}^+\}$ is a SOFA score class, $g \in \{F, M\}$ is biological sex, and $a \in \mathbb{N}^+$ is the patient age at the time of ICU admission. The MIMIC training and test datasets are denoted by $\mathcal{D}^{\text{MIMIC}}_{\text{train}} = \{(\mathbf{d}_{\text{time}}, \mathbf{d}_{\text{attr}})_i : 0 < i < N_{\text{train}}\}$ and $\mathcal{D}^{\text{MIMIC}}_{\text{test}} = \{(\mathbf{d}_{\text{time}}, \mathbf{d}_{\text{attr}})_i : 0 < i < N_{\text{test}}\}$, where $\mathcal{D}^{\text{MIMIC}}_{\text{train}} \cap \mathcal{D}^{\text{MIMIC}}_{\text{test}} = \emptyset$.

2.2 Generative modeling of EHR time series

The generation of synthetic data is an unsupervised problem in which a model learns the probability distribution of a set of interdependent variables from observed training data. Suppose we have training data $\mathcal{D}^{\text{train}} = \{\mathbf{d}_i : \mathbf{d}_i \in \mathbb{D} \subseteq \mathbb{R}^R, 0 < i < N_{\text{train}}\}$ sampled from a distribution P_{true}. To empirically

[2]Vital signs: `https://github.com/MIT-LCP/mimic-code/blob/main/mimic-iv/concepts_postgres/measurement/vitalsign.sql`

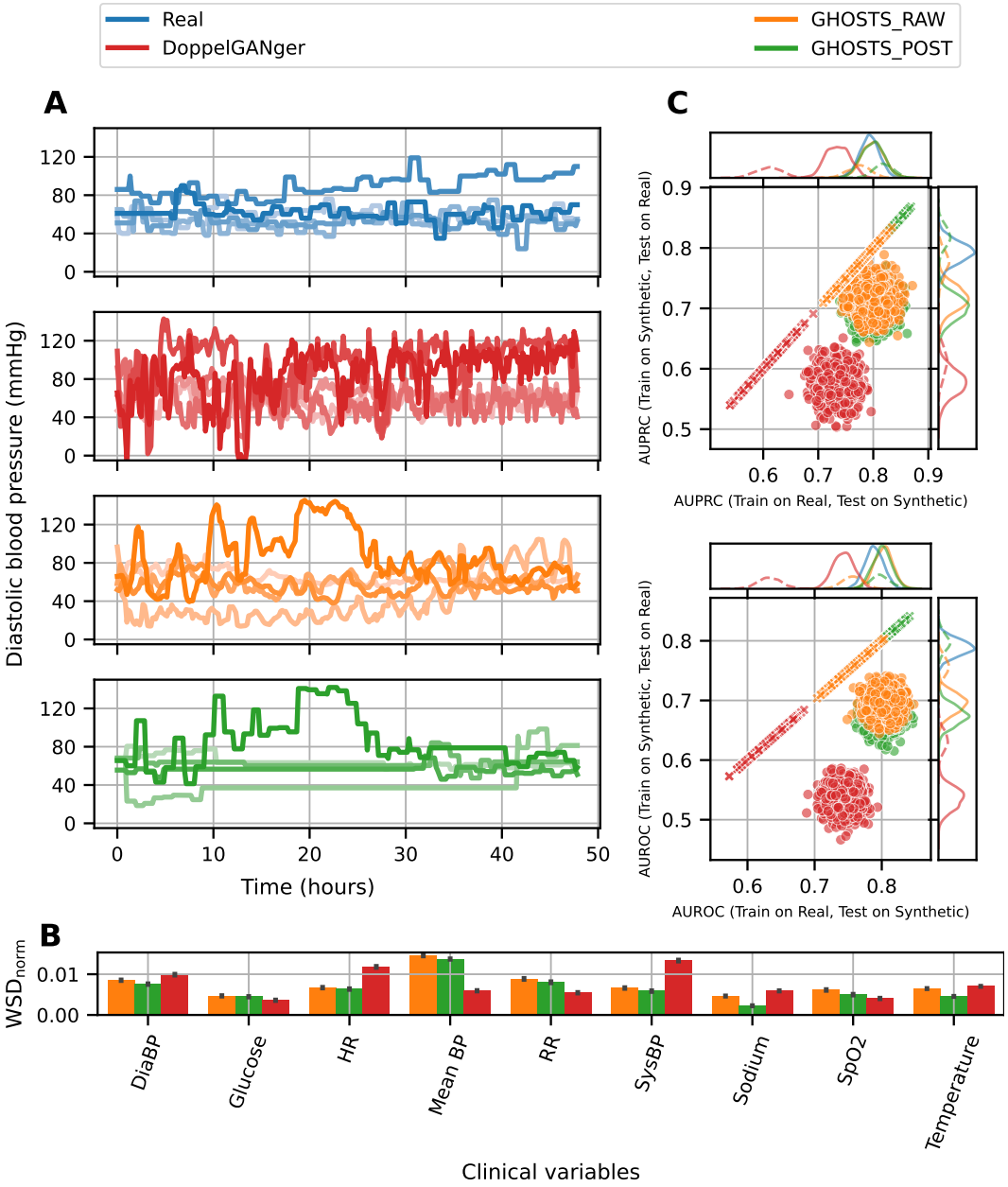

Figure 1: Characteristics of real intensive care unit (ICU) time series data compared to synthetic data generated by the DoppelGANger method as well as GHOSTS with and without postprocessing (GHOSTS_RAW and GHOSTS_POST). Real data were obtained from the MIMIC-IV database and used to train the GHOSTS and DoppelGANger models. **Panel A**: Examples of real and synthetic diastolic blood pressure time series. Curves with different color tones correspond to independent samples. **Panel B**: Comparison of feature-wise univariate empirical distributions of generated and real time-series in terms of the $\mathrm{WSD}_{\mathrm{norm}}$ (normalized Wasserstein distance). **Panel C**: Performance of combined MLP-LSTM classifiers using time series features to distinguish high and low sequential organ failure assessment (SOFA) scores of ICU patients. Performance is evaluated in terms of the area under the precision-recall curve (AUPRC, top panel) and the area under the receiver operating characteristics curve (AUROC, bottom panel). Each colored circle corresponds to a bootstrap sample of real and synthetic test sets, where the y-coordinate encodes the performance of classifiers trained on synthetic and tested on real data, while the x-coordinate represents the opposite case. Density curves on the sides show marginal performance distributions. Dashed curves and colored crosses correspond to classifiers trained and tested on synthetic data.

estimate P_{true}, we define a parametric family of probability densities, $(P_{\boldsymbol{\theta}})_{\boldsymbol{\theta}\in\mathbb{R}^P}$ (Arjovsky et al., 2017). The task is then to find the parameter vector $\boldsymbol{\theta}$ that maximizes the likelihood of the training data, that is, $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \log P_{\boldsymbol{\theta}}(\mathbf{d}_i)$. Generative ML models implicitly represent $P_{\boldsymbol{\theta}}$ by a learnable generator function $G_{\boldsymbol{\theta}} : \mathbb{Z} \to \mathbb{D}$ that maps random inputs $\mathbf{z} \in \mathbb{Z} \subseteq \mathbb{R}^Q$ to samples $\tilde{\mathbf{d}}$.

Established families of generative models include generative adversarial network (GAN, Goodfellow et al., 2014), variational autoencoders (Kingma and Welling, 2019), and diffusion-based models (Sohl-Dickstein et al., 2015). In the context of EHR generation, predominantly GANs have been used. A GAN consists of two parts: the generator $G_{\boldsymbol{\theta}}$ and a discriminator $D_{\boldsymbol{\vartheta}}$, where $\boldsymbol{\vartheta} \in \mathbb{R}^S$. The discriminator is a classifier that tries to distinguish generated from real training samples, thereby evaluating the realism of the generated data. The aim of $G_{\boldsymbol{\theta}}$ is then to minimize the loss of $D_{\boldsymbol{\vartheta}}$, and the objective of $D_{\boldsymbol{\vartheta}}$ is to maximize the loss of $G_{\boldsymbol{\theta}}$. The two networks are, thus, competing with each other, as reflected by the min-max GAN cost function $L = \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\vartheta}} \mathbb{E}_{\mathbf{d}\sim P_{\text{train}}}[\log D_{\boldsymbol{\vartheta}}(\mathbf{d})] - \mathbb{E}_{\tilde{\mathbf{d}}\sim P_G}[\log(1 - D_{\boldsymbol{\vartheta}}(\tilde{\mathbf{d}}))]$, where P_{train} is the empirical training data distribution, $D_{\boldsymbol{\vartheta}}(\mathbf{d})$ is the estimated probability that $\mathbf{d}$ is a sample from P_{train}, P_G is the distribution obtained by sampling the generator via $\mathbf{z} \sim \mathcal{U}, \tilde{\mathbf{d}} = G_{\boldsymbol{\theta}}(\mathbf{z}), \mathcal{U}$ is a uniform distribution, and $\mathbb{E}[\cdot]$ denotes expectation.

It is well-known that GANs can suffer from mode collapse and the vanishing gradient problem (e.g., Arjovsky et al., 2017). To overcome these problems, improvements to the original GAN model were suggested (Arjovsky et al., 2017; Gulrajani et al., 2017). Arjovsky et al. (2017) introduced the Wasserstein GAN (WGAN), demonstrating that the use of the Wasserstein distance as a loss function can mitigate mode collapse. Unfortunately, precise computation of Wasserstein distances is intractable. Instead, the Kantorovich-Rubinstein duality, which holds for K-Lipschitz functions, is used to approximate it (Arjovsky et al., 2017). The resulting loss minimized by WGANs is $L = \min_{\boldsymbol{\theta}} \max_{D\in\mathcal{L}} \mathbb{E}_{\mathbf{d}\sim P_{\text{train}}}[D(\mathbf{d})] - \mathbb{E}_{\tilde{\mathbf{d}}\sim P_G}[D(\tilde{\mathbf{d}})]$, where $\mathcal{L}$ is a set of 1-Lipschitz functions. Practically, replacing the GAN by the WGAN loss function implies that the discriminator is not anymore required to output strict probabilities between 0 to 1. This is considered to prevent both the vanishing gradients and mode collapse problems (Arjovsky et al., 2017). However, the loss formulation preserves its guarantee only under the assumption that the discriminator is K-Lipschitz. To ensure this, Arjovsky et al. (2017) proposed to clamp the discriminator weights after each update. Further work by Gulrajani et al. (2017) complements weight clipping by introducing an additional penalty on the gradient norm for random samples from the training distribution, leading to the extended loss function

$$L^{\text{WGAN\_GP}} = \min_{\boldsymbol{\theta}} \max_{D\in\mathcal{L}} \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\text{train}}}[D(\mathbf{d})] - \mathop{\mathbb{E}}_{\tilde{\mathbf{d}}\sim P_G}[D(\tilde{\mathbf{d}})] + \lambda \mathop{\mathbb{E}}_{\hat{\mathbf{d}}\sim P_{\hat{\mathbf{d}}}}[(\|\nabla_{\hat{\mathbf{d}}} D(\hat{\mathbf{d}})\|_2 - 1)^2], \quad (1)$$

where λ is a scaling coefficient and $\hat{\mathbf{d}} \sim P_{\hat{\mathbf{x}}}$ is a point uniformly sampled to lie between a random pair of points drawn from P_{train} and P_G

GAN architectures have been introduced and extensively used for image generation tasks but the concepts have also been extended to EHR data. Most existing approaches, however, are restricted to generating either static EHR attributes or time series data but not both. We discuss these in Appendix A. To our knowledge, the only existing method that is capable of jointly generating static and time-variant data is DoppelGANger (DG, Lin et al., 2020). DG is a WGAN with an additional gradient penalty (Gulrajani et al., 2017) designed for the generation of longitudinal networking data. Its loss is, therefore, equivalent to Equation 1. DG decouples the generation of static and time series data by first generating synthetic static data, then conditioning the time series generation on the newly generated static data. To do that it uses dense layers for the generation of the static features (tabular data) and a long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) for the generation of the time series. Second, there are two discriminators: one for the combined static and time series data, and another one only for the static attributes. Note that a separate attribute discriminator is included to ensure that the low-dimensional static attributes are not dominated by the high-dimensional time series data and that equal importance is given to both types of features in the overall cost function.

2.3 Proposed Method: Generator of Hospital Time Series (GHOSTS)

GHOSTS is an extension of the DG method designed to overcome its limitations when applied to ICU time series. To ensure the conservation of characteristic spatio-temporal dynamics in the generated

time series, GHOSTS employs additional mechanisms. Most importantly, we add two regularizing terms to the original DG loss function. The GHOSTS cost function reads

$$\begin{aligned} L^{\text{GHOSTS}} = &\min_{\boldsymbol{\theta}} \max_{D \in \mathcal{L}} \mathbb{E}[D(\mathbf{d}_{\text{time}}, \mathbf{d}_{\text{attr}})] - \mathbb{E}[D(\tilde{\mathbf{d}}_{\text{time}}, \tilde{\mathbf{d}}_{\text{attr}})] \\ &+ \lambda \, \mathbb{E}[(\|\nabla_{(\hat{\mathbf{d}}_{\text{time}}, \hat{\mathbf{d}}_{\text{attr}})} D(\hat{\mathbf{d}}_{\text{time}}, \hat{\mathbf{d}}_{\text{attr}})\|_2 - 1)^2] \\ &+ \alpha \sum_{f \in F} \|(\tilde{\mathbf{d}}_{\text{time}} \boldsymbol{\Gamma})_f\|_1 + \beta \sum_{f \in F} \|(\tilde{\mathbf{d}}_{\text{time}} \boldsymbol{\Delta})_f\|_1 \, , \end{aligned} \tag{2}$$

where $(\mathbf{d}_{\text{time}}, \mathbf{d}_{\text{attr}}) \sim P_{\text{train}}$, $(\tilde{\mathbf{d}}_{\text{time}}, \tilde{\mathbf{d}}_{\text{attr}}) \sim P_G$, $(\hat{\mathbf{d}}_{\text{time}}, \hat{\mathbf{d}}_{\text{attr}}) \sim P_{(\hat{\mathbf{d}}_{\text{time}}, \hat{\mathbf{d}}_{\text{attr}})}$, α and β are positive scaling coefficients, and $\|\cdot\|_1$ denotes the ℓ_1-vector-norm. $\boldsymbol{\Gamma}$ is the $|T| \times |T|$ linear type-II discrete cosine transform (DCT) matrix (Ahmed et al., 1974) with entries $\Gamma_{t,k} = 2\cos\left(\frac{\pi k(2t+1)}{2|T|}\right)$. Similarly, $\boldsymbol{\Delta}$ denotes the $|T| \times (|T| - 1)$ discrete first-order differencing operator with respect to the time dimension with entries $\Delta_{t,k} = -1$ if $t = k$, $\Delta_{t,k} = 1$ if $t = k + 1$, and $\Delta_{t,k} = 0$ otherwise.

The fourth term encourages sparsity of each individual time series feature in the spectral domain. This is achieved by penalizing the ℓ_1-norm of the coefficients of a temporal discrete cosine transform, which drives some frequency coefficients to zero. While the spectrum is not supposed to be completely sparse, it is expected to be dominated by higher harmonic frequencies of each feature's individual sampling frequency. Frequencies in between these higher harmonics are, therefore, expected to be absent, implying the respective DCT coefficients to be zero. The fifth term penalizes the ℓ_1-norm of the 1^{st} temporal derivative of each time series. This leads to sparsification meaning that, with β chosen to be appropriately high, most temporal differences of the generated data will be zero (c.f. the concept of total variation denoising, Rudin et al., 1992). In the time domain, this corresponds to piecewiese constant time courses, which is consistent with the structure of the data (c.f., Figure 1. This penalty, thus, encodes the preferences for sampling intervals larger than 10 minutes while allowing for uneven and inconsistent sampling schemes across features.

The GHOSTS architecture consists of the generator $G_{\boldsymbol{\theta}}$ and the discriminator D. The generator consists of two neural networks. The first one is a static attribute generator implemented as a multi-layer perceptron (MLP) with three dense layers, ReLU activations, and a batch normalization layer. The second one is an LSTM network acting as a time series generator. In analogy to the DG architecture, discriminator consists of two independent networks: a general discriminator and an attribute-only discriminator. The general discriminator is applied to the combination of the attribute and time series, and the attribute discriminator is applied only to the static attributes. Both are MLPs with five dense layers, ReLU activations, and a minibatch discrimination bypass layer.

Since we observed mode collapse during the development of GHOSTS, we equip the discriminator with minibatch discrimination (MBD) layers as proposed by Salimans et al. (2016). The idea of MBD is to provide information about the samples in a batch to the discriminator and to penalize a lack of diversity in the batch. Minibatch discrimination is practically implemented as a layer parallel to the discriminator accepting the same input, but the outputs of the minibatch discrimination layer are concatenated with the output of the discriminator and passed to the final linear layer. We train GHOSTS using the Adam optimizer and reuse hyperparameter settings proposed for DoppelGANger (Lin et al., 2020). Prior to training, all features are scaled to values between 0 and 1 using a maximum absolute scaling scheme. This scaling is reversed before the generated data are returned. The regularization constants α and β are set to $\alpha = 0.1$ and $\beta = 10$. The number of latent dimensions of the MBD layers is set to 5. Values for other hyperparameters are specified in-text.

Postprocessing

We observed that, despite the structural extensions to the existing DG methods described above, the generated data still contained subtle structural deviations from the real data. Real data contain certain regularities that are difficult to enforce with loss functions alone. Specifically, different physiological measures are constrained to attain only a finite number of different values due to quantization effects induced by the measurement process. Moreover, time series of distinct physiological quantities are also characterized by a finite number and characteristic distribution of sampling intervals. Accounting for these observations we devised the following routine to align the distributions of raw feature values and sampling intervals produced by GHOSTS to the empirical training distributions.

1. Each feature value in the generated time series is replaced by the closest value from the featurewise set of discrete values.
2. For each feature, the empirical distribution of the sampling (estimated using histogram with bin width 1) intervals is estimated from the training data.
3. For each feature of each generated sample, new sampling points are defined by drawing randomly from the empirical distribution estimated from the training data in step 2.
4. The generated time series are downsampled to the sampling points defined in step 3 by taking the values at the sampling point and forward filling until the next sampling point.

2.4 Performance metrics

The generation of synthetic hospital time series is a relatively novel field. When assessing the quality of generated data, their nature as time series with characteristic spatio-temporal dynamicsneeds to be taken into account, which requires dedicated performance metrics. Here, we discuss quantitative metrics to assess the faithfulness of generated data with respect to the training distribution as well as their utility with respect to a binary downstream classification task.

Faithfulness and diversity

Comparing the distributions of such high-dimensional hospital time series is non-trivial, especially in the presence of rich spatio-temporal correlations within and between features. Here, we evaluate the closeness of distributions up to second-order moments, capturing the faithful reproduction of spatio-temporal correlations in the generated data.

Similarity of raw feature values We evaluate the similarity of the univariate distributions of the values of individual features in the real and synthetic training data by calculating the Wasserstein distance (WSD, Ramdas et al., 2017) between these distributions. In this univariate case, the WSD between probability density functions reduces to the total area between the corresponding cumulative distribution functions (CDF, c.f., Ramdas et al., 2017; Lipp and Vermeesch, 2023). Since different time series features have distinct ranges, we normalize Wasserstein distances per feature by the difference between the attained maximum and minimum values. Normalized WSD (WSD_{norm}) are calculated based on empirically estimated CDFs, see Subsection B.1 for details.

Similarity of second-order statistics We also measure the similarity of second-order statistics of real and generated data. Based on feature by feature correlation matrices, we define the following metrics: adapted correlation accuracy (CorrAcc, Tao et al., 2021; Li et al., 2023), mean absolute difference (CorrMAD, Li et al., 2023), and mean squared error (CorrMSE). In addition, we measure the reproduction of auto- and cross-spectral statistics based on lagged auto- and cross-correlation matrices, giving rise to two further metrics: autocorrelation divergence (AD), and cross-correlation divergence (CCD). These metrics are described in detail in Subsection B.2

Utility

To assess the utility of the synthetic data in a realistic clinical prediction setting, we consider the prediction of high or low SOFA scores from time series data. Given complete real and synthetic training and test datasets, static attributes $\mathbf{d}_{\text{attr}}$ are removed to yield classifier inputs and corresponding SOFA score labels (outputs).Only samples with either high (SOFA^{+}) or low (SOFA^{-}) SOFA score are included, defining a binary classification task. We investigate all four combinations of training a model on either real or synthetic training data and testing the model on either real or synthetic data test data. These combinations are referred to as 'train on synthetic, test on real' (TSTR), 'train on real, test on real' (TRTR), 'train on synthetic, test on synthetic' (TSTS), and 'train on real, test on synthetic' (TRTS). Utility is defined as the performance of a model trained on synthetic data when applied to real data (TSTR case), whereas the TSTS and TRTR cases provide upper bounds of the performance to be expected.

As classifiers, we use two different models denoted LSTM and LSTM-MLP as described in Appendix C. Test performance is measured using the area under the receiver operating characteristics curve (AUROC) and the area und er the precision-recall curve (AUPRC), see again Appendix C.

3 Results

All experiments reported here were performed on a workstation with 128 cores, 256 GB of memory, and an A100 (40GB memory) GPU. Computation of the complete experiments took approximately 7 days.

First, we trained the GHOSTS and DG models on the $\mathcal{D}_{\text{train}}^{\text{MIMIC}}$ dataset as outlined for GHOSTS in Subsection 2.3. For DG, we use the author's implementation[3] with its default parameters (Lin et al., 2020). Next, we sampled $N_{\text{train}} = 17,600$ and $N_{\text{test}} = 4,400$ samples using each model, giving rise to datasets $\mathcal{D}_{\text{train}}^{\text{DG}}$, $\mathcal{D}_{\text{test}}^{\text{DG}}$, $\mathcal{D}_{\text{train}}^{\text{GHOSTS\_RAW}}$, and $\mathcal{D}_{\text{test}}^{\text{GHOSTS\_RAW}}$. After postprocessing the synthetic GHOSTS data, we obtained additional datasets $\mathcal{D}_{\text{train}}^{\text{GHOSTS\_POST}}$ and $\mathcal{D}_{\text{test}}^{\text{GHOSTS\_POST}}$. We sampled data unconditionally and confirmed that both GHOSTS and DoppelGANger preserved the distribution of class labels in the generated datasets resulting in an approximately 50:50 split.

Table 1: Quantitative evaluation of how faithfully synthetic MIMIC data generated by DoppelGANger, GHOSTS without postprocessing (GHOSTS_RAW) and GHOSTS with postprocessing (GHOSTS_POST) reproduces aspects of real MIMIC data. WSD_{norm} refers to the normalized Wasserstein distance between univariate distributions of raw values of individual time series features presented in Equation (3). CorrAcc, CorrMSE, and CorrMAD refer to correlation accuracy, mean squared error, and mean absolute difference of real and synthetic $|F| \times |F|$ temporal Pearson correlations between features, corresponding to Equations (8), (10), and (9). AD, CCD denote auto- and correlation divergences derived from $|F| \times |F| \times N_{\text{lag}}$ lagged cross-covariance matrices, respectively, corresponding to Equations (13) and (15). Data are reported as mean ± standard error of the mean, and ↓/↑ indicate that lower/higher values are better, respectively. Uncertainty was estimated using bootstrap method (K=1,000)

	DoppelGANger	GHOSTS_RAW	GHOSTS_POST
WSD_{norm} ↓	0.0074 ± 0.0001	0.0075 ± 0.0001	**0.0064 ± 0.0001**
CorrAcc ↑	0.8611 ± 0.0000	**0.9444 ± 0.0000**	0.8889 ± 0.0000
CorrMSE↓	0.0043 ± 0.0001	**0.0006 ± 0.0000**	0.0033 ± 0.0001
CorrMAD↓	0.0433 ± 0.0005	**0.0182 ± 0.0005**	0.0337 ± 0.0005
AD↓	0.0184 ± 0.0003	**0.0146 ± 0.0004**	0.0307 ± 0.0005
CCD↓	0.3497 ± 0.0031	**0.3108 ± 0.0043**	0.6648 ± 0.0052

The quality of the generated data was assessed in terms of their faithfulness, diversity, and utility with respect to SOFA score classification using the metrics outlined in Subsection 2.4 and Appendix C.

Table 1 summarizes the results for different faithfulness metrics. Uncertainty was estimated using the bootstrap method (K=1,000). Across all metrics, GHOSTS achieves either superiority over DG, or comparable results. Overall, GHOSTS_RAW outperforms DG in all but one metric, and is on par in terms of the WSD_{norm} metric, whereas GHOSTS_POST outperforms DG in terms of CorrAcc, CorrMSE, and CorrMAD but not in terms of AD and CCD. Interestingly, GHOSTS_RAW tends to perform better than GHOSTS_POST according to all but one (WSD_{norm}) metric. Overall, these results indicate that both GHOSTS variants represent a marked improvement compared to DoppelGANger, although the postprocessing does not seem to lead to a more faithful representation of spatio-temporal dynamics. To break down results to the individual clinical variables, we compare time-series distributions across features. As can be seen in Figure 1 B, both GHOSTS varients outperform DG for five out of nine features in terms of the most general WSD_{norm} metric.

Figure 1 C depicts the utility of the synthetic data for the SOFA score classification task, which is operationalized as the performance of LSTM-MLP classifiers and measured using the AUPRC and AUROC metrics. Blue color marks the performance of the baseline model, which was trained and also tested on real MIMIC data. The x-axis shows the performance of models trained on real data and tested on synthetic data generated by different models. Conversely, the y-axis shows the performance of model trained on synthetic and tested on real data. Each colored circle represents the performance of these classifiers on one bootstrap sample drawn from the respective test sets. Dashed curves and colored crosses correspond to classifiers trained and tested on synthetic data. Training and testing on synthetic GHOSTS data leads to high performance, similar to what is achieved by models trained

[3] `https://github.com/fjxmlzn/DoppelGANger`

and tested on real data. This indicates that GHOSTS learned to generate discriminative features. The performance is similar for models trained on real MIMIC data (TRTS settings) and deteriorates moderately in the relevant setting in which models are tested on real MIMIC data (TSTR setting). Notably, GHOSTS with and without postprocessing substantially outperforms DoppelGANger in all settings and with resepect to both metrics. Refer to Table 2 for a tabular summary of the same results.

4 Discussion

We presented GHOSTS, a novel generative model architecture tailored to generate realistic EHR data comprising heterogeneous longitudinal and tabular data types. Previous methods are either unable to generate medical time series data such as data collected in ICUs or tend to generate unrealistic data with even sampling patterns and generated values that are, frequently outside of the range of real training data. Moreover, EHR data generated by existing methods often shows signs of mode collapse, indicating convergence issues during model training. Through the introduction of novel loss functions and postprocessing routines, GHOSTS overcomes these limitations and is the first method that is able to generate unevenly sampled realistic ICU time series. We trained and validated GHOSTS on a large patient cohort extracted from the MIMIC IV database. The results indicate a significant gain over the state-of-the-art DoppelGANger model both in terms of the faithful reconstruction of essential features distributions in the MIMIC data and in terms of the classification performance in a clinical downstream classification task.

As part of the numerical benchmarks conducted in this work, we trained GHOSTS on the large MIMIC IV dataset, thereby creating a synthetic MIMIV IV patient cohort. We plan to publish this dataset, the trained MIMIC-GHOSTS model as well as all code associated with data extraction, pre- and postprocessing, model design and training, and performance evaluation in the near future. We expect that the public availability of these tools will promote machine learning research in the ICU field and lead to the development of novel methods and models. Specifically, we expect GHOSTS to enable in-silico clinical trials, where it is necessary to generate synthetic data for separate control and treatment groups. GHOSTS already has a built-in mechanism to condition its output on arbitrary static attributes. Thereby, it is able to generate counterfactual patient trajectories that can be used to estimate treatment effects (Myles et al., 2023). This is especially relevant in intensive care, where clinical trials frequently cannot be conducted due to ethical concerns.

GHOSTS is still under refinement, though, and the results presented here raise several questions and need to be considered preliminary. One current observation is that the devised postprocessing routine does seem to improve the overall faithfulness as measured by $\mathrm{WSD}_{\mathrm{norm}}$. However, it also seems to adversely affect the reconstruction of spatio-temporal dynamics encoded in the auto- and cross-spectra of the data. Notwithstanding, this result had negligible influence on the performance of ML models in a clinically relevant classification task.

Besides the method itself, this study also has limitations. While we tried to exhaustively measure the quality of the generated data with different performance metrics, we did not assess the realism of the generated data as perceived by human practitioners, which could be an important factor influencing the acceptance of ML models trained on synthetic data by clinicians. In future work, we plan to conduct studies with clinical experts to guide the assessment and further development of our model. We also plan to train GHOSTS on additional patient cohorts and to study the across-site transferability of clinical prediction models and the possible role of synthetic data to improve transferability. Finally, our quantitative evaluation focuses on the aspects of faithfulness, diversity, and utility. In that, we have spared out further important desiderata. Most importantly, it is of utmost importance that synthetic data fulfill privacy requirements. While the design of GHOSTS effectively prevents mode collapse, there is no theoretical guarantee to exclude that the model may memorize and reproduce individual patients' data. Future studies will, therefore, benchmark GHOSTS with respect to privacy aspects and common attacks on privacy.

5 Conclusion

In summary, we have presented GHOSTS as the first generative model capable of generating realistic ICU times with associated static patient attributes. As demonstrated on a large ICU data corpus, data synthesized by GHOSTS accurately resembles real training data and preserves details such as

artifacts introduced by uneven sampling intervals of different time series measurements. Moreover, data generated by GHOSTS can be used to train machine learning models to achieve competitive performance on real data in clinical prediction tasks. Future work will extend, further scrutinize, and, eventually, publish GHOSTS as an open-access resource for clinical research.

Acknowledgments and Disclosure of Funding

This work has been performed within the "Metrology for Artificial Intelligence in Medicine (M4AIM)" programme funded by the German Federal Ministry for Economy and Climate Action (BMWK) in the frame of the QI-Digital Initiative, and received further support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 758985). Niklas Giesa is funded by the German Academic Scholarship Foundation.

References

Ahmed, N., Natarajan, T., Rao, K., 1974. Discrete Cosine Transform. IEEE Transactions on Computers C-23, 90–93. doi:10.1109/T-C.1974.223784.

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A Next-generation Hyperparameter Optimization Framework, in: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (Eds.), Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, ACM. pp. 2623–2631. doi:10.1145/3292500.3330701.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein Generative Adversarial Networks, in: Precup, D., Teh, Y.W. (Eds.), Proceedings of the 34th International Conference on Machine Learning, PMLR. pp. 214–223.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for Hyper-Parameter Optimization, in: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F.C.N., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain, pp. 2546–2554.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J., 2017. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks, in: Doshi-Velez, F., Fackler, J., Kale, D., Ranganath, R., Wallace, B., Wiens, J. (Eds.), Proceedings of the 2nd Machine Learning for Healthcare Conference, PMLR. pp. 286–305.

Davis, J., Goadrich, M., 2006. The Relationship between Precision-Recall and ROC Curves, in: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874. doi:10.1016/j.patrec.2005.10.010.

Giesa, N., Haufe, S., Menk, M., Weiß, B., Spies, C.D., Piper, S.K., Balzer, F., Boie, S.D., 2024. Predicting postoperative delirium assessed by the Nursing Screening Delirium Scale in the recovery room for non-cardiac surgeries without craniotomy: A retrospective study using a machine learning approach. PLOS Digital Health 3, e0000414. doi:10.1371/journal.pdig.0000414.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets, in: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C., 2017. Improved Training of Wasserstein GANs, in: Guyon, I., Luxburg, U.v., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 5767–5777.

Harutyunyan, H., Khachatrian, H., Kale, D.C., Ver Steeg, G., Galstyan, A., 2019. Multitask learning and benchmarking with clinical time series data. Scientific Data 6, 96. doi:10.1038/s41597-019-0103-9.

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Johnson, A.E.W., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T.J., Hao, S., Moody, B., Gow, B., Lehman, L.w.H., Celi, L.A., Mark, R.G., 2023. MIMIC-IV, a freely accessible electronic health record dataset. Scientific Data 10, 1. doi:10.1038/s41597-022-01899-x.

Kaissis, G.A., Makowski, M.R., Rückert, D., Braren, R.F., 2020. Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence 2, 305–311. doi:10.1038/s42256-020-0186-1.

Kim, B.N., Dolz, J., Jodoin, P.M., Desrosiers, C., 2021. Privacy-Net: An Adversarial Approach for Identity-Obfuscated Segmentation of Medical Images. IEEE Transactions on Medical Imaging 40, 1737–1749. doi:10.1109/TMI.2021.3065727.

Kingma, D.P., Welling, M., 2019. An Introduction to Variational Autoencoders. Foundations and Trends® in Machine Learning 12, 307–392. doi:10.1561/2200000056.

Li, J., Cairns, B.J., Li, J., Zhu, T., 2023. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. npj Digital Medicine 6, 98. doi:10.1038/s41746-023-00834-7.

Li, L., Jamieson, K.G., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2017. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. Journal of Machine Learning Research (JMLR) 18, 185:1–185:52.

Li, N., Li, T., Venkatasubramanian, S., 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, in: 2007 IEEE 23rd International Conference on Data Engineering, IEEE, Istanbul. pp. 106–115. doi:10.1109/ICDE.2007.367856.

Lichtner, G., Balzer, F., Haufe, S., Giesa, N., Schiefenhövel, F., Schmieding, M., Jurth, C., Kopp, W., Akalin, A., Schaller, S.J., Weber-Carstens, S., Spies, C., von Dincklage, F., 2021. Predicting lethal courses in critically ill COVID-19 patients using a machine learning model trained on patients with non-COVID-19 viral pneumonia. Scientific Reports 11, 13205. doi:10.1038/s41598-021-92475-7. tex.web_url_date: 2022-09-13.

Lin, Z., Jain, A., Wang, C., Fanti, G., Sekar, V., 2020. Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions, in: Proceedings of the ACM Internet Measurement Conference, ACM, Virtual Event USA. pp. 464–483. doi:10.1145/3419394.3423643.

Lipp, A., Vermeesch, P., 2023. Short communication: The Wasserstein distance as a dissimilarity metric for comparing detrital age spectra and other geological distributions. Geochronology 5, 263–270. doi:10.5194/gchron-5-263-2023.

Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M., 2007. L -diversity: Privacy beyond k -anonymity. ACM Transactions on Knowledge Discovery from Data 1, 3. doi:10.1145/1217299.1217302.

Maglio, S., Park, C., Tognarelli, S., Menciassi, A., Roche, E.T., 2021. High-Fidelity Physical Organ Simulators: From Artificial to Bio-Hybrid Solutions. IEEE Transactions on Medical Robotics and Bionics 3, 349–361. doi:10.1109/TMRB.2021.3063808.

Meyer, A., Zverinski, D., Pfahringer, B., Kempfert, J., Kuehne, T., Sündermann, S.H., Stamm, C., Hofmann, T., Falk, V., Eickhoff, C., 2018. Machine learning for real-time prediction of complications in critical care: a retrospective study. The Lancet Respiratory Medicine 6, 905–914. doi:10.1016/S2213-2600(18)30300-X.

Myles, P., Ordish, J., Tucker, A., 2023. The potential synergies between synthetic data and in silico trials in relation to generating representative virtual population cohorts. Progress in Biomedical Engineering 5, 013001. doi:10.1088/2516-1091/acafbf.

Nistal-Nuño, B., 2022. Developing machine learning models for prediction of mortality in the medical intensive care unit. Computer Methods and Programs in Biomedicine 216, 106663. doi:10.1016/j.cmpb.2022.106663.

Ramdas, A., Trillos, N., Cuturi, M., 2017. On Wasserstein Two-Sample Testing and Related Families of Nonparametric Tests. Entropy 19, 47. doi:10.3390/e19020047.

Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60, 259–268. doi:10.1016/0167-2789(92)90242-F.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved Techniques for Training GANs, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA. pp. 2234–2242. Event-place: Barcelona, Spain.

Samarati, P., Sweeney, L., 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.

Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics, in: Proceedings of the 32nd international conference on international conference on machine learning - volume 37, JMLR.org. pp. 2256–2265. Place: Lille, France Number of pages: 10.

Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A., Miklau, G., 2021. Benchmarking Differentially Private Synthetic Data Generation Algorithms. CoRR abs/2112.09238. arXiv:2112.09238.

Tomašev, N., Glorot, X., Rae, J.W., Zielinski, M., Askham, H., Saraiva, A., Mottram, A., Meyer, C., Ravuri, S., Protsyuk, I., Connell, A., Hughes, C.O., Karthikesalingam, A., Cornebise, J., Montgomery, H., Rees, G., Laing, C., Baker, C.R., Peterson, K., Reeves, R., Hassabis, D., King, D., Suleyman, M., Back, T., Nielson, C., Ledsam, J.R., Mohamed, S., 2019. A clinically applicable approach to continuous prediction of future acute kidney injury. Nature 572, 116–119. doi:10.1038/s41586-019-1390-1.

Vincent, J.L., Moreno, R., Takala, J., Willatts, S., De Mendonça, A., Bruining, H., Reinhart, C.K., Suter, P.M., Thijs, L.G., 1996. The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure: On behalf of the Working Group on Sepsis-Related Problems of the European Society of Intensive Care Medicine (see contributors to the project in the appendix). Intensive Care Medicine 22, 707–710. doi:10.1007/BF01709751.

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., McLachlan, S., 2018. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association 25, 230–238.

World Health Organization (WHO), 2022. International Classification of Diseases, Eleventh Revision (ICD-11). Publisher:.

Yale, A., Dash, S., Dutta, R., Guyon, I., Pavao, A., Bennett, K.P., 2020. Generation and evaluation of privacy preserving synthetic health data. Neurocomputing 416, 244–255. doi:10.1016/j.neucom.2019.12.136.

A Existing approaches for synthetic EHR data generation

With Synthea (Walonoski et al., 2018) being a notable exception, most state-of-the art methods for EHR generation are based on generative ML modeling. To our knowledge, none of the proposed methods is, however, capable of generating combined static and time series patient data, leaving DoppelGANger as the only possible comparison for GHOSTS.

Synthea Synthea (Walonoski et al., 2018) is a framework for the generation of "synthetic patients". Synthea collects summary statistics of real-world patients, such as means and percentiles, and combines them with hand-crafted rules encoded in state machines to generate synthetic static patient data. By using only aggregate summary statistics, it aims to be private by construction. This approach ensures that the generated data are somewhat consistent with the underlying data distribution at the level of individual features. However, complex distributions, involving, for example, multiple modes or higher-order interactions between features are not adequately modeled, limiting the realism of the data generated by Synthea. In particular, the generating of non-prototypical patient trajectories requires non-trivial effort, thus limiting Synthea's usability.

medGAN MedGAN (Choi et al., 2017) is a GAN-based approach that uses a combination of a GAN and an autoencoder to generate discrete features such as diagnosis or medication/treatment codes. medGAN can also generate timing information for such features but is unable to generate complete time series for continuous-valued quantities such as vital signs, limiting its applicability to generate heterogeneous ICU data.

HealthGAN An improvement upon medGAN is HealthGAN (Yale et al., 2020). It uses the state-of-the-art WGAN architecture (Gulrajani et al., 2017) to allow the generation of continuous and discrete data. However, as both the generator and the discriminator are consist of MLPs with fully-connected layers, HealthGAN is not suitable to model longitudinal time-series data.

EHR-M-GAN EHR-M-GAN (Li et al., 2023) is able to generate both continuous- and discrete-valued longitudinal data. Just as MedGAN, it combines the GAN approach with a variational autoencoder while also incorporating ideas from self-supervised learning to extract common representations from correlated data of mixed-types. In addition, the generator of the GAN uses a bilateral LSTM to accommodate additional information provided by the autoencoder. The limitation of EHR-M-GAN is, however, that is unable to generate static attributes like sex and age.

B Faithfulness metrics

B.1 Similarity of raw feature values

Let $eCDF_f(\mathcal{D}^{\text{MIMIC}}_{\text{train}})$ and $eCDF_f(\mathcal{D}^{\text{generated}}_{\text{train}})$ be eCDFs of one of the features $f \in F$ in real and generated data respectively, where $eCDF$ is a step function that jumps up by 1/n at each of the n data points, then we define Wasserstein distance between them as:

$$WSD_f = W_1(eCDF_f(\mathcal{D}^{\text{MIMIC}}_{\text{train}}), eCDF_f(\mathcal{D}^{\text{generated}}_{\text{train}})), \text{ where } \quad W_1(X,Y) = \int_{\theta\in\Theta} \|X - Y\| d\theta \ , \tag{3}$$

where X and Y are cumulative distribution functions.

The Wasserstein distance can be normalized to be between 0 and 1 as follows:

$$WSD_{\text{norm}} = \frac{1}{|F|} \sum_{f\in F} \frac{W_1(eCDF_f(\mathcal{D}^{\text{MIMIC}}_{\text{train}}), eCDF_F(\mathcal{D}^{\text{generated}}_{\text{train}}))}{\max(\mathcal{D}^{\text{MIMIC}}_{\text{train}}, \mathcal{D}^{\text{generated}}_{\text{train}}) - \min(\mathcal{D}^{\text{MIMIC}}_{\text{train}}, \mathcal{D}^{\text{generated}}_{\text{train}})} \ , \tag{4}$$

B.2 Similarity of second-order statistics

The pairwise Pearson correlation coefficient (PCC) is defined as

$$r_{i,j} = C_{i,j} / \sqrt{C_{i,i} * C_{j,j}} \ , \tag{5}$$

where

$$C_{i,j} = \frac{1}{N-1} \sum_{t=1}^{|T|} (d_{i,t} - \overline{d_i})(d_{j,t} - \overline{d_j}), \text{ where } \overline{d_i} = \frac{1}{|T|} \sum_{t=1}^{|T|} d_{i,t} \tag{6}$$

is covariance.$d_{i,j}$ is an entry in matrix $d_{\text{time}} = (d_{f,t})$, where $f \in F$ and $t \in T$. PCC is computed between all pairs of features and every sample within a dataset: $PCC_n = \{r_{i,j} | i, j \in F\}$ for $n \in N$.

Sample-wise covariances are then aggregated to obtain mean and standard deviations across all samples:

$$M = \frac{1}{N}\sum_{k=1}^{N} PCC_n, \quad \sigma = \sqrt{\frac{\sum_{k=1}^{N}(PCC_n - \mu)^2}{N-1}} \tag{7}$$

Let K be the set of all pairs (i,j) of elements of F, then correlation accuracy (CorrAcc, Tao et al., 2021) Next, mean PCC for each feature pair $\mu_{i,j} = \frac{1}{N}\sum_{n=1}^{N} r_{i,j,n}$, where $(i,j) \in K$, is discretized into seven bins: low, medium, and high positive correlation ($\mu_j \in [.1,.3)$, $\mu \in [.3,.5)$, $\mu \in [.5,1]$, respectively), low, medium, and high negative correlation (defined analogously), and no correlation ($\mu \in (-.1,.1)$).

$$f_{i,j} = \begin{cases} positive\ low & \text{if } \mu_{i,j} \in [.1,.3] \\ positive\ medium & \text{if } \mu_{i,j} \in [.3,.5] \\ \dots \end{cases}$$

$$CorrAcc = \frac{1}{N}\sum_{(i,j)\in K} 1(f_{i,j}^{\text{train}} = f_{i,j}^{\text{generated}}) \tag{8}$$

To obtain metrics that are more sensitive to minor deviations in PCC, we also compute the mean absolute difference (CorrMAD) and the mean squared error (CorrMSE) between PCCs observed in training and generated data:

$$\text{CorrMAD} = \frac{1}{N}\sum_{(i,j)\in K} \left|\mu_{ij}^{train} - \mu_{ij}^{generated}\right| \tag{9}$$

$$\text{CorrMSE} = \frac{1}{N}\sum_{(i,j)\in K} \left(\mu_{ij}^{train} - \mu_{ij}^{generated}\right)^2 . \tag{10}$$

We define the autocorrelation of the time series $d_{i,f,t}$ of feature f, sample i and at lag h as

$$r_{i,f,h} = C_{i,f,h}/C_{i,f,0} \ , \tag{11}$$

where $C_{i,f,h}$ is the autocovariance

$$C_{i,f,h} = \frac{1}{\|T\|}\sum_{t=1}^{N-h-1} (d_{i,f,t} - \overline{d_{i,f}})(d_{i,f,t+h} - \overline{d_{i,f}}) \ . \tag{12}$$

Let the autocorrelation function be $P_{\text{auto}}(h) = \{r_{i,f,h} : i \in N, f \in F\}$, then the distribution of PCC is $P_{\text{auto}}^{\text{train}} = P_{\text{auto}}(h)$ for $h \in H, |H| \leq |T|$, where H is a sequence of lag values, applied to the real data. $P_{\text{auto}}^{\text{generated}}$ is defined analogously for the generated data. Autocorrelation divergence is defined as

$$\text{AD} = WSD(P_{\text{auto}}^{\text{train}}, P_{\text{auto}}^{\text{generated}}) \ . \tag{13}$$

Let the cross-correlation function be a $P_{\text{cross}}(h) = \{r_{i,f_j,f_k,h} : i \in N, f_j, f_k \in F\}$, then the distribution of PCC is $P_{\text{cross}}^{\text{train}} = P_{\text{cross}}(h)$ for $h \in H, |H| \leq |T|$, where H is a sequence of lag values, applied to the real data.

$$r_{i,f_j,f_k,h} = C_{i,f_j,f_k,h}/C_{i,f_j,f_k,0}, \text{ where } C_{i,f_j,f_k,j} \text{ is a covariance} \tag{14}$$

$$C_{i,f_j,f_k,h} = \frac{1}{\|T\|}\sum_{t=1}^{N-h-1} (d_{i,f_j,t} - \overline{d_{i,f_j}})(d_{i,f_k,t+h} - \overline{d_{i,f_k}})$$

Then, cross-correlation divergence is

$$\text{CCD} = WSD(P_{\text{cross}}^{\text{train}}, P_{\text{cross}}^{\text{generated}}) \tag{15}$$

C Utility metrics

Two different deep neural network classifiers C are used to distinguish low from high SOFA scores for utility assessment. Both take time series data $\mathbf{d}_{\text{time}}$ as input and output a continuous value $C(\mathbf{d}_{\text{time}}) \in \mathbb{R}$ that can be intepreted as the probability of the patient from which $\mathbf{d}_{\text{time}}$ was recorded has a high SOFA score. Prior to training, all features are scaled to values between 0 and 1 using a maximum absolute scaling scheme.

An LSTM-based architecture is used to classify based on continuous time series data only. In addition, the LSTM is combined with an MLP acting on summary statistics from the same time series (LSTM-MLP). The LSTM network consists of two parts: an LSTM layer with dropout, and a sequence of dense layers. First, the input is passed to the LSTM layers. Then, the output features of the last time step from the last layer of the LSTM layers are passed to the dense layers after which the logistic loss is applied. The numbers of nodes of the LSTM and dense layers, the numbers of LSTM and dense layers, and the dropout rate are hyperparameters that are optimized.

To select the hyperparameters of the method, we conducted hyperparameter optimization (HO) using the Optuna library (Akiba et al., 2019) in combination with tree-structured Parzen estimators (Bergstra et al., 2011), hyperband pruning (Li et al., 2017), and a reduction factor of 2. We reserve 20% of the training data as a validation split for HO. The optimization was conducted until the first 200 samples were generated, using the average precision of the classification of the validation set as the performance metric. Confidence intervals for the validation performance were obtained using bootstrapping (K=1,000).

The LSTM-MLP classifier uses the same LSTM architecture in combination with an MLP network. The MLP takes descriptive summary statistics of the time series data as parallel input features. Concretely, we use the 20^{th}, 50^{th}, and 90^{th} percentiles of the time series as features for the MLP. The MLP consists of a series of dense layers, whose number and width are hyperparameters. The outputs of the last dense layers of the LSTM and MLP subnetworks are concatenated and passed to a final linear layers before the logistic activation function is applied.

The performance of the classification task is evaluated using the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC). The ROC curve depicts the TPR as a function of the FPR over all possible classifier thresholds (Fawcett, 2006). Analogously, the precision-recall curve illustrates the trade-off between precision and TPR (Davis and Goadrich, 2006). The AUROC is computed by numerically integrating the ROC curve, aggregating the trade-off between TPR and FPR in a single number, while, analogously, AUPRC is computed as the integral of the precision-recall curve:

$$\text{AUROC} = \sum_n (FPR_n - FPR_{n-1}) TPR_n$$
$$\text{AUPRC} = \sum_n (TPR_n - TPR_{n-1}) P_n \ ,$$

where FPR_n, TPR_n, and P_n are FPR, TPR, and precision at the n^{th} threshold, and where the number of thresholds is equal or less than the number of samples. For binary classification tasks, the performance of a random classifier is characterized by AUROC $= 0.5$ and AUPRC $= {}^{1}/_{N^+}$, where N^+ is the number of positive test samples.

D Additional results

Table 2: Performance of the MLP-LSTM model in the downstream data utility task. Performance was assessed using two performance scores: area under precision-recall curve (AUPRC), calculated as average precision, and area under receiver operator curve (AUROC). Table shows performance of MLP-LSTM classifier in the data utility task, using two metrics: *Train on Synthetic, Test on Real*, and *Train on Real, Test on Synthetic*. *Train on Synthetic, Test on Real*: classifier was trained on synthetic data generated by respective method, and then tested on the baseline data. *Train on Real, Test on Synthetic*: classifier was trained on the baseline data, and then tested on the synthetic data generated by respective method.

	AUROC		AUPRC	
	Train on real, test on synth.	Train on synth., test on real	Train on real, test on synth.	Train on synth., test on real
Real	0.789 ± 0.014	**0.789 ± 0.014**	0.793 ± 0.018	**0.793 ± 0.018**
DoppelGANger	0.743 ± 0.015	0.536 ± 0.018	0.735 ± 0.021	0.577 ± 0.022
GHOSTS_POST	0.801 ± 0.014	0.674 ± 0.017	**0.799 ± 0.020**	0.701 ± 0.020
GHOSTS_RAW	**0.802 ± 0.014**	0.695 ± 0.017	**0.799 ± 0.020**	0.717 ± 0.020

Table 3: Performance of the LSTM model in the downstream data utility task. Performance was assessed using two performance scores: area under precision-recall curve (AUPRC), calculated as average precision, and area under receiver operator curve (AUROC). Table shows performance of LSTM classifier in the data utility task, using two metrics: *Train on Synthetic, Test on Real*, and *Train on Real, Test on Synthetic*. *Train on Synthetic, Test on Real*: classifier was trained on synthetic data generated by the respective method, and then tested on the baseline data. *Train on Real, Test on Synthetic*: classifier was trained on the baseline data, and then tested on the synthetic data generated by the respective method.

	AUROC		AUPRC	
	Train on real, test on synth.	Train on synth., test on real	Train on real, test on synth.	Train on synth., test on real
Real	0.620 ± 0.018	0.620 ± 0.018	**0.647 ± 0.023**	**0.647 ± 0.023**
DoppelGANger	0.562 ± 0.019	0.564 ± 0.018	0.589 ± 0.024	0.605 ± 0.022
GHOSTS_POST	0.625 ± 0.017	0.626 ± 0.017	0.627 ± 0.023	0.631 ± 0.023
GHOSTS_RAW	**0.637 ± 0.017**	**0.662 ± 0.016**	0.634 ± 0.023	0.635 ± 0.023

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The authors confirm that both abstract and introduction contain claims that accurately reflect papers contribution. Introduction contains enumeration of specific contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4 contains a detailed discussion of study limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not contain theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main sections of the paper as well as appendices contain the information necessary for reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The authors are planning to release the generated dataset and the code in the future work, but right now the authors are not ready to provide it yet.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The details about training and testing are specified in Section 3 and Section 2 and in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide error bars on all figures, as well as a standard deviation for the tables. All error bars and std were computed with statistical bootstrap method as indicated in text.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3 provides information on the hardware used as well as time of execution.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Authors have read the Code of Ethics and confirm that the research conducted conforms with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The authors have discussed the impace in the Section 4.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not release the data and models. In the future, such artifacts will be released under data usage agreements on the Physionet platform.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Authors describe, cite and mention the terms of use in Subsection 2.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The authors do not release new assets in this submission.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper uses de-identified data. Given the de-identified nature of the data, the Beth Israel Deaconess Medical Center's ethical committee waived the requirement for informed consent (Johnson et al., 2023).

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper uses de-identified data. Given the de-identified nature of the data, the Beth Israel Deaconess Medical Center's ethical committee waived the requirement for informed consent (Johnson et al., 2023)

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.