IGAMT: Privacy-Preserving Electronic Health Record Synthesization with Heterogeneity and Irregularity

Wenjie Wang*[1] Pengfei Tang[3], Jian Lou*[2], Yuanming Shao[1], Lance Waller[3], Yi-an Ko[3], Li Xiong[3]

[1] ShanghaiTech University
[2] ZJU-Hangzhou Global Scientific and Technological Innovation Center
[3] Emory University
wangwj1@shanghaitech.edu.cn, pengfei.tang@microsoft.com, jian.lou@zju.edu.cn, shaoym1@shanghaitech.edu.cn, lwaller@emory.edu, yi-an.ko@emory.edu, lxiong@emory.edu

Abstract

Utilizing electronic health records (EHR) for machine learning-driven clinical research has great potential to enhance outcome predictions and treatment personalization. Nonetheless, due to privacy and security concerns, the secondary use of EHR data is regulated, constraining researchers' access to EHR data. Generating synthetic EHR data with deep learning methods is a viable and promising approach to mitigate privacy concerns, offering not only a supplementary resource for downstream applications but also sidestepping the privacy risks associated with real patient data. While prior efforts have concentrated on EHR data synthesis, significant challenges persist: addressing the heterogeneity of features including temporal and non-temporal features, structurally missing values, and irregularity of the temporal measures, and ensuring rigorous privacy of the real data used for model training. Existing works in this domain only focused on solving one or two aforementioned challenges. In this work, we propose *IGAMT*, an innovative framework to generate privacy-preserved synthetic EHR data that not only maintains high quality with heterogeneous features, missing values, and irregular measures but also achieves differential privacy with enhanced privacy-utility trade-off. Extensive experiments prove that *IGAMT* significantly outperforms baseline and state-of-the-art models in terms of resemblance to real data and performance of downstream applications. Ablation studies also prove the effectiveness of the techniques applied in *IGAMT*.

1 Introduction

The availability of electronic health records (EHR) not only improves patient care but also boosts the advancement of medical research. However, due to privacy and security concerns, secondary use of EHR data for research purposes is always regulated, thus constraining researchers' access to EHR data (Choi et al. 2017).

A practical and promising solution to mitigate the privacy concern is to generate synthetic EHR data that are realistic for machine learning tasks, offering not only a supplementary resource for downstream applications but also avoiding the privacy risks associated with real patient data. To achieve this goal, synthetic EHR data need to retain the sophisticated characteristics of the real data because these attributes can substantially impact the usage of the synthetic data in the downstream tasks. The specific characteristics of EHR data are shown in Figure 1 and summarized below:

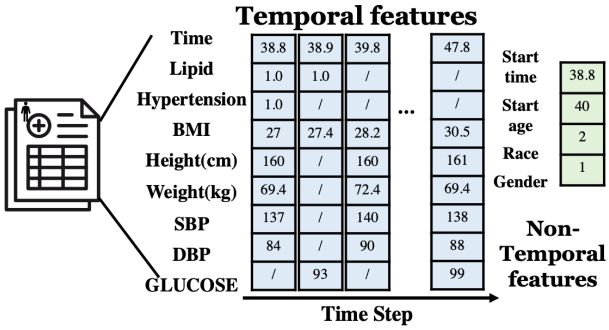


Figure 1: Illustration of EHR raw data

1. Heterogeneity of features: Each record has both temporal and non-temporal features. Some features are time-related (blue blocks) such as heart rate, which will be recorded at each visit. For temporal features, each record can be viewed as a matrix consisting of multiple visits (time steps) and each visit contains multi-dimensional features (Shickel et al. 2017). Some features are non-temporal (not related to time) such as demographic features including gender and race (green blocks).
2. Missing values: EHRs may contain structurally missing data that correspond to specific clinical scenarios. That is, certain events or measurements are intentionally omitted or not recorded by clinicians. For example, additional tests (e.g., glucose levels) will not be measured (i.e., become missing values) if the patient's vital signs (e.g., blood pressure) are normal during a clinical visit. The illustration of such missing value is represented as "/" in Figure 1) (Bang, Wang, and Yang 2020).
3. Irregularity of features: The temporal features may be measured in different frequencies, for instance, some features are measured on an hourly time scale while others are on a monthly time scale (Shickel et al. 2017; Bang, Wang, and Yang 2020).

*Corresponding Authors

Capturing these sophisticated characteristics of heterogeneous features, missing values, and irregular measures poses challenges to deep learning models. Besides the challenge in representation learning of these characteristics, another challenge lies in crafting synthetic EHR data that retains these characteristics. Most of the existing works focused on isolated aspects of these characteristics, resulting in synthesized EHR data that cannot fulfill the downstream requirements. For instance, some works (Neil, Pfeiffer, and Liu 2016) only concentrated on the representation learning of irregular measures while disregarding the impact of missing values, which can lead to completely opposite diagnosis. The detailed related works about EHR representation learning and synthesizing are discussed in Section 2.

Privacy leakage is another major challenge for models built on sensitive data like EHR. Synthetic data is typically generated by a deep generative model trained on real data, therefore when the model and synthetic data are published, the original data can be still inferred and incurs privacy leakage (Rahman et al. 2018). To prevent this issue, differential privacy (DP), a formal mathematical privacy-preserving framework, is widely applied in the model training stage (Beaulieu-Jones et al. 2019; Lee et al. 2020). One limitation of the state-of-the-art DP techniques, like gradient perturbation (Abadi et al. 2016), is that they can undermine the utility of the model because of the randomization introduced in the model. Therefore, how to mitigate utility degradation and balance the trade-off between utility and privacy is a major challenge. Existing works on EHR data synthesization can neither maintain all the special characteristics nor provide a formal privacy guarantee to the training data (Choi et al. 2017; Beaulieu-Jones et al. 2019; Lee et al. 2020; Baowaly et al. 2019; Chin-Cheong, Sutter, and Vogt 2020; **?**).

Contributions. In this work, we propose the Imitative Generative Adversarial Mixed-embedding Transformer (*IGAMT*) to generate differentially private EHR with sophisticated characteristics. As shown in Figure 3a, the architecture of *IGAMT* contains three generative adversarial networks (GANs) (Goodfellow et al. 2014) and an autoencoder (Hinton and Salakhutdinov 2006). *IGAMT* leverages transformer (Vaswani et al. 2017) to capture both temporal and non-temporal features. In addition, we utilize masks and time embedding to capture missing values and irregular measures and combine sequence-to-sequence autoencoder with transformer and GAN to better maintain the sophisticated characteristics. We further adopt a new structure, *Imitator*, to reduce the randomization required by the DP technique while keeping the complex architecture for enhanced privacy and utility trade-off.

IGAMT is the first framework to generate differentially private EHR data of high quality with heterogeneous features, missing values, and irregular measures. Our key contributions are listed as follows:

1. We propose an EHR data generative model that not only maintains the specific characteristics of EHR but also provides a differential privacy (DP) guarantee.
2. We leverage sequence-to-sequence transformer with missing value masks, time embedding, and non-temporal embedding in our generative model to learn the sophisticated characteristics of EHR and generate synthetic data.
3. We incorporate a novel *Imitator* in our architecture to imitate the behaviors of the decoder. Applying gradient perturbation to the *Imitator* rather than the decoder itself improves the model utility (quality of the synthetic EHR) while preserving the same level of DP.
4. Extensive experiments on real-world EHR data demonstrate that *IGAMT* significantly outperforms baseline and state-of-the-art models in terms of resemblance of the synthetic data to real data and performance of downstream applications and achieves enhanced privacy utility trade-off.

2 Related Work

In this section, we briefly introduce the existing work on EHR data representation learning and synthesization, and the differential privacy techniques, especially the applications in generative models.

EHR representation learning. Several works focused on representation learning of EHR data by building specific neural networks to capture these characteristics. Neil, Pfeiffer, and Liu (2016) proposed a novel recurrent network, Phased-LSTM, to capture irregular measures of temporal data, and Bang, Wang, and Yang (2020) further improved Phased-LSTM to fit missing values and irregular measures.

EHR data synthesization. For EHR synthesization, Choi et al. (2017) proposed *medGAN* to generate multi-label discrete records. However, *medGAN* only works on discrete features and does not address the potential privacy leakage. Hyland, Esteban, and Rätsch (2018) proposed recurrent conditional GAN (*RCGAN*), which can generate temporal medical features. However, *RCGAN* does not take non-temporal features, missing values and privacy protection into consideration. Xu et al. (2019) built *CTGAN* for tabular medical data, but cannot be directly applied to EHR data. Baowaly et al. (2019) introduced *medWGAN* and *medBGAN* on top of *medGAN* by replacing GAN with more powerful variants, WGAN (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) and boundary-seeking GAN (BGAN) (Hjelm et al. 2017). However, they did not take temporal features and privacy preservation into consideration.

Differential privacy. *Differential Privacy (DP)* (Dwork 2011; Dwork et al. 2006; Dwork, Roth et al. 2014) is a theoretical privacy framework for aggregate data analysis, which ensures the output of a randomized algorithm is indistinguishable between two neighboring datasets that differ in one record (or bounded by a distance metric) with a certain probability. *Gradient perturbation* is a common practice to achieve DP for deep learning models by injecting perturbation into the gradient of each parameter (Song, Chaudhuri, and Sarwate 2013; Bassily, Smith, and Thakurta 2014; Abadi et al. 2016; Wang, Ye, and Xu 2017; Lee and Kifer 2018; Yu et al. 2019; Wang et al. 2021).

Privacy-preserving generative model for EHR. To obtain a privacy-preserving generative model for EHR data, Beaulieu (Beaulieu-Jones et al. 2019) applied DP into the

training process of the discriminator of *AC-GAN* (Odena, Olah, and Shlens 2017). However, this work does not take the temporal features and missing values into consideration. Chin-Cheong, Sutter, and Vogt (2020) proposed a DP-GAN to generate heterogeneous EHRs with non-temporal features and missing values. However, temporal features are still missed in this work. Lee et al. (2020) proposed a dual adversarial autoencoder (*DAAE*) to generate temporal EHR and employ DP during training to prevent privacy leakage. *DAAE* is the existing state-of-the-art generative model with DP for EHR, but is incapable of capturing non-temporal features, missing values, and irregular measures. We will use *DAAE* as the baseline comparison to demonstrate the effectiveness of *IGAMT*.

3 Preliminaries

Differential Privacy

Differential Privacy (DP) ensures that the output of a randomized algorithm is indistinguishable between two neighboring datasets that differ in one record (or bounded by a distance metric) with a certain probability.

Definition 1. *(ϵ, δ)-Differential Privacy) A randomized mechanism $\mathcal{M} : \mathbf{D} \to \mathbf{R}$ with domain $\mathbf{D}$ and range $\mathbf{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent input datasets $\mathcal{D}, \mathcal{D}' \in \mathbf{D}$ and for any subset of outputs $\mathbf{S} \subseteq \mathbf{R}$ it holds that*

$$Pr(\mathcal{M}(\mathcal{D}) \in \mathbf{S}) \leq e^{\epsilon} Pr(\mathcal{M}(\mathcal{D}') \in \mathbf{S}) + \delta,$$

where ϵ denotes the privacy level (or privacy budget) and δ denotes the probability that the inequality breaks.

The lower the ϵ, the stronger the privacy. The common approach to achieving (ϵ, δ)-DP is the Gaussian mechanism that adds calibrated Gaussian noise to the output.

Gradient perturbation. The most commonly used approach to achieve differential privacy in deep learning systems is gradient perturbation. It injects calibrated noise into the gradient during training with the following objective function and gradient update.

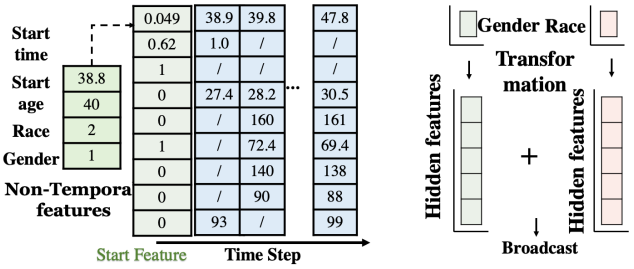
$$\mathcal{J}(\theta_t) = \frac{1}{N} \sum_{i=1}^{N} l(\mathbf{x}_i, y_i, \theta_t), \quad \theta_{t+1} = \theta_t - \eta(\nabla \mathcal{J}(\theta_t) + \mathbf{p})$$

where θ_t denotes the parameter at training step t, $\nabla \mathcal{J}(\theta_t)$ denotes the gradient which is bounded by a clipping norm or constrained by Lipschitz continuity of loss function l, and $\mathbf{p}$ denotes the gradient perturbation typically as a Gaussian noise $\mathcal{N}(0, \sigma^2)$.

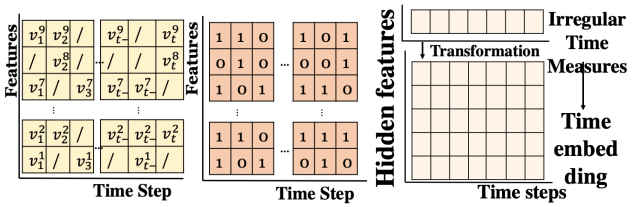
Moment accountant is commonly used to quantify the overall privacy cost from multiple iterations of the entire training.

Theorem 1. *Moment Accountant (Abadi et al. 2016). Let $\mathcal{F} : \mathbb{R}^v \to \mathbb{R}^w$ be an w-dimensional model, and its sensitivity $\Delta_{\mathcal{F}} = \max_{\mathcal{D},\mathcal{D}'} \|\mathcal{F}(\mathcal{D}) - \mathcal{F}(\mathcal{D}')\|_2$. Given training batch size B, the total training size N, the number of training steps T, gradient perturbation with Gaussian noise $\mathcal{N}(0, \sigma^2)$, there exists constants c_1 and c_2 for any $\epsilon < c_1(\frac{B}{N})^2 T$, $\mathcal{F}$ is (ϵ, δ)-DP for any $\delta > 0$, if we choose*

$$\sigma \geq c_2 \frac{B/N\sqrt{Tlog(1/\delta)}}{\epsilon}.$$



(a) Non-temporal feature representation learning



(b) Missing values (c) Irregularity

Figure 2: Model representation learning.

Dual Adversarial Autoencoder (*DAAE*)

The architecture Dual Adversarial Autoencoder (*DAAE*) (Lee et al. 2020) combines a recurrent autoencoder with two generative adversarial networks (GANs) (Goodfellow et al. 2014). Two discriminators in GAN can not only distinguish the central hidden state in the autoencoder but also distinguish real data from reconstructed data and synthetic data.

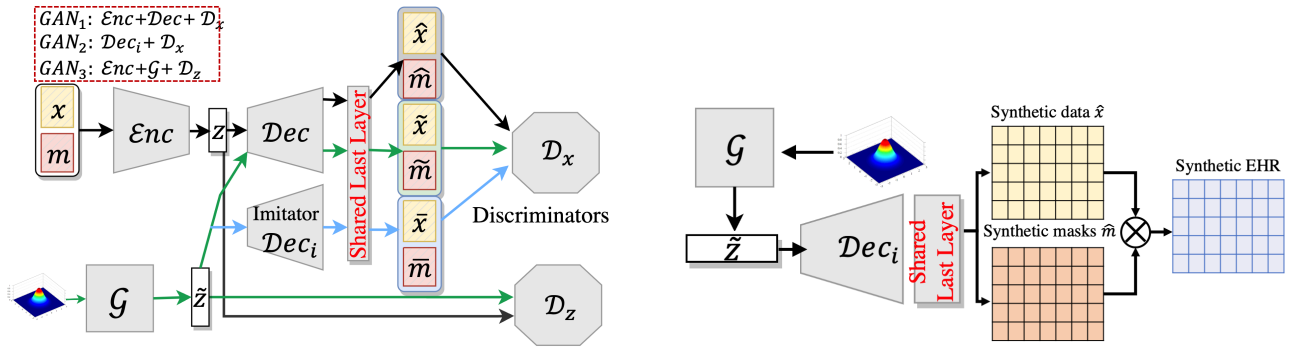
4 *IGAMT*

In this section, we will first present the architecture of *IGAMT* and demonstrate how *IGAMT* solves three challenges: feature representation learning, synthetic EHR generation, and privacy preservation. Then we will introduce the training process of *IGAMT*.

Representation Learning

IGAMT incorporate sequence-to-sequence autoencoder (seq2seq AE) to capture the sophisticated characteristics from data. Transformer (Vaswani et al. 2017) is used to implement both encoder and decoder, which uses self-attention to capture the correlation among features at different time steps. We improve Transformer by incorporating several well-designed techniques to learn sophisticated feature representations of EHR.

Non-temporal features. To simultaneously learn temporal and non-temporal feature representation and capture the connection between these features, non-temporal features are transformed to a vector of the same size as temporal features at each time point which are denoted as the `start feature`, as shown on the left of Figure 2a. In addition, to better learn the non-temporal feature representations, we also transform gender and race into embedding vectors respectively, and broadcast them to all time steps before applying them to hidden states, as shown on the right of Figure 2a.



(a) The training framework of IGAMT.

(b) An illustration of synthesization.

Figure 3: Model architecture of IGAMT training and synthesization.

Missing values. The missing data in EHR have structural patterns and correspond to specific clinical scenarios instead of missing at random. Synthetic EHR data directly generated from deep learning models cannot learn these structural missing values. The models will generate continuous values for each feature and timestep, thus the characteristics of missing values in the real EHR data will be lost in the synthetic EHR. To overcome this challenge and better capture missing values, we create a mask consisting of 1s and 0s to mark the element-wise missing value positions as shown in Figure 2b. Seq2seq AE of *IGAMT* will not only generate synthetic data but also its corresponding synthetic mask. In this way, element-wise multiplication of data and mask will generate the final synthetic EHR data which maintains the missing value characteristics of the real EHR.

Irregular measures. To better capture irregular measures, time steps are extracted from EHRs by calculating the increment of two neighboring time steps and adding 0 as the initial increment. Then we transform the time features into embedding vectors as shown in Figure 2c. These embeddings are then applied to the hidden states during training.

Loss function for representation learning. Seq2seq Transformer AE with specific embeddings in *IGAMT* is leveraged to capture the characteristics related to heterogeneous features, missing values, and irregular measures. This autoencoder takes x and its mask m as the input and generates synthetic data $\hat{x}$ and mask $\hat{m}$, the reconstruction loss is the cross-entropy loss ($\mathcal{CE}$):

$$\mathcal{L}_{rec} = \mathcal{CE}(\mathbf{x}, \hat{\mathbf{x}}) + \mathcal{CE}(m, \hat{m}), \tag{1}$$

where $\mathbf{x}$ denotes element-wise multiplication of x and m, and parameters will be optimized accordingly.

Architectures

Training framework. As shown in Figure 3a, the architecture of *IGAMT* has three modules. First, as explained in the previous section, a seq2seq AE with transformer blocks ($\mathcal{E}nc$ and $\mathcal{D}ec$) is implemented to learn the sophisticated feature representation. This module serves as the generator and, together with the discriminator $\mathcal{D}_x$, constitutes the first GAN (GAN_1). The goal of the discriminator $\mathcal{D}_x$ is to discriminate between the real data and missing value mask (x and m) and the fake ones generated by the seq2seq AE. This GAN is the main part of the *IGAMT* architecture to generate synthetic EHR.

Second, to improve the generative ability of *IGAMT*, we incorporate another GAN (GAN_2) formed by generator $\mathcal{G}$ with $\mathcal{E}nc$ and discriminator $\mathcal{D}_z$ as discussed in *DAAE*. The goal of the discriminator $\mathcal{D}_z$ is to discriminate between "real" hidden states z from encoder $\mathcal{E}nc$ and "fake" states $\tilde{z}$ from the generator $\mathcal{G}$. This module is to improve the model coverage rate and quality of generated sequences by adversarially learning both the continuous latent distribution (z from encoder $\mathcal{E}nc$ and $\tilde{z}$ from generator $\mathcal{G}$) and the data distribution.

Third, the *Imitator* $\mathcal{D}ec_i$ together with the generator $\mathcal{G}$ and the discriminator $\mathcal{D}_x$ constitutes GAN_3. The *Imitator* is incorporated to support differential privacy (DP). Directly applying the DP technique to *IGAMT* without the imitator will bring overwhelmingly large noise to the training process. This occurs because perturbations are required for both generator $\mathcal{D}ec$ and discriminator $\mathcal{D}_x$ as both parts access the real data ($\mathcal{D}_x$ access real data from the forward pass while $\mathcal{D}ec$ from back-propagation). This process will ultimately compromise the model utility and the quality of the synthetic EHRs. We explain below how the imitator is utilized to support DP and analyze the DP in more detail later.

DP guarantee. To reduce the DP randomization and maintain the model utility, we introduce a novel module *Imitator* with the same structure as $\mathcal{D}ec$ to mimic the behavior of the decoder $\mathcal{D}ec$. Compared with $\mathcal{D}ec$, the *Imitator* $\mathcal{D}ec_i$ does not access real data (because it uses $\tilde{z}$ from the generator $\mathcal{G}$), thus only adding gradient perturbation to the discriminator $\mathcal{D}_x$ can ensure DP for $\mathcal{D}ec_i$ (post-processing theorem of DP). Similarly, $\mathcal{G}$ does not access real data, thus only adding gradient perturbation to the discriminator $\mathcal{D}_z$ can ensure DP for $\mathcal{G}$. $\mathcal{G}$ and $\mathcal{D}ec_i$ can be then used to generate synthetic data with DP. Note that the architecture of generators in GAN is always much more complicated than the discriminators which require more gradient perturbation to achieve the same level of DP guarantee. Therefore, incorporating *Imitator* can significantly reduce the DP randomization and improve the model utility. In practice, to better

Algorithm 1: *IGAMT* algorithm

Input: preprocessed training EHRs x and masks m, total training epoch T, gradient perturbation scale σ, learning rate η, batch size B, discriminators update frequency base f_b and frequency hit f_h, gradient clipping norm C

1 $t = 0$;
2 initialize parameters of *IGAMT*;
3 **while** $t < T$ **do**
4 get mini-batch EHRs $x_{(t)}$ and masks $m_{(t)}$;
5 $\mathbf{z}_{(t)} = \mathcal{E}nc(x_{(t)}, m_{(t)})$;
6 $\hat{x}_{(t)}, \hat{m}_{(t)} = \mathcal{D}ec(x_{(t)}, m_{(t)}, \mathbf{z}_{(t)})$;
7 $\tilde{\mathbf{z}}_{(t)} = \mathcal{G}(B)$ (generate synthetic hidden states);
8 sample start features $\mathbf{s}_f$ and craft start masks $\mathbf{s}_m$;
9 $\tilde{x}_{(t)}, \tilde{m}_{(t)} = \mathcal{D}ec(\mathbf{s}_f, \mathbf{s}_m, \tilde{\mathbf{z}}_{(t)})$;
10 $\bar{x}_{(t)}, \bar{m}_{(t)} = \mathcal{D}ec_i(\mathbf{s}_f, \mathbf{s}_m, \tilde{\mathbf{z}}_{(t)})$;
11 **if** $t\%f_b < f_h$ **then**
`// Update` $\mathcal{D}_x$ `with DP perturbation`
12 $\mathcal{L}_{\mathcal{D}_x} = d_{\mathcal{D}_x}(\mathbf{x}, \hat{\mathbf{x}}) + d_{\mathcal{D}_x}(\mathbf{x}, \tilde{\mathbf{x}}) + d_{\mathcal{D}_x}(\mathbf{x}, \bar{\mathbf{x}})$;
13 $grad^{\mathcal{D}_x}_{(t)} = \frac{1}{B}\nabla_{\theta^{\mathcal{D}_x}_{(t)}}\mathcal{L}_{\mathcal{D}_x}$;
14 $grad^{\mathcal{D}_x}_{(t)} = grad^{\mathcal{D}_x}_{(t)}/max(1, \|grad^{\mathcal{D}_x}_{(t)}\|/C)$;
15 $\theta^{\mathcal{D}_x}_{(t+1)} = \theta^{\mathcal{D}_x}_{(t)} - \eta\,(grad^{\mathcal{D}_x}_{(t)} + \mathcal{N}(0, \sigma^2))$;
`// Update` $\mathcal{D}_z$ `with DP perturbation`
16 $\mathcal{L}_{\mathcal{D}_z} = d_{\mathcal{D}_z}(\tilde{\mathbf{z}}, \mathbf{z})$;
17 $grad^{\mathcal{D}_z}_{(t)} = \frac{1}{B}\nabla_{\theta^{\mathcal{D}_z}_{(t)}}\mathcal{L}_{\mathcal{D}_z}$;
18 $grad^{\mathcal{D}_z}_{(t)} = grad^{\mathcal{D}_z}_{(t)}/max(1, \|grad^{\mathcal{D}_z}_{(t)}\|/C)$;
19 $\theta^{\mathcal{D}_z}_{(t+1)} = \theta^{\mathcal{D}_z}_{(t)} - \eta\,(grad^{\mathcal{D}_z}_{(t)} + \mathcal{N}(0, \sigma^2))$;
20 **end**
21 θ^{last}: parameters of the shared last layer between $\mathcal{D}ec$ and $\mathcal{D}ec_i$;
`// Update` $\mathcal{E}nc$
22 $\mathcal{L}_{\mathcal{E}nc} = \mathcal{D}_z(\mathbf{z}) + \mathcal{L}_{rec}$;
23 $\theta^{\mathcal{E}nc}_{(t+1)} = \theta^{\mathcal{E}nc}_{(t)} - \eta\,\frac{1}{B}\nabla_{\theta^{\mathcal{E}nc}_{(t)}}\mathcal{L}_{\mathcal{E}nc}$;
`// Update` $\mathcal{D}ec$ `excluding the last layer`
24 $\mathcal{L}_{\mathcal{D}ec} = -\mathcal{D}_x(\hat{\mathbf{x}}) - \mathcal{D}_x(\tilde{\mathbf{x}}) + \mathcal{L}_{rec}$;
25 $\theta^{Dec}_{(t+1)} = \theta^{Dec}_{(t)} - \eta\,\frac{1}{B}\nabla_{\theta^{Dec}_{(t)}}\mathcal{L}_{\mathcal{D}ec}$;
`// Update` $\mathcal{D}ec_i$`'s last layer with gradient perturbation`
26 $\mathcal{L}_{im} = \text{MSE}(\hat{\mathbf{x}}, \bar{\mathbf{x}}) + \text{MSE}(\tilde{\mathbf{x}}, \bar{\mathbf{x}}) + \text{MSE}(\mathbf{x}, \bar{\mathbf{x}})$;
27 $\mathcal{L}_{\mathcal{D}ec_i} = -\mathcal{D}_x(\bar{\mathbf{x}}) + \mathcal{L}_{im}$;
28 $grad^{last}_{(t)} = \frac{1}{B}\nabla_{\theta^{last}_{(t)}}\mathcal{L}_{\mathcal{D}ec_i}$;
29 $\widehat{grad}^{last}_{(t)} = grad^{last}_{(t)}/max(1, \|grad^{last}_{(t)}\|/C) + \mathcal{N}(0, \sigma^2)$;
30 $\theta^{last}_{(t)} = \theta^{last}_{(t)} - \eta\,(\widehat{grad}^{last}_{(t)})$;
`// Update` $\mathcal{D}ec_i$ `excluding the last layer with chain rule`
31 $grad^{\mathcal{D}ec_i}_{(t)} = \widehat{grad}^{last}_{(t)} * \nabla^{Output(\theta^{\mathcal{D}ec_i}_{(t)})}_{\theta^{\mathcal{D}ec_i}_{(t)}}$;
32 $\theta^{\mathcal{D}ec_i}_{(t+1)} = \theta^{\mathcal{D}ec_i}_{(t)} - \eta\,\frac{1}{B}grad^{\mathcal{D}ec_i}_{(t)}$;
`// Update` $\mathcal{G}$
33 $\mathcal{L}_{\mathcal{G}} = -\mathcal{D}_z(\tilde{\mathbf{z}})$;
34 $\theta^{\mathcal{G}}_{(t+1)} = \theta^{\mathcal{G}}_{(t)} - \eta\,\frac{1}{B}\nabla_{\theta^{\mathcal{G}}_{(t)}}\mathcal{L}_{\mathcal{G}}$;
35 **end**

Output: $\mathcal{D}ec_i$ and $\mathcal{G}$

guide the *Imitator* to mimic $\mathcal{D}ec$, we let these two structures share the same last layer during training ($\mathcal{D}ec$ and $\mathcal{D}ec_i$ have the same architecture) and also utilize an imitation loss for the imitator. We will analyze the DP in detail in the following sections.

Synthesization framework. We explained the training architecture of the *IGAMT* in the above section. After training, we use the DP components of *IGAMT* to generate synthetic EHR, as shown in Figure 3b, which contains G and $\mathcal{D}ec_i$ including the shared last layer of $\mathcal{D}ec$ and $\mathcal{D}ec_i$. The synthesization process can be divided into the following steps: 1) sampling random states from a Gaussian distribution, 2) $\mathcal{G}$ takes random states as the input and generates central hidden states $\tilde{z}$, 3) the *Imitator* $\mathcal{D}ec_i$ takes $\tilde{z}$ as input and generates data and masks, and 4) assemble the generated data and masks to form the synthetic EHR.

The synthetic EHRs generated from *IGAMT* retain the heterogeneous features, missing values, and irregular measures. Moreover, because the generative model is differentially private, these synthetic EHRs are correspondingly privacy-preserved.

Loss Functions and Optimization

In this section, we will first elaborate on each loss function designed to solve each challenge. Then we will present our optimization process and training algorithm (Algorithm 1).

Discriminator $\mathcal{D}_x$. $\mathcal{D}ec$ and $\mathcal{D}ec_i$ are generators in two GANs respectively, sharing the same discriminator $\mathcal{D}_x$. The loss for $\mathcal{D}_x$ consists of the discrimination loss between each synthetic data generated from $\mathcal{D}ec$ and $\mathcal{D}ec_i$ and the real data, which can be stated as:

$$\mathcal{L}_{\mathcal{D}_x} = d_{\mathcal{D}_x}(\mathbf{x}, \hat{\mathbf{x}}) + d_{\mathcal{D}_x}(\mathbf{x}, \tilde{\mathbf{x}}) + d_{\mathcal{D}_x}(\mathbf{x}, \bar{\mathbf{x}}) \tag{2}$$

where $\mathbf{x}$ denotes element-wise multiplication of x and m, $\hat{\mathbf{x}}$, $\tilde{\mathbf{x}}$ and $\bar{\mathbf{x}}$ denotes the generator outputs of the three GANs respectively (illustrated in Figure 3a), and $d_{\mathcal{D}_x}(u, v) = \mathcal{D}_x(v) - \mathcal{D}_x(u)$. The updates of $\mathcal{D}_x$ are using gradient perturbation (Algorithm 1 lines 12-15) to ensure $\mathcal{D}_x$ is DP.

Generator $\mathcal{D}ec$ and Imitator $\mathcal{D}ec_i$. The *Imitator* $\mathcal{D}ec_i$ and $\mathcal{D}ec$ share the same last layer during training. $\mathcal{D}ec$ excluding the last layer is optimized through the back-propagation of associated discrimination loss and the reconstruction loss. The discriminator tries to minimize the discrimination loss while the generator tries to maximize it:

$$\mathcal{L}_{\mathcal{D}ec} = -\mathcal{D}_x(\hat{\mathbf{x}}) - \mathcal{D}_x(\tilde{\mathbf{x}}) + \mathcal{L}_{rec} \tag{3}$$

where $\mathcal{L}_{rec}$ refers to Equation1. $\mathcal{D}ec$ excluding the last layer is updated without gradient perturbation (line 24-25). We note that $\mathcal{D}ec$ except the shared last layer is not DP since the back propagation uses the real data to compute the gradient for updating those layers.

The loss for $\mathcal{D}ec_i$ consists of two parts, the imitation loss and the associated discrimination loss. The goal is to generate $\bar{x}$ and $\bar{m}$ that is close to both real data and the other two sources of synthetic data generated by $\mathcal{D}ec$. The loss in optimizing $\mathcal{D}ec_i$ can be stated as:

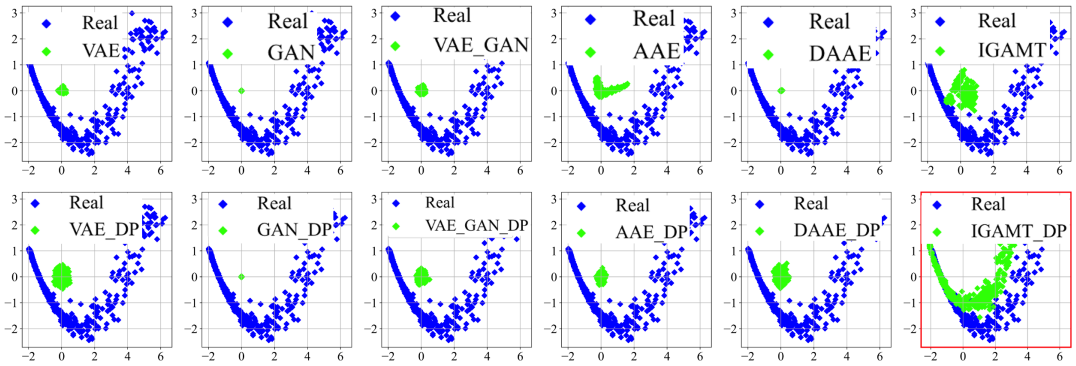


Figure 4: PCA visualization of real and synthetic EHRs on MIMIC-IV-ED

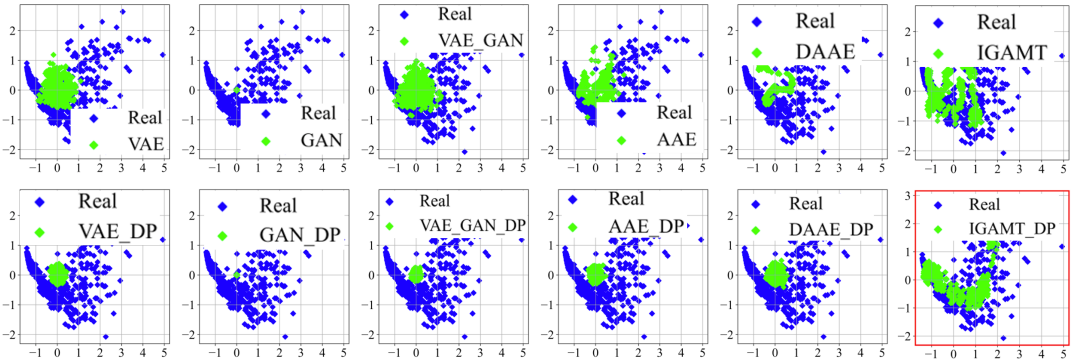


Figure 5: PCA visualization for real and synthetic on Emory Synergy

$$\mathcal{L}_{im} = \text{MSE}(\hat{\mathbf{x}}, \bar{\mathbf{x}}) + \text{MSE}(\tilde{\mathbf{x}}, \bar{\mathbf{x}}) + \text{MSE}(\mathbf{x}, \bar{\mathbf{x}}) \quad (4)$$

$$\mathcal{L}_{\mathcal{D}ec_i} = -\mathcal{D}_x(\bar{\mathbf{x}}) + \mathcal{L}_{im} \quad (5)$$

To update the parameters of $\mathcal{D}ec_i$, we first update the last layer with gradient perturbation (line 28-30). This ensures the last layer is DP. Then, we update the remaining layers of $\mathcal{D}ec_i$ with the chain rule (line 31-32). As the gradient of the last layer is DP, and the gradient computation for the remaining layers does not use real data ($\mathcal{D}ec_i$ is based on $\tilde{z}$ from generator $\mathcal{G}$), so the remaining layers also ensure DP. Hence, the entire *Imitator* $\mathcal{D}ec_i$ including the shared last layer is DP.

Intuitively, while the *Imitator* $\mathcal{D}ec_i$ mimics $\mathcal{D}ec$, this mimicry is limited to a controlled extent, by making both discriminator $\mathcal{D}_x$ and the shared last layer DP. Consequently, the *Imitator* will not memorize the training data the same way as $\mathcal{D}ec$.

Encoder $\mathcal{E}nc$, generator $\mathcal{G}$, and discriminator $\mathcal{D}_z$. The GAN to improve the generative ability of *IGAMT* consists of the encoder $\mathcal{E}nc$, the generator $\mathcal{G}$ and discriminator $\mathcal{D}_z$, where $\mathcal{E}nc$ provides "real" hidden states z, $\mathcal{G}$ synthesizes "fake" states $\tilde{z}$, and $\mathcal{D}_z$ aims to distinguish z from $\tilde{z}$. The loss for the encoder consists of two parts, the reconstruction loss $\mathcal{L}_{rec}$ and the loss from discriminator $\mathcal{D}_z$, which can be stated as :

$$\mathcal{L}_{\mathcal{E}nc} = \mathcal{D}_z(z) + \mathcal{L}_{rec} \quad (6)$$

The loss for discriminator $\mathcal{D}_z$ and generator $\mathcal{G}$ are:

$$\mathcal{L}_{\mathcal{D}_z} = d_{\mathcal{D}_z}(\tilde{z}, \mathbf{z}) \quad (7)$$

$$\mathcal{L}_{\mathcal{G}} = -\mathcal{D}_z(\tilde{z}) \quad (8)$$

$\mathcal{D}_z$ is updated with gradient perturbation (line 16-19) to ensure DP. The updates of $\mathcal{E}nc$ and $\mathcal{G}$ are denoted in line 22-23 and 33-34 respectively. Since $\mathcal{G}$ is updated with the discrimination loss of $\mathcal{D}_z$ which is DP, $\mathcal{G}$ also ensures DP.

Complete algorithm. Algorithm 1 shows the complete training process of *IGAMT*. At the start of training, $\mathcal{E}nc$ and $\mathcal{D}ec$ together with $\mathcal{G}$ and $\mathcal{D}ec_i$ reconstructs and synthesizes $\bar{x}_{(t)}, \bar{m}_{(t)}$, $\hat{x}_{(t)}, \hat{m}_{(t)}$ and $\tilde{x}_{(t)}, \tilde{m}_{(t)}$ (lines 4 - 10). Lines 11 to 34 show the optimization of each component in *IGAMT*, which can be divided into two stages: updating discriminators (line 11 - 20) and updating generators (line 21 - 34). The discriminators are updated less frequently than the generators at the ratio of f_h/f_b (line 11). To guarantee DP, gradient perturbations are applied when updating discriminators $\mathcal{D}_x$ (line 13-15), $\mathcal{D}_z$ (line 17-19) and the last layer of $\mathcal{D}ec$ (line 28-30).

DP Analysis

As mentioned before, once the model is trained, we are only releasing $\mathcal{G}$ and $\mathcal{D}ec_i$ for the synthesization. To guarantee DP of these two components, gradient perturbation is applied to the discriminators $\mathcal{D}_x$, $\mathcal{D}_z$, and the shared last layer $\mathcal{D}ec_i^{last}$ of $\mathcal{D}ec$ and $\mathcal{D}ec_i$. As $\mathcal{D}_x$, $\mathcal{D}_z$ and $\mathcal{D}ec_i^{last}$ are trained

using the same dataset, the overall privacy can be analyzed under simple composition, and DP guarantee for each part is analyzed under moment accountant (Algorithm 1). Therefore, the total privacy of the final generative model ($\mathcal{G}$ and $\mathcal{D}ec_i$) is (ϵ_1+ϵ_2+ϵ_3, δ_1+δ_2+δ_3)-DP if $\mathcal{D}_x$, $\mathcal{D}_z$ and $\mathcal{D}ec^{last}$ are (ϵ_1, δ_1)-DP, (ϵ_2, δ_2)-DP and (ϵ_3, δ_3)-DP respectively.

5 Experiments

In this section, we demonstrate the effectiveness of *IGAMT*[*] using synthetic EHRs from two aspects: visual similarity to real data and downstream applications with comparable performance to real data.

Experimental Setup

Baselines. We compare *IGAMT* with *DAAE*, the existing state-of-the-art generative model for EHR data with DP. Since it is incapable of capturing non-temporal features, missing values, and irregular measures, we slightly adapt the it to conduct a fair comparison. We also build four more baselines: VAE (Variational Autoencoder), GAN, VAE-GAN (Larsen et al. 2016), and AAE (Makhzani et al. 2015) to have a more comprehensive comparison.

EHRs and data preprocessing. We use two EHR datasets in this work. One is Physionet MIMIC-IV-ED (Goldberger et al. 2000), which is an open-sourced EHR dataset that encompasses over 425,000 ED stays collected from emergency department (ED) admissions from 2011 to 2019. In the paper, we utilize a subset that covers all vital sign data, which comprises 14,024 training, 1,753 validation, and 1,754 testing records. The other one is from the Emory Synergy project, which contains 5,747 training, 718 validation, and 719 testing records.

EHR data is preprocessed before feeding into the model. For temporal feature preprocessing, we first normalize them to the range of [0, 1]. Then for the irregular measures in the time-space (Section 4), we extract the time features and follow a similar process to scale them to [0, 1]. We also pad the time feature of all the examples to 50. For non-temporal feature preprocessing (Section 4), we similarly normalize them to [0, 1]. Then, we transform the discrete features into one-hot vectors to form the start features, which have the same size as the temporal features of each timestep.

The preprocessing of missing values (Section 4) is to generate a mask consisting of 1 and 0s where 0 represents the missing values. After preprocessing, each record has 50 time steps with each timestep having 10 and 9 features for MIMIC-IV-ED and Emory Synergy respectively.

Privacy budget. The privacy budgets used in the experiments are $(\epsilon, \delta) = (1.5, 1e-5)$. Our experiments currently use equal budget allocation among the three components.

Experimental Results

Evaluation 1. PCA visulization. We use PCA to reduce the real and synthetic data to two-dimensional space and visually show the difference between real and synthetic EHR. PCA results aim to validate IGAMT's ability to capture the feature distributions of real EHR by measuring subspace similarity. It reflects whether synthetic data maintains the underlying structure and correlations present in the real data. Figure 4 and Figure 5 demonstrate the results on MIMIC-IV-ED and Emory Synergy datasets. In both figures, the blue dots represent the real EHR, and the green dots represent the synthetic EHR from different models. The first row shows the non-DP results of the baseline model and *IGAMT* and the second row shows the results from the DP version of models corresponding to the first row. From the result, we can note that after dimension reduction, the synthetic data generated by *IGAMT* can fit the real data the best. The similarity in the principal components suggests that the subspace of the synthetic data closely aligns with that of the real data in terms of the inherent similarity and underlying structure, which indicates that *IGAMT* is well-designed for the synthesization of temporal EHR compared with the baseline architectures.

In addition, the DP technique applied during training can degrade the performance of baseline models. This trend is more notable when the architecture is more complex. However, for *IGAMT*, incorporating gradient perturbation does not compromise the model utility which verifies the effectiveness of the *Imitator* module. It overcomes the large randomization typically required for the generator and significantly enhances privacy utility trade-off.

Evaluation 2. Closer look at the feature similarity. To provide a more detailed comparison of temporal features between real and synthetic EHRs, we pick three vital temporal features ("time in year", "heart rate", "SBP"), and randomly sample 100 EHRs from real test data and synthetic data, and plot the average value of the three selected features over 50 time steps. As shown in Figure 8, the blue curve represents the real EHRs, and black represents EHRs from *IGAMT*. For all three feature plots, black curves partially match the patterns of real features and outperform *DAAE*, which indicates that the synthetic temporal features generated from *IGAMT* better maintain the characteristics of real temporal features.

We also compare the KL divergence of feature distributions between real and synthetic data generated by *IGAMT* and *DAAE*. As shown in Table 1, *IGAMT* dominates on almost all features, especially on features #2, #3, #5 #6, #8.

To illustrate the statistics of missing values and irregular measures in EHRs, we count the mark-off positions per feature in masks and plot the histogram of counts averaged over features among 1000 samples, and calculate the elapsed time between two neighboring time steps and plot the histogram of the elapsed time averaged over time steps among 1000 samples. The results are shown in Figure 6 and Figure 7 respectively. As can be seen, while all baseline models fail, *IGAMT* is able to capture the distributions of missing values and elapsed time between visits that resemble the real data, thanks to its time embedding and the missing values masks.

Evaluation 3. Unsupervised downstream application: clustering. To demonstrate that synthetic data generated by *IGAMT* are not only visually similar to the real data but also maintain the same characteristics of the real data for downstream tasks, we conduct unsupervised and super-

[*]https://github.com/Emory-AIMS/IGAMT

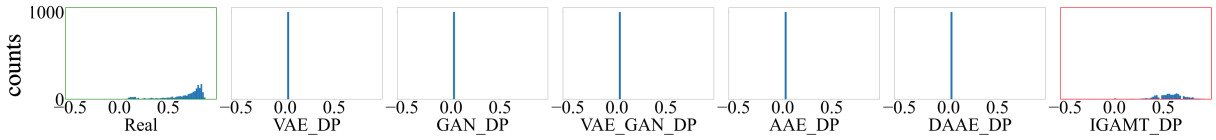


Figure 6: Feature similarity: missing values histogram of real and synthetic EHRs

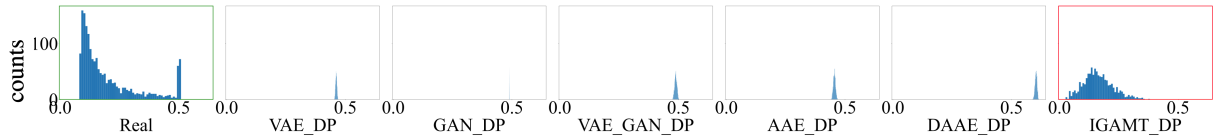


Figure 7: Feature similarity: elapsed time histograms of real and synthetic EHRs

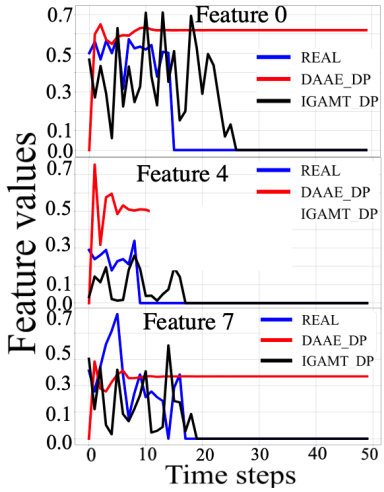


Figure 8: Visualization of three vital features

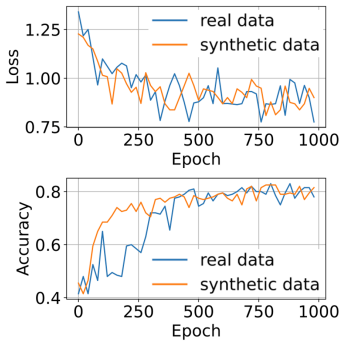


Figure 10: Downstream classification performance

vised downstream applications. First, we use clustering as an unsupervised downstream application to show synthetic data and real data have similar clustering results. We use Minkowski distance and cosine similarity to measure the distance of clustering centers from synthetic data and real data. The lower Minkowski distance and higher cosine similarity represent better performance.

As shown in Table 2, *IGAMT* outperforms all other baseline models by achieving the smallest Minkowski distance and highest cosine similarity overall models. This indicates that the synthetic data generated by *IGAMT* can best maintain the clustering performance of the real data, which reflects the higher synthetic ability of *IGAMT*.

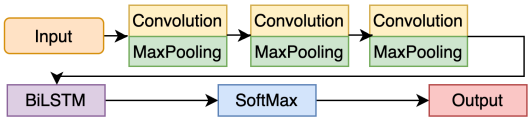


Figure 9: Downstream Classifier Architecture

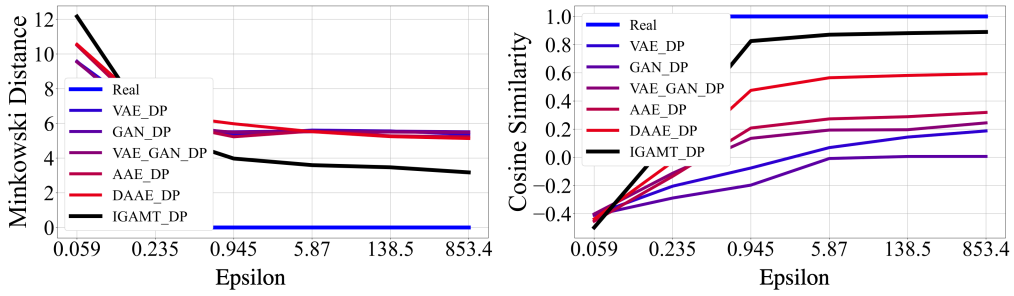
Evaluation 4. Supervised downstream application: classification. Another downstream application involved is the classification task. We label the data by separating the "DBP" feature values into 4 categories. DBP stands for diastolic blood pressure, which measures the pressure in the arteries. The high DBP is an indicator of hypertension, thus we use different ranges of DBP values as labels to train a DBP classification model.

We train classifiers for real and synthetic data respectively using the same size of training data with the same training epochs under the same architecture. A CNN-LSTM (Sainath et al. 2015) model is adopted as shown in Figure 9. Initially, the data undergoes permutation and is fed into three layers of CNN. Then the data is reshaped and fed into a bidirectional LSTM with a hidden size of 128. As shown in Figure 10, after 1000 training epochs, the training loss and test accuracy for the real data converge to 0.866 and 80.50% respectively, while the synthetic data demonstrates a loss of 0.871 and an accuracy of 82.00%. These results indicate that the synthetic EHR by *IGAMT* can maintain the characteristics of the real EHR and achieve comparable downstream performance.

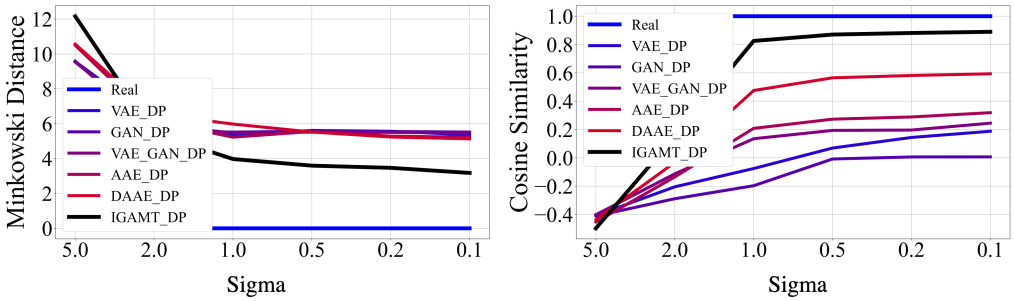
Evaluation 5. Privacy analysis. The challenge of applying DP in deep learning systems is to balance the utility-privacy trade-off. Figure 11a and 11b demonstrate the overall Minkowski distance and cosine similarity of the downstream clustering results using models with different DP budget ϵ and perturbation magnitude σ, where the black curve represents the *IGAMT* while other curves represent the baseline models. We can note that under the same privacy

Model	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9
$DAAE_{DP}$	0.2	25	24.9	33	11.01	15.18	46.8	48.33	32.7	45.02
$IGAMT_{DP}$	0.2	20.78	7.88	3.37	11.09	0.75	0.04	34.06	2.98	33.87

Table 1: Feature similarity: KL divergence of feature distribution



(a) Downstream clustering performance with different ϵ



(b) Downstream clustering performance with different σ

Figure 11: Minkowski distance and cosine similarity on different models.

Model	Minkowski Distance	Cosine Similarity
VAE_{DP}	5.36	-0.07
GAN_{DP}	5.51	-0.19
VAE_GAN_{DP}	5.47	-0.16
AAE_{DP}	5.22	0.20
$DAAE_{DP}$	5.97	0.47
$IGAMT_{DP}$	**3.97**	**0.82**

Table 2: Downstream clustering performance

budget (the same ϵ and σ), *IGAMT* can achieve the lowest Minkowski distance and highest cosine similarity, achieving the best synthetic performance.

6 Conclusions and Future Works

In this paper, we proposed a novel framework *IGAMT* to generate differentially private EHRs with heterogeneous features, missing values, and irregular measures. *IGAMT* leverages missing value masks and sequence-to-sequence transformers with well-designed embeddings to learn the underlying characteristics of EHRs and generate synthetic data of high quality. By leveraging the elaborate architecture and objective functions, the *Imitator* of *IGAMT* is capable of imitating the behaviors of the decoder while reducing the randomization required to achieve DP for the generator. After training with gradient perturbation, *IGAMT* will release $\mathcal{G}$ and $\mathcal{D}ec_i$ including the last shared layer with $\mathcal{D}ec$ as a DP generative model. We demonstrate that *IGAMT* achieves state-of-the-art performance in synthesizing DP EHRs.

Our experiments currently use equal privacy budget allocation among the three DP components. It can be further optimized for future work. For example, the gradient perturbation of the discriminator $\mathcal{D}_x$ will not be needed if we are not using its loss for updating generator $\mathcal{G}$. We also plan to utilize the more advanced DP analysis approach such as (Balle and Wang 2018; Wang et al. 2023) for tighter privacy analysis and further improve privacy and utility trade-off.

Acknowledgements

We thank all reviewers for their constructive comments. This work is partially supported by the National Natural Science Foundation of China (NSFC) 62206207, National Institutes of Health (NIH) R01LM013712, R01ES033241, UL1TR002378, National Science Foundation (NSF) IIS-2302968, CNS-2124104, CNS-2125530, and a Synergy II Nexus Award (Differentially-Private, Synthetic Controls for the Center for Health Discovery and Well-Being (CHDWB) Cohort: Data Science to Assess Health, Wellness and Disease) from the Woodruff Health Science Center of Emory University. The content is the responsibility of the authors and does not represent the official views of the sponsors.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Balle, B.; and Wang, Y.-X. 2018. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, 394–403. PMLR.

Bang, S.-J.; Wang, Y.; and Yang, Y. 2020. Phased-lstm based predictive model for longitudinal ehr data with missing values.

Baowaly, M. K.; Lin, C.-C.; Liu, C.-L.; and Chen, K.-T. 2019. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3): 228–241.

Bassily, R.; Smith, A.; and Thakurta, A. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 464–473. IEEE.

Beaulieu-Jones, B. K.; Wu, Z. S.; Williams, C.; Lee, R.; Bhavnani, S. P.; Byrd, J. B.; and Greene, C. S. 2019. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7): e005122.

Chin-Cheong, K.; Sutter, T.; and Vogt, J. E. 2020. Generation of differentially private heterogeneous electronic health records. *arXiv preprint arXiv:2006.03423*.

Choi, E.; Biswal, S.; Malin, B.; Duke, J.; Stewart, W. F.; and Sun, J. 2017. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, 286–305. PMLR.

Dwork, C. 2011. A firm foundation for private data analysis. *Communications of the ACM*, 54(1): 86–95.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4): 211–407.

Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K.; and Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786): 504–507.

Hjelm, R. D.; Jacob, A. P.; Che, T.; Trischler, A.; Cho, K.; and Bengio, Y. 2017. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*.

Hyland, S.; Esteban, C.; and Rätsch, G. 2018. Real-valued (medical) time series generation with recurrent conditional gans.

Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; and Winther, O. 2016. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, 1558–1566. PMLR.

Lee, D.; Yu, H.; Jiang, X.; Rogith, D.; Gudala, M.; Tejani, M.; Zhang, Q.; and Xiong, L. 2020. Generating sequential electronic health records using dual adversarial autoencoder. *Journal of the American Medical Informatics Association*, 27(9): 1411–1419.

Lee, J.; and Kifer, D. 2018. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1656–1665.

Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; and Frey, B. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Neil, D.; Pfeiffer, M.; and Liu, S.-C. 2016. Phased lstm: Accelerating recurrent network training for long or event-based sequences. *arXiv preprint arXiv:1610.09513*.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, 2642–2651. PMLR.

Rahman, M. A.; Rahman, T.; Laganière, R.; Mohammed, N.; and Wang, Y. 2018. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.*, 11(1): 61–79.

Sainath, T. N.; Vinyals, O.; Senior, A.; and Sak, H. 2015. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4580–4584.

Shickel, B.; Tighe, P. J.; Bihorac, A.; and Rashidi, P. 2017. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE journal of biomedical and health informatics*, 22(5): 1589–1604.

Song, S.; Chaudhuri, K.; and Sarwate, A. D. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, 245–248. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, C.; Su, B.; Ye, J.; Shokri, R.; and Su, W. J. 2023. Unified Enhancement of Privacy Bounds for Mixture Mechanisms via f-Differential Privacy. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wang, D.; Ye, M.; and Xu, J. 2017. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, 2722–2731.

Wang, W.; Tang, P.; Lou, J.; and Xiong, L. 2021. Certified robustness to word substitution attack with differential privacy. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1102–1112.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. Modeling tabular data using conditional gan. *arXiv preprint arXiv:1907.00503*.

Yu, L.; Liu, L.; Pu, C.; Gursoy, M. E.; and Truex, S. 2019. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, 332–349. IEEE.