Synthesizing Multimodal Electronic Health Records via Predictive Diffusion Models

Yuan Zhong
The Pennsylvania State University
University Park, PA, USA
yfz5556@psu.edu

Xiaochen Wang
The Pennsylvania State University
University Park, PA, USA
xcwang@psu.edu

Jiaqi Wang
The Pennsylvania State University
University Park, PA, USA
jqwang@psu.edu

Xiaokun Zhang
Dalian University of Technology
Dalian, Liaoning, China
dawnkun1993@gmail.com

Yaqing Wang
Purdue University
West Lafayette, IN, USA
wang5075@purdue.edu

Mengdi Huai
Iowa State University
Ames, IA, USA
mdhuai@iastate.edu

Cao Xiao
GE Healthcare
Seattle, WA, USA
Cao.Xiao@gehealthcare.com

Fenglong Ma*
The Penn State University
University Park, PA, USA
fenglong@psu.edu

Abstract

Synthesizing electronic health records (EHR) data has become a preferred strategy to address data scarcity, improve data quality, and model fairness in healthcare. However, existing approaches for EHR data generation predominantly rely on state-of-the-art generative techniques like generative adversarial networks, variational autoencoders, and language models. These methods typically replicate input visits, resulting in inadequate modeling of temporal dependencies between visits and overlooking the generation of time information, a crucial element in EHR data. Moreover, their ability to learn visit representations is limited due to simple linear mapping functions, thus compromising generation quality. To address these limitations, we propose a novel EHR data generation model called EHRPD. It is a diffusion-based model designed to predict the next visit based on the current one while also incorporating time interval estimation. To enhance generation quality and diversity, we introduce a novel time-aware visit embedding module and a pioneering predictive denoising diffusion probabilistic model (P-DDPM). Additionally, we devise a predictive U-Net (PU-Net) to optimize P-DDPM. We conduct experiments on two public datasets and evaluate EHRPD from fidelity, privacy, and utility perspectives. The experimental results demonstrate the efficacy and utility of the proposed EHRPD in addressing the aforementioned limitations and advancing EHR data generation.

CCS Concepts

• **Information systems** → *Data mining*; • **Applied computing** → **Health informatics**; • **Computing methodologies** → Artificial intelligence; *Neural networks*.

*Corresponding authors.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671836

Keywords

Electronic Health Records, Medical Data Synthesis, Diffusion Models, Multimodal Data Mining

ACM Reference Format:
Yuan Zhong, Xiaochen Wang, Jiaqi Wang, Xiaokun Zhang, Yaqing Wang, Mengdi Huai, Cao Xiao, and Fenglong Ma. 2024. Synthesizing Multimodal Electronic Health Records via Predictive Diffusion Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671836

1 Introduction

In healthcare, the utilization of Electronic Health Records (EHR) data is pivotal for advancing data-driven methodologies in both research and clinical practice [34]. EHR data possess distinctive characteristics, as illustrated in Figure 1, including sequential and temporal visit records, irregular time intervals between consecutive visits, and the presence of multiple modalities. However, effectively harnessing such intricate data encounters a significant challenge due to the scarcity of high-quality EHR datasets. To address this challenge, the generation of EHR data becomes an essential solution, providing a means to produce synthetic yet realistic supplements of patient data for constructing robust healthcare application models.

Recent advancements in EHR data generation primarily depend on cutting-edge generative techniques, such as generative adversarial networks (GAN) [3, 6, 36], variational autoencoders (VAE) [4, 8], and language models (LM) [30, 32]. These methodologies adhere to a common pipeline, as depicted in Figure 2. This pipeline generally includes an encoder to encode visit V_i into a representation $\mathbf{v}_i$, a generative model to generate the latent representation $\hat{\mathbf{v}}_i$, and a decoder to map $\hat{\mathbf{v}}_i$ to the generated visit $\hat{V}_i$. Despite their notable performance achievements, they still encounter several limitations.

• **Inadequate modeling of temporal dependencies between visits.** Real EHR data, as shown in Figure 1, comprises visits ordered in time, with inherent temporal dependencies among them. However, existing methodologies employ a visit-replicating approach, generating a synthesis $\hat{V}_i$ for the input V_i without explicitly addressing the temporal relationships between visits. An optimal generative model should inherently incorporate these temporal characteristics, such as directly using the current visit V_i to generate the next visit V_{i+1}.

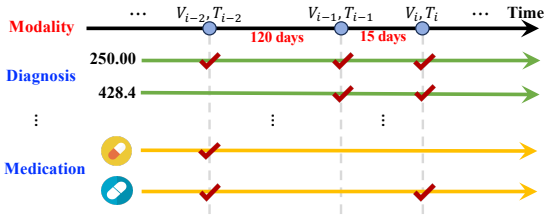

Figure 1: An example of multimodal EHR data, where V_i denotes the visit information and T_i represents its time.

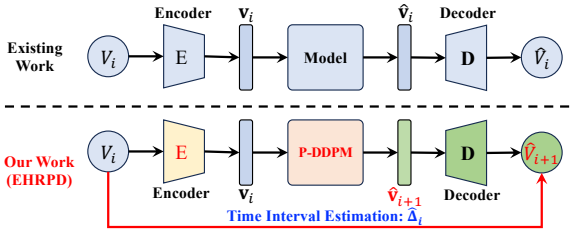

Figure 2: Pipeline comparison between existing approaches and our proposed EHRPD.

• **Failure to simultaneously generate time intervals between visits.** Current models aimed at generating detailed data frequently overlook a crucial aspect of patient healthcare: time information. As illustrated in Figure 1, time information is a vital component of EHR data, playing a significant role in modeling disease progression. Therefore, effectively capturing temporal dependencies between visits necessitates incorporating the modeling of time intervals between visits concurrently, enabling accurate characterization of patient health trajectories.

• **Limited capability in learning visit representations.** Owing to the discrete nature of certain EHR modalities like diagnosis codes, procedures, and medication codes, existing models embed the input visit V_i into a continuous representation $\mathbf{v}_i$ first, as depicted in Figure 2. The quality of the generated EHR data directly hinges on $\mathbf{v}_i$. However, current methods only utilize simple linear layers as the mapping function, potentially insufficient for representing the complexity of EHR data. Hence, there is a need to explore alternative approaches for learning visit representations.

• **Lack of a robust generation model to balance data diversity and quality.** From a modeling perspective, GAN-based approaches encounter the issue of model collapsing during training [3, 6, 36], while VAE-based approaches rely on a strong Gaussian assumption that may not align well with EHR data [4, 8]. LM-based approaches either depend on additional knowledge to generate diverse data, making it difficult to control data quality [32], or utilize autoregressive masked language training techniques to ensure quality but sacrifice diversity [30]. None of the existing models adequately address this challenging task. Therefore, the development of a powerful and comprehensive EHR generation model is urgently needed in healthcare.

To comprehensively address the aforementioned limitations, we introduce EHRPD[1] in this paper, a diffusion-based model outlined in Figure 3. Unlike existing approaches, EHRPD aims to capture the temporal characteristic of EHR data by generating the next visit $\hat{V}_{i+1}$ based on the current visit V_i. Specifically, EHRPD takes multimodal EHR visit $V_i = \{M_i^1, \cdots, M_i^N\}$ as input, where N denotes the number of modalities. Initially, EHRPD encodes the input V_i using the designed **time-aware visit embedding** module, which facilitates the modeling of fine-grained code appearance patterns concerning time intervals when learning the visit embedding, denoted as $\mathbf{v}_i$.

The learned visit embedding $\mathbf{v}_i$ is then utilized to generate the latent representation of the subsequent visit $\hat{\mathbf{v}}_{i+1}$ via a novel **predictive denoising diffusion probabilistic model** (P-DDPM). P-DDPM comprises three key processes – a forward noise addition process, a backward denoising diffusion process, and a predictive mapping process to encapsulate the temporality between visits in each diffusion step. To learn the latent representation $\hat{\mathbf{v}}_{i+1}$, we integrate the backward denoising diffusion process with a novel predictive U-Net (PU-Net). The obtained representation $\hat{\mathbf{v}}_{i+1}$ is then employed to generate multimodal EHR data $\hat{V}_{i+1}$ through the decoding **EHR prediction** module. The **catalyst representation learning** module is dedicated to estimating the time interval between V_i and $\hat{V}_{i+1}$, as well as assembling the catalyst information $\mathbf{\Phi}_i$ used in PU-Net, including demographics $\mathcal{D}$, historical EHR representation $\mathbf{h}_i$, and the estimated time interval embedding $\hat{\Delta}_i$.

In summary, the proposed EHRPD not only addresses the modeling of temporality between visits but also facilitates the simultaneous estimation of time intervals. Furthermore, the introduced time-aware visit embedding module can learn comprehensive visit embeddings by explicitly capturing code appearance patterns. Additionally, the design of P-DDPM inherits the robustness of existing DDPMs [15, 37] while leveraging the diversity and quality of the generated EHR data through the noise addition and denoising process via the proposed PU-Net. Finally, extensive experiments are conducted on MIMIC-III and Breast Cancer Trial datasets to validate the proposed EHRPD from fidelity, privacy, and utility perspectives. Experimental results demonstrate the superiority of EHRPD in EHR generation.

2 Related Work

EHR Data Generation. To generate synthetic medical data to alleviate data scarcity, data generation methods in the medical domain that are equipped with GAN [3, 6, 36, 38], VAE [4, 8], LM [30, 32], and DDPM [15, 37] have shown great success from their debut. Earlier methods [3, 6] perform visit-level code aggregation to produce one or a few feature vectors and generate synthetic ones with GAN. However, this summarization would inevitably lose temporal dynamics and lead to inferior performance. To address this problem, recent work [4, 6, 8, 30, 32, 36] aims to generate EHR data on the visit level. These fine-grained methods learn and leverage the hidden visit-wise relationship in EHR data with sequential learning techniques such as Tansformer, achieving state-of-the-art results. However, these methods ignore the time information of the patient's visit that contains crucial information such as disease progression and thus are suboptimal in their performance.

Denoising Diffusion Probabilistic Models. The diffusion model has achieved considerable success in various tasks. One of its primary applications is image generation, as demonstrated in

[1] EHRPD code repository: https://anonymous.4open.science/r/EHRPD-465B

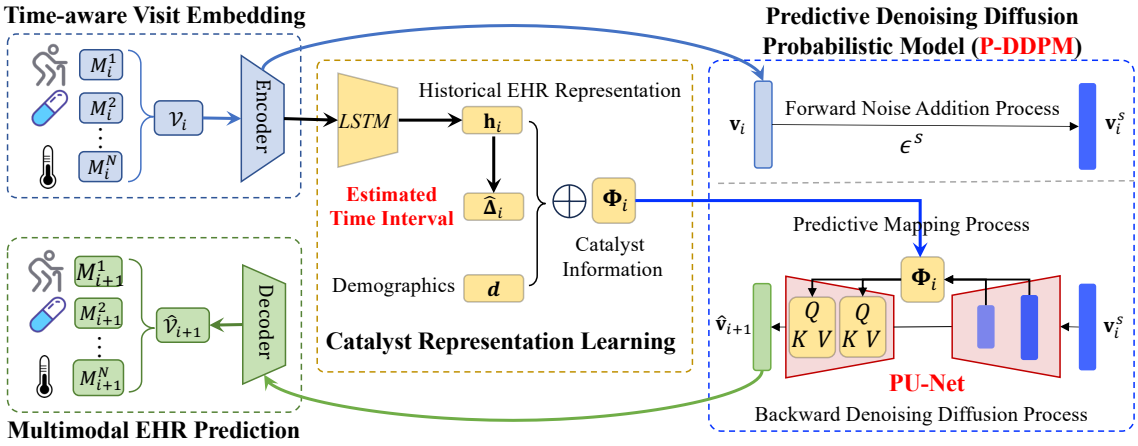

Figure 3: Overview of the proposed EHRPD model.

works [16, 25, 26]. It can also be adapted to time series forecasting and imputation [24, 25, 29]. Besides, the Discrete Diffusion Model [1] adapts diffusion to discrete data space and fosters work such as [10, 17]. Specific to the medical domain, the diffusion model has been used to generate healthcare data, including works such as [11, 15, 22, 37]. However, these methods are either task-oriented or not designed for sequential EHR generation.

3 Methodology

This work is dedicated to the generation of realistic high-dimensional, longitudinal, and multimodal Electronic Health Record (EHR) data. Given the sequential temporality inherent in EHR data, our objective is to simultaneously generate the next visit $\hat{V}_{i+1}$ and its associated time interval $\hat{\Delta}_i$ (i.e., $\hat{\Delta}_i = \hat{T}_{i+1} - T_i$), where $\hat{T}_{i+1}$ means the estimated time for $\hat{V}_{i+1}$. This generation process relies on the entire historical visit sequence $\mathcal{V}_{1:i} = [(V_1, T_1), \cdots, (V_i, T_i)]$ in conjunction with demographic information $\mathcal{D}$. Mathematically, this is represented as:

$$\{\hat{V}_{i+1}, \hat{\Delta}_i\} = g(\mathcal{V}_{1:i}, \mathcal{D}), \tag{1}$$

where $g(\cdot)$ denotes the generation function. Each visit $V_i = \{M_i^1, \cdots, M_i^N\}$ encompasses N modalities, including diagnosis codes, medication codes, lab test items, and so on.

To achieve this, we propose a novel diffusion-based EHR generation model, referred to as EHRPD, illustrated in Figure 3. This model comprises four main modules: (1) time-aware visit embedding, (2) predictive denoising diffusion probabilistic model, (3) catalyst representation learning, and (4) multimodal EHR prediction. In the following section, we provide a detailed explanation of each module.

3.1 Time-aware Visit Embedding

The easiest way to embed each visit $V_i = \{M_i^1, \cdots, M_i^N\}$ is applying a linear mapping function for each modality M_i^n on its modality-level binary representation $\mathbf{y}_i^n \in \{0, 1\}^{|C_n|}$. However, this approach ignores nuances of the code's evolution against time. To address this deficiency, we propose a time-aware visit embedding approach to capture the temporal evolution of medical code individually.

Time-aware Code Embedding. For the j-th code $c_i^{n,j}$ that appears in the n-th modality, we record its most recent appearance time, which is then subtracted by T_i to obtain the code-level time gap denoted as $\tau_i^{n,j}$. $\tau_i^{n,j} = 0$ for the first visit. A smaller $\tau_i^{n,j}$ usually indicates a higher importance level for time-aware code embedding learning, which is described as follows:

$$\mathbf{c}_i^{n,j} = \pi_i^{n,j} \text{MLP}_c([\mathbf{e}_i^{n,j}; \boldsymbol{\tau}_i^{n,j}]) + (1 - \pi_i^{n,j})\mathbf{e}_i^{n,j}, \tag{2}$$

where $[;]$ denotes the concatenation operation, $\mathbf{e}_i^{n,j}$ is the basic code embedding, and $\boldsymbol{\tau}_i^{n,j}$ is the time gap embedding calculated by the positional embedding used in Transformer. $\pi_i^{n,j}$ is a gating indicator to decide whether to incorporate time gap information into code embedding learning, which is obtained via a Gumbel-Softmax layer as follows:

$$\pi_i^{n,j} = \text{Binarize}\left(\frac{\exp\left((\log(\mathbf{p}_i^{n,j}[0]) + G_0)/\eta\right)}{\sum_{y=0}^{1}\exp\left((\log(\mathbf{p}_i^{n,j}[y]) + G_y)/\eta\right)}\right), \tag{3}$$

where $\mathbf{p}_i^{n,j}$ is the softmax layer output on top of a linear function on the concatenated $[\mathbf{e}_i^{n,j}; \boldsymbol{\tau}_i^{n,j}]$, G is the noise following the Gumbel distribution, and η is a hyperparameter.

Visit Embedding. We use a modality-level attention mechanism to learn the aggregated time-aware visit embedding as follows:

$$\begin{aligned}
\mathbf{v}_i &= \sum_{n=1}^{N} \psi_i^n \mathbf{z}_i^n, \\
\boldsymbol{\psi}_i &= \text{Softmax}\left(\text{MLP}_\psi([\mathbf{z}_i^1; \cdots; \mathbf{z}_i^N])\right), \\
\mathbf{z}_i^n &= \text{ReLU}\left(\text{MLP}_z\left(\sum_{j=1}^{|C_n|} \mathbf{c}_i^{n,j}\right)\right).
\end{aligned} \tag{4}$$

3.2 Predictive Denoising Diffusion Probabilistic Models (P-DDPM)

Existing diffusion-based models, like DDPM [12] and Glide [23], achieve generation by reconstructing the original input data. However, the task of EHR generation differs significantly from other tasks, as it aims to generate sequential, time-ordered EHR data using Eq. (1) instead of reconstructing input data. To address this distinction, we propose a novel approach called the Predictive Denoising Diffusion Probabilistic Model (P-DDPM) to generate the visit information V_{i+1}. Specifically, we treat the visit V_i as a reactant and V_{i+1} as the product. In generating V_{i+1} using V_i, the aggregated information from $\mathcal{V}_{1:i}$ and $\mathcal{D}$ can be treated as the catalyst in P-DDPM. Thus, the proposed P-DDPM contains three components – a forward noise addition process, a predictive mapping process, and a backward denoising diffusion process – to tackle the challenges associated with sequential EHR data generation effectively.

Forward Noise Addition Process. The forward noise addition process is fixed to a Markov chain that gradually adds Gaussian noise to the representation of V_i (i.e., $\mathbf{v}_i$ or $\mathbf{v}_i^0$ detailed in Section 3.1 Eq. (4)) as follows:

$$\begin{aligned} q(\mathbf{v}_i^{1:S}|\mathbf{v}_i^0) &= \prod_{s=1}^{S} q(\mathbf{v}_i^s|\mathbf{v}_i^{s-1}), \\ q(\mathbf{v}_i^s|\mathbf{v}_i^{s-1}) &= \mathcal{N}(\mathbf{v}_i^s; \sqrt{1-\beta_s}\mathbf{v}_i^{s-1}, \beta_s\mathbf{I}), \end{aligned} \tag{5}$$

where S is the number of diffusion steps, $q(\mathbf{v}_i^{1:S}|\mathbf{v}_i^0)$ is the approximate posterior, and β_s is the variance schedule at step s. Let $\alpha_s = 1-\beta_s$ and $\bar{\alpha}_s = \prod_{j=1}^{s}\alpha_j$, we can reparametrize the above Gausssian steps in Eq. (5) to obtain the closed-form solution of $\mathbf{v}_i^s$ at any step s without adding noise step by step as follows:

$$\mathbf{v}_i^s = \sqrt{\alpha_s}\mathbf{v}_i^{s-1} + \sqrt{1-\alpha_s}\epsilon_{s-1} = \sqrt{\bar{\alpha}_s}\mathbf{v}_i^0 + \sqrt{1-\bar{\alpha}_s}\epsilon, \tag{6}$$

where $\epsilon \in \mathcal{N}(\mathbf{0}, \mathbf{I})$. The details of the forward process can be found in Appendix Section 6.1.1.

Predictive Mapping Process. To generate the next visit V_{i+1} using V_i, we need to first model the relationship between these two consecutive visits. In healthcare, such a relationship is usually modeled by a disease progress function, which is equivalent to a mapping function to predict V_{i+1} using V_i along with other information. Mathematically, we define such a predictive mapping function $f(\cdot)$ at each diffusion step as follows:

$$\mathbf{v}_{i+1}^s = f(\mathbf{v}_i^s, \mathbf{\Phi}_i), \tag{7}$$

where $\mathbf{\Phi}_i$ is the embedding of the aggregated information from $\mathcal{V}_{1:i}$ and $\mathcal{D}$ (detailed in Section 3.3), which plays a role of the catalyst during the generation.

Backward Denoising Diffusion Process. The backward denoising diffusion process in existing diffusion models aims to reverse the above forward process and sample from $q(\mathbf{v}_i^{s-1}|\mathbf{v}_i^s)$ to recreate the true sample $\mathbf{v}_i^0$. Different from these approaches, our work aims to generate the next visit's representation, i.e., $\mathbf{v}_{i+1}^0$, using $\mathbf{v}_i^0$ based on their relationship modeled in Eq. (7). Mathematically, the reverse process of $\mathbf{v}_{i+1}^0$ can be formulated as follows:

$$\begin{aligned} q(\mathbf{v}_{i+1}^{s-1}|\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0) &= q(\mathbf{v}_{i+1}^s|\mathbf{v}_{i+1}^{s-1}, \mathbf{v}_{i+1}^0)\frac{q(\mathbf{v}_{i+1}^{s-1}|\mathbf{v}_{i+1}^0)}{q(\mathbf{v}_{i+1}^s|\mathbf{v}_{i+1}^0)}, \\ q(\mathbf{v}_{i+1}^{s-1}|\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0) &= \mathcal{N}(\mathbf{v}_{i+1}^{s-1}; \hat{\boldsymbol{\mu}}_s(\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0), \hat{\beta}_s\mathbf{I}), \end{aligned} \tag{8}$$

By simplifying Eq. (8) according to the Gaussian distribution's density function, we can obtain the variance $\hat{\beta}_s$ and mean $\hat{\boldsymbol{\mu}}_s(\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0)$ of $q(\mathbf{v}_{i+1}^{s-1}|\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0)$ as follows:

$$\begin{aligned} \hat{\beta}_s &= \frac{1-\bar{\alpha}_{s-1}}{1-\bar{\alpha}_s}\beta_s, \\ \hat{\boldsymbol{\mu}}_s(\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0) &= \frac{\sqrt{\alpha}_s(1-\bar{\alpha}_{s-1})}{1-\bar{\alpha}_s}\mathbf{v}_{i+1}^s + \frac{\sqrt{\bar{\alpha}_{s-1}}\beta_s}{1-\bar{\alpha}_s}\mathbf{v}_{i+1}^0. \end{aligned} \tag{9}$$

Recall in the forward process, we have obtained $\mathbf{v}_{i+1}^s = \sqrt{\bar{\alpha}_s}\mathbf{v}_{i+1}^0 + \sqrt{1-\bar{\alpha}_s}\epsilon$ in Eq. (6). Thus, the mean value of the closed-form solution to the backward diffusion process can be obtained by substituting $\mathbf{v}_{i+1}^0$ in Eq. (9) as follows:

$$\hat{\boldsymbol{\mu}}_s(\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0) = \frac{1}{\sqrt{\alpha}_s}(\mathbf{v}_{i+1}^s - \frac{1-\alpha_s}{\sqrt{1-\bar{\alpha}_s}}\epsilon_s). \tag{10}$$

By substituting $\mathbf{v}_{i+1}^s$ in Eq. (10) with the predictive mapping process in Eq. (7), we finally have the closed-form solution as follows:

$$\hat{\boldsymbol{\mu}}_s(\mathbf{v}_{i+1}^s, \mathbf{v}_{i+1}^0) = \frac{1}{\sqrt{\alpha}_s}(f(\mathbf{v}_i^s, \mathbf{\Phi}_i) - \frac{1-\alpha_s}{\sqrt{1-\bar{\alpha}_s}}\epsilon_s). \tag{11}$$

The details of the derivation of the backward reverse process can be found in Appendix Section 6.1.3.

P-DDPM Learning. We typically use a U-Net with parameters θ to train the proposed P-DDPM by approximating Eq. (11), i.e.,

$$\boldsymbol{\mu}_\theta^s(\mathbf{v}_{i+1}^s, s) = \frac{1}{\sqrt{\alpha}_t}(f(\mathbf{v}_i^s, \mathbf{\Phi}_i) - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(f(\mathbf{v}_i^s, \mathbf{\Phi}_i), s)). \tag{12}$$

Different from conventional U-Net architecture which only takes the noised embedding as input, we design a new predictive U-Net (PU-Net) equipped with the capability to condition on $\mathbf{\Phi}_i$ during the generation process. PU-Net also contains two paths of learning – the downsampling path and the upsampling path.

PU-Net takes $\mathbf{v}_i^s$ as the input of the first layer. Then, at each layer l, the downsampling operations include a ResNet block with a 1-D convolution operation to generate the input of layer $l+1$ as follows:

$$\mathbf{v}_{i,l+1}^s = \text{Conv}(\text{ResBlock}(\mathbf{v}_{i,l}^s)). \tag{13}$$

The upsampling path at the l-th layer consists of an information aggregator, a ResNet block, and a deconvolutional (DeConv) operation to reconstruct the input. The information aggregator is a self-attention block (SelfAtt) to fuse the embeddings of $\mathbf{v}_{i,l}^s$ and the transformed catalyst embedding $\mathbf{\Phi}_{i,l}$ by a linear function on $\mathbf{\Phi}_i$. The upsampling operation can be formulated as follows:

$$\hat{\mathbf{v}}_{i+1,l}^s = \text{DeConv}(\text{ResBlock}(\hat{\mathbf{v}}_{i+1,l+1}^s, \text{SelfAtt}(\mathbf{v}_{i,l}^s, \mathbf{\Phi}_{i,l}))). \tag{14}$$

Figure 4 shows the designed PU-Net; the detailed derivation can be found in Appendix Section 6.2.

P-DDPM Reconstruction Loss. Let $\hat{\mathbf{v}}_{i+1} = \hat{\mathbf{v}}_{i+1,0}^s$ denote the output of PU-Net that is trained on a randomly selected diffusion step $s \in [1, \cdots, S]$. The objective function of PU-Net is the mean

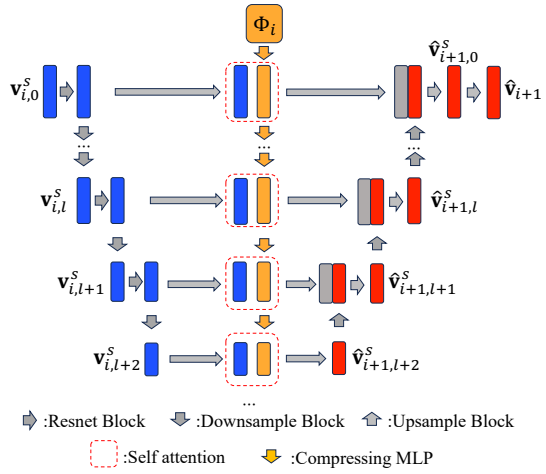

Figure 4: Illustration of PU-Net.

squared errors between the generated $\hat{\mathbf{v}}_{i+1}$ and the learned embedding $\mathbf{v}_{i+1}$ in Section 3.1 at each training epoch as follows:

$$\mathcal{L}_d(V_i) = \frac{1}{d_v}||\hat{\mathbf{v}}_{i+1} - \mathbf{v}_{i+1}||^2, \tag{15}$$

where d_v is dimension size of $\hat{\mathbf{v}}_{i+1}$, and the learned visit embedding $\mathbf{v}_{i+1}$ can be treated as ground truths.

Multimodal EHR Prediction. The predicted embedding $\hat{\mathbf{v}}_{i+1}$ can also be used to predict medical codes in each modality. Specifically, for each modality M_i^n, we use a linear layer to map $\hat{\mathbf{v}}_{i+1}$ to a modality-level representation, and then a Sigmoid function is used to predict the probability of a medical code on top of a multilayer perceptron (MLP) as follows:

$$\hat{\mathbf{y}}_{i+1}^n = \text{Sigmoid}(\text{MLP}_y(\hat{\mathbf{v}}_{i+1})), \tag{16}$$

Finally, we can use a Focal loss to train the multimodal EHR predictor as follows:

$$\begin{aligned}\mathcal{L}_e(V_i) = -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{|C_n|}\sum_{j=1}^{|C_n|}[y_{i+1}^{n,j}\kappa(1-\hat{y}_{i+1}^{n,j})^\gamma \log(\hat{y}_{i+1}^{n,j}) \\ + (1-y_{i+1}^{n,j})(1-\kappa)(\hat{y}_{i+1}^{n,j})^\gamma \log(1-\hat{y}_{i+1}^{n,j})],\end{aligned} \tag{17}$$

where $|C_n|$ denotes the number of identical codes in each modality M_i^n, $y_{i+1}^{n,j}$ is a binary ground truth to indicate whether the j-th code $c_{i+1}^{n,j}$ of the n-th modality presents in visit V_{i+1}, and κ and γ are hyperparameters.

To optimize Eqs. (15) and (17), we need to obtain the catalyst representation $\mathbf{\Phi}_i$. Next, we introduce the details of catalyst representation learning in Section 3.3.

3.3 Catalyst Representation $\mathbf{\Phi}_i$ Learning

As discussed in Section 3.2, the catalyst information is significantly important in the proposed EHRPD during the generation with P-DDPM, which "translates" the information from $\mathbf{v}_i^s$ to $\mathbf{v}_{i+1}^s$ at each diffusion step s via Eq. (7). Next, we explain how we construct the catalyst representation $\mathbf{\Phi}_i$.

EHR Historical Information Representation. In clinical practice, professionals often rely on a patient's historical medical records for a comprehensive view of their past health issues and as a crucial tool for informed decision-making. These records offer a timeline of medical events, treatments, and diagnoses that provide insights into the patient's health trajectory and are useful for predicting future health scenarios. Thus, we incorporate historical medical information as one of the conditioning factors to aid our generation process. Based on the learned visit embedding using Eq. (4), we utilize an LSTM network to accumulate a hidden state $\mathbf{h}_i$ for each visit as follows:

$$\mathbf{h}_i = \text{LSTM}(\mathbf{h}_{i-1}, \mathbf{v}_i). \tag{18}$$

Time Interval Estimation. Not only do clinical professionals diagnose a patient's health condition, but they also make a crucial decision in determining the optimal timing for the follow-up visit that best suits the current health condition of the patient. This decision is often to cope with the urgency and nature of the patient's condition: patients suffering from acute illnesses may need to return within a matter of days, while those with chronic diseases revisit with a more prolonged and periodic pattern. In our approach, we use the current health condition $\mathbf{h}_i$ to make predictions on the time interval till the next follow-up visit with a continuous time LSTM [21], as shown in Eq. (19). We utilize a linear layer on hidden state $\mathbf{h}_i$ to learn an intensity measure $\boldsymbol{\lambda}_i$ of the current visit, which represents a patient's medical urgency. This intensity is subtracted from 1, giving a close to 0 output if the patient's condition is urgent. Then, the second equation predicts the time gap and ensures it is strictly above 0 with the Softplus activation function:

$$\begin{aligned}\boldsymbol{\lambda}_i &= 1 - \tanh(\text{MLP}_\lambda(\mathbf{h}_i)), \\ \hat{\Delta}_i &= \text{Softplus}(\text{MLP}_\Delta(\boldsymbol{\lambda}_i)).\end{aligned} \tag{19}$$

Demographic Information Embedding. Demographic information is also treated as a key factor in decision-making. Thus, we encode the demographic information $\mathcal{D}$ into a dense representation $\mathbf{d}$ using an MLP layer, i.e., $\mathbf{d} = \text{MLP}_d(\mathcal{D})$.

Catalyst Representation Learning. Finally, the catalyst representation $\mathbf{\Phi}_i = [\mathbf{h}_i; \hat{\Delta}_i; \mathbf{d}]$ is obtained by concatenating the historical representation $\mathbf{h}_i$, the embedding of time interval through the positional embedding on $\hat{\Delta}_i$ (i.e., $\hat{\Delta}_i$), and the demographic embedding $\mathbf{d}$.

3.4 Time Interval Estimation Loss

The proposed EHRPD generates not only the next visit information but also the time interval between visits V_i and V_{i+1}. We take another MSE loss $\mathcal{L}_{\text{time}}$ between the real-time gap Δ_i and the predicted time gap $\hat{\Delta}_i$ using Eq. (19) as follows:

$$\mathcal{L}_t(V_i) = (\Delta_i - \hat{\Delta}_i)^2, \tag{20}$$

3.5 EHRPD Loss

Finally, we define the total loss $\mathcal{L}$ of a patient with $|\mathcal{V}|$ visits as the weighted sum of all three loss components by ω_d, ω_e, and ω_t, as follows:

$$\mathcal{L} = \frac{1}{|\mathcal{V}|-1}\sum_{i=1}^{|\mathcal{V}|-1}(\omega_d\mathcal{L}_d(V_i) + \omega_e\mathcal{L}_e(V_i) + \omega_t\mathcal{L}_t(V_i)). \tag{21}$$

4 Experiment

Due to the limited space, we put more results in the Appendix.

MIMIC-III		Breast Cancer Trial	
Total Patients	46,520	Total Patients	970
Diagnosis	1,071	Adverse Events	50
Drug Codes	1,439	Medications	100
Lab Items	710	Lab Categories	9
Procedure Codes	711	Treatments	4

Table 1: Statistics of two main datasets.

4.1 Datasets

We use two publicly available datasets to validate the performance of the proposed EHRPD, including MIMIC-III [14] and Breast Cancer Trial[2] from Project Data Sphere. The statistics of these two datasets are listed in Table 1. For the **MIMIC-III** dataset, we extract all 46, 520 patients' diagnoses, prescriptions, lab items, and procedure codes as four modalities of interest. For the **Breast Cancer Trial** dataset, we mainly follow the data preprocessing procedure described in TWIN [8], extracting adverse events, medications, lab categories, and treatment codes. For both datasets, each patient's EHR data is represented by a sequence of admissions ordered by admission time, where each admission consists of four lists of codes from each modality accordingly. Lastly, we add demographic information, such as sex, age, race, etc., as a static feature vector. We randomly split each dataset into train, validation, and test sets, with a ratio of 75 : 10 : 15.

4.2 Implementation Details

Our model is implemented in PyTorch and trained on an NVIDIA RTX A6000 GPU. We use the Adam optimizer with learning rate and weight decay both set to 10^{-3}. We set the Focal Loss parameter in Eq. (17) to $\kappa = 0.75$ and $\gamma = 5$. For the total EHRPD loss in Eq. (21), we set $\omega_d = 0.5$, $\omega_e = 1000$, and $\omega_t = 0.01$. The dimension of $\mathbf{h}$, $\mathbf{v}$, $\mathbf{\Delta}$, and $\mathbf{d}$ are set to 256, and the PU-Net dimension list is $[1024, 512, 256]$.

4.3 Fidality Assessment

We perform experiments to evaluate the generated data quality with two evaluation metrics and various baseline models, emphasizing the temporal coherence and cross-modality consistency of the generated data.

4.3.1 Baselines. Our selected baseline models include MLP, GAN-based models (medGAN [3] and synTEG [36]), VAE-based models (EVA [4] and TWIN [8]), diffusion-based approaches (TabDDPM [15], Meddiff [11], and ScoEHR [22]), and language model-based approaches (PromptEHR [32] and HALO [30]). Appendix Section 6.3 describes each model's detailed explanation.

4.3.2 Experiment Design and Evaluation Metrics. In this experiment, we use MIMIC-III and the Breast Cancer Trial as input databases separately. Each model produces a synthetic dataset corresponding to the original one, maintaining a 1:1 ratio. To assess the effectiveness of these EHR generation models, we focus on the following evaluation metrics. **Longitudinal Imputation Perplexity (LPL)** is a specialized metric used to evaluate EHR generation models. This metric adapts the traditional concept of perplexity from natural language processing to suit the unique temporal structure of EHR data. The LPL metric effectively captures the model's ability to predict the sequence of medical events over time, considering the chronological progression of a patient's health condition. In contrast to the LPL, which concentrates on the temporal coherence within a single modality, **Cross-modality Imputation Perplexity (MPL)** extends this concept to encompass the interrelations among different modalities, by assessing the model's proficiency in integrating and predicting across various types of data modalities, making it a more comprehensive measure of a model's ability to handle the multifaceted nature of EHR data.

4.3.3 Experimental Results. Table 2 shows the experimental results on the LPL and MPL metrics of all models tested on each of the data sources. *Lower score, indicates better model performance.* On the MIMIC-III dataset, our proposed model consistently outperforms other models across all four modalities in both LPL and MPL metrics. For instance, in the Diagnosis modality, our model achieves the best LPL score of 15.97 and MPL score of 17.95, significantly better than the next best baseline model, TWIN, which scored 26.28 and 27.68 in LPL and MPL, respectively. Though PromptEHR slightly outperforms our model in the Lab Category modality within the Breast Cancer Trial dataset, its performance across other modalities is less consistent. This variation indicates that while PromptEHR can be effective in certain scenarios, its output generally exhibits greater variability and less reliability compared to our model. Such inconsistency can lead to diminished effectiveness in diverse medical data scenarios, underlining our model's superior adaptability and robustness. Overall, our model's consistent performance across various metrics and modalities reinforces its effectiveness and broad applicability in medical data generation.

4.4 Privacy Assessment

We also evaluate the privacy-preserving capability of our model against other generation baseline models. The privacy-preserving capability is how likely the generated data can be traced back to the original data. We conduct our experiments with the Presence Disclosure Sensitivity metric.

4.4.1 Experiment Design and Evaluation Metric. We start with a predefined percentage of patient records from the training set, labeling them as "*known*" or "*compromised*". The aim is to identify these known records within the generated dataset. If the i-th visit of a patient is matched back to one of the synthetic visits generated by this patient by similarity score, we count it as a successful attack. We use the metric **Presence Disclosure Sensitivity (PD)** [8] to evaluate the security of our datasets. PD is the proportion of known patient records correctly matched in the generated dataset against the total number of compromised records. *The lower the PD value, the better the security performance.* This metric effectively gauges the risk of individual patient identification in the generated dataset, serving as an indicator of the dataset's privacy and data protection capabilities.

4.4.2 Experimental Results. Table 3 shows the experiment results on MIMIC-III in terms of PD with varying percentages of known patient records, ranging from 10% to 50%. Our analysis reveals that our model consistently outperforms the baseline models across all

[2]Clinical Trial ID: NCT00174655, https://www.projectdatasphere.org/

Dataset	Modality	Metric	MLP	medGAN	synTEG	EVA	TWIN	TabDDPM	Meddiff	ScoEHR	PromptEHR	HALO	EHRPD
MIMIC-III	Diagnosis	LPL	325.55	242.30	36.87	29.62	26.28	108.79	664.54	685.17	126.23	149.66	**15.97**
		MPL	352.54	257.48	45.61	31.63	27.68	114.18	670.91	691.55	128.05	192.13	**17.95**
	Drug	LPL	553.63	403.02	83.38	43.79	40.94	179.22	936.28	934.87	167.48	166.11	**20.53**
		MPL	551.67	405.75	82.66	44.02	40.86	178.70	936.14	950.27	136.04	202.01	**19.15**
	Lab Item	LPL	168.25	77.10	26.80	20.05	17.47	54.69	413.41	432.11	107.22	322.51	**15.11**
		MPL	166.61	87.12	30.34	19.97	17.41	54.44	412.33	431.09	98.52	303.09	**13.99**
	Procedure	LPL	290.38	234.81	49.33	27.39	21.26	98.03	471.81	486.81	51.18	22.68	**14.53**
		MPL	286.53	245.28	44.00	30.49	24.26	102.72	479.96	499.89	31.13	39.04	**18.89**
Breast Cancer Trial	Adverse Event	LPL	8.42	8.00	8.21	6.08	6.08	9.31	49.04	51.82	12.37	34.83	**5.96**
		MPL	9.37	9.42	9.70	8.30	8.52	10.86	50.13	51.57	12.14	31.51	**8.02**
	Medication	LPL	8.82	9.53	8.21	5.39	5.56	11.13	99.35	99.31	19.34	31.22	**4.96**
		MPL	8.73	11.67	10.08	6.95	7.10	12.95	98.83	99.10	19.80	33.61	**5.87**
	Lab Category	LPL	9.33	10.41	9.63	9.07	9.09	9.06	10.95	10.93	**8.55**	9.14	9.01
		MPL	9.22	10.08	10.03	9.09	9.11	9.03	10.96	10.97	**8.66**	9.28	9.09
	Treatment	LPL	7.29	9.43	9.09	3.09	3.12	3.67	4.77	5.01	5.10	3.44	**2.63**
		MPL	4.47	4.83	4.43	2.89	2.92	3.22	4.84	5.00	5.63	3.05	**2.41**

Table 2: EHR data generation evaluation of different approaches on two datasets with two metrics.

Approach	10%	20%	35%	50%
MLP	13.53	13.38	13.03	13.07
medGAN	17.06	17.19	17.56	17.79
synTEG	13.21	13.02	12.71	12.76
EVA	13.36	13.17	12.84	13.36
TWIN	13.36	13.16	12.84	12.89
TabDDPM	13.45	13.66	13.50	13.68
Meddiff	14.94	17.00	18.35	19.18
ScoEHR	14.12	15.56	16.44	16.91
PromptEHR	14.44	12.86	12.90	13.31
HALO	13.52	13.79	13.88	13.79
EHRPD	**12.60**	**12.77**	**12.53**	**12.25**

Table 3: Privacy assessment on MIMIC-III with different percentages of known patients under the metric PD.

tested scenarios. Notably, as the percentage of known patients increases, our model maintains its effectiveness in protecting patient privacy. For instance, at 10% known patients, our model achieves a Presence Disclosure Sensitivity of 12.60, and even with 50% known patients, it has the lowest metric of 12.25. This demonstrates a robust defense against privacy breaches, even as the challenge escalates with more known patient records. In comparison, other models like medGAN, TabDDPM, and PromptEHR show higher sensitivity, indicating a greater risk of patient identification in their generated datasets. For example, medGAN's sensitivity ranges from 17.06 to 17.79, which is significantly worse than ours. These results underscore the effectiveness of our model in ensuring the privacy and protection of patient data.

4.5 Utility Assessment

We experiment with the utility of the generated dataset from two databases on various downstream tasks under both multimodal and unimodal settings. We also conduct experiments to assess the effectiveness of the time gap prediction module of our model.

4.5.1 Data Preprocessing. We follow the FIDDLE [28] guidelines for data preprocessing and adapt their label definitions to process the MIMIC-III database, focusing on three critical health outcomes of a multimodal setting: Acute Respiratory Failure (ARF), Shock, and Mortality. Additionally, we employ another data preprocessing method from Retain [7] to obtain diagnosis codes for heart failure risk prediction, demonstrating our model's effectiveness in an unimodal context. Furthermore, following the work of TWIN [8], we select patients with severe outcomes and death as positive labels.

4.5.2 Multimodal Risk Prediction Analysis. To evaluate the quality of synthetic data generated by our approach, we designed an experiment to determine whether integrating synthetic data into the training process enhances the performance of downstream task-oriented models. We take all four time-series modalities and stationary demographic information as input features to conduct multimodal risk prediction experiments on acute respiratory failure (ARF), Shock, and Mortality datasets. We choose the following models as baselines: F-LSTM [28], F-CNN [28], RAIM [35], and DCMN [9], and three evaluation metrics: AUPR (the area under the Precision-Recall curve), F1 and Kappa, following [31]. *Baseline models and the results on ARF and Shock are explained in Appendix Section 6.4 and Section 6.7, respectively.*

For each dataset, models are trained using either only original data or a blend of synthetic and original data at a 1:1 ratio. The results, detailed in Table 4, reveal that our method, EHRPD, consistently outperforms baseline models under most conditions. Notably, under the Mortality task with the F-CNN architecture, EHRPD demonstrates a 2% improvement in both AUPR and F1 metrics compared to baselines. In contrast, the HALO model shows superior performance in specific metrics when paired with the F-LSTM architecture, suggesting a particularly effective synergy between HALO's synthetic data and the F-LSTM model. Overall, these results imply our model's potential to provide reliable synthetic data to augment multimodal risk prediction models.

4.5.3 Unimodal Risk Prediction Analysis. To simulate a scenario where multimodal data are unavailable, we conduct the following unimodal risk prediction task on Heart Failure disease with diagnosis code only. The backbone risk prediction models are LSTM [13], Dipole [19], Retain [7], AdaCare [20], and HiTANet [18], and are explained in Appendix Section 6.5. We utilize the same evaluation metric and synthetic-real data ratio as the multimodal experiment.

Model	F-LSTM			F-CNN			RAIM			DCMN		
Metric	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa
Orginal	0.5710	0.4705	0.4221	0.5810	0.5132	0.4554	0.5849	0.5000	0.4280	0.5438	0.4742	0.4298
MLP	0.6344	0.5408	0.4747	0.6614	0.5882	0.4950	0.6226	0.5571	0.4819	0.5733	0.4975	0.4245
medGAN	0.6210	0.5685	0.4946	0.6563	0.6098	0.5337	0.6159	0.5455	0.4789	0.5668	0.5473	0.4793
synTEG	0.6309	0.5556	0.4815	0.6597	0.6026	0.5220	0.6490	0.5891	0.5185	0.5804	0.5674	0.4958
EVA	0.6313	0.5572	0.4907	0.6487	0.5703	0.4897	0.6366	0.5438	0.4890	0.5585	0.5076	0.4326
TWIN	0.6410	0.5503	0.4846	0.6642	0.5929	0.5412	0.6469	0.5876	0.5292	0.6283	0.5687	0.4992
TabDDPM	0.6489	0.5586	0.4939	0.6534	0.5672	0.5022	0.6228	0.5572	0.4858	0.5428	0.5112	0.4354
Meddiff	0.6337	0.5502	0.4823	0.6163	0.5504	0.4636	0.6161	0.5289	0.4597	0.5594	0.4969	0.4186
ScoEHR	0.6408	0.5438	0.4812	0.6392	0.6033	0.5272	0.5949	0.4964	0.4191	0.6089	0.5333	0.4594
PromptEHR	0.6580	0.5677	0.5041	0.6682	0.6079	0.5383	0.6419	0.5648	0.4923	0.6279	0.6036	0.5351
HALO	**0.6673**	0.5547	0.4885	0.6139	0.5234	0.4562	0.5812	0.4779	0.4119	0.6124	0.5746	0.4957
EHRPD	0.6658	**0.5870**	**0.5251**	**0.6835**	**0.6159**	**0.5425**	**0.6548**	**0.5936**	**0.5327**	**0.6385**	**0.6147**	**0.5448**

Table 4: Result evaluation of the Mortality task on multimodal EHR data.

Backbone	Adacare			Dipole			HiTANet			LSTM			Retain		
Metric	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa
Orginal	0.6242	0.6136	0.3627	0.5856	0.5740	0.3349	0.6203	0.5978	0.3740	0.5943	0.5758	0.3461	0.5989	0.5913	0.3720
MLP	0.6640	0.6376	0.4185	0.6872	0.6470	0.4511	0.6692	0.6562	0.4488	0.6706	0.6374	0.4460	0.6476	0.6279	0.4158
medGAN	0.6669	0.6400	0.4089	0.6915	0.6371	0.4490	0.6781	0.6492	0.4376	0.6667	0.6332	0.4431	0.6424	0.6285	0.4028
synTEG	0.6711	0.6121	0.4142	0.6820	0.6215	0.4174	0.6851	0.6607	0.4641	0.6676	0.6312	0.4353	0.6319	0.6285	0.4224
EVA	0.6590	0.6424	0.4189	0.6795	0.6527	0.4513	0.6813	0.6511	0.4372	0.6629	0.6307	0.4267	0.6527	0.6240	0.4208
TWIN	0.6739	0.6252	0.4328	0.6603	0.6409	0.4406	0.6789	0.6690	0.4274	0.6546	0.6264	0.4162	0.6540	0.6382	0.4187
TabDDPM	0.6677	0.6243	0.3877	0.6851	0.6342	0.4312	0.6633	0.6581	0.4480	0.6687	0.6317	0.4301	0.6465	0.6152	0.4067
Meddiff	0.6659	0.6323	0.4171	0.6756	0.6249	0.4190	0.6684	0.6301	0.4277	0.6672	0.6188	0.4119	0.6591	0.6204	0.4161
ScoEHR	0.6701	0.6395	0.4284	0.6719	0.6296	0.4117	0.6774	0.6340	0.4238	0.6624	0.5980	0.3966	0.6469	0.6282	0.4166
PromptEHR	0.6810	0.6462	0.4100	0.6748	0.6359	0.4334	0.6541	0.6182	0.4076	0.6642	0.6222	0.4178	0.6582	0.6251	0.4211
HALO	0.6742	0.6312	0.4295	0.6907	0.6562	0.4604	0.6841	0.6578	0.4489	0.6619	0.6301	0.4252	0.6518	0.6266	0.4196
EHRPD	**0.6856**	**0.6523**	**0.4385**	**0.7018**	**0.6630**	**0.4735**	**0.7017**	**0.6777**	**0.4699**	**0.6824**	**0.6484**	**0.4506**	**0.6603**	**0.6397**	**0.4382**

Table 5: Result evaluation on Heart Failure prediction task on unimodal EHR data.

The results of these experiments in Table 5 show our model outperforms other generation models. Compared to the best-performing model PromptEHR with Adacare, our model achieves a 3% higher Kappa. One notable point is that to make a fair comparison between baseline models, the real data does not contain the time interval between visits. Thus, backbone methods that rely on learning time information, such as HiTANet, do not perform optimally. However, we can see that when our model provides extra time information in the synthetic dataset, HiTANet's performance greatly increases and is better than others: the AUPR and F1 of HiTANet rise by 8%. This experiment not only underscores our model's effectiveness in generating high-quality unimodal data but also demonstrates its unique capability to enrich synthetic datasets with critical temporal information, thereby offering comprehensive support for advanced predictive analytics.

4.5.4 Time Interval Prediction. While the previous experiment implicitly confirmed our model's time interval prediction capability and effectiveness for downstream risk prediction, we directly compare ours to a range of established time-series forecasting baselines in this experiment. The task is defined to use historical time stamps till T_i to predict T_{i+1} and will include V_i if the model structure allows. The selected baseline methods include the Autoregressive Integrated Moving Average (ARIMA) [5], Support Vector Regression (SVR) [2], Gradient Boosting Regression Trees (GBRT) [27],

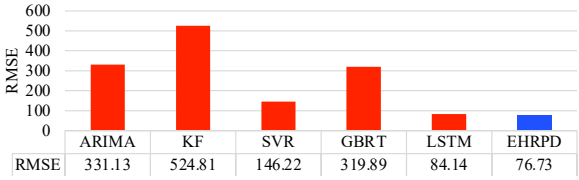

Figure 5: Illustration of time interval prediction with RMSE.

Kalman Filter (KF) [33], and LSTM [13], which are explained in Appendix Section 6.6. We use Root Mean Squared Error (RMSE) as the metric. The less the RMSE, the better the fit of the model.

We visualize the results in Figure 5, with red-colored bars representing baseline models and blue representing ours, as well as RMSE values on the bottom. We can see that timestamp-only methods do not perform well. The conventional LSTM performs closely to EHRPD, while EHRPD shows the best performance with the lowest RMSE. This experiment demonstrates our model's ability to integrate various additional information and deliver more accurate predictions on the intervals leading up to the next patient visit.

4.5.5 Severe Outcome Prediction. In the healthcare domain, datasets often have a smaller scale compared to extensive public resources like MIMIC-III, highlighting the necessity for generation models to efficiently generate with limited data. With this concern, we assess our method's performance on the Breast Cancer Trial dataset, which presents a challenging environment with relatively

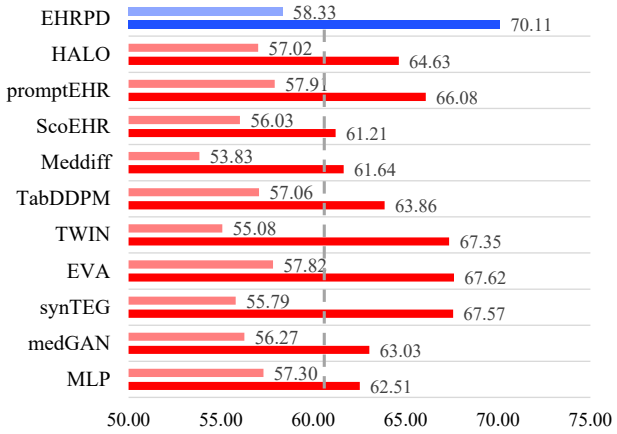

Figure 6: Illustration of severe outcome prediction from the synthetic or synthetic-real hybrid datasets.

few data entries. Following the settings in TWIN [8], our task is to predict the severe outcome and death defined in the data preprocessing section, and the selected metric is the Area Under the Receiver Operating Characteristic (AUROC) score. The size of the generated dataset is equal to the training dataset.

An LSTM network is used to learn the sequential visit-level hidden states, which are then utilized by an MLP to make a binary prediction. The results are depicted in Figure 6. The dashed line represents the AUROC value achieved using the real dataset. Our model is colored in blue, while baselines are colored in red. For each pair of the histogram, the light-colored one is the performance from synthetic data only, while the darker one is from synthetic data plus real data. Our model achieves the best performance under both synthetic-only and hybrid settings. This comparative experiment provides a clear visualization of our model's capability to generate synthetic data that is both realistic and effective for advanced predictive tasks.

4.5.6 Adverse Event Prediction. While the previous experiments show our model's generation capability on the patient level, now we evaluate the fine-grained code-level generation capability and see whether the generated visit is coherent with its predecessor. Thus, in this section, our task is to predict the next visit's adverse events with the current visit's multimodal codes. The only training datasets available are the synthetic ones with AUROC as the metric. The size of the generated dataset is equal to the real training dataset. We utilize linear layers to embed medical codes and aggregate to visit level, and then an MLP predicts the next visit's adverse events.

Our findings are visually represented in Figure 7, where the horizontal dashed line indicates the performance with the real training set. Red histograms show the performance of baseline models, while blue ones highlight that of our model. We can observe that our model is closest to the real dataset's performance, while PromptEHR achieves the second-best performance, likely due to the sequential generation nature of the model design that helps preserve the visit-to-visit consistency. However, HALO behaves worse than expected. This can be attributed to its design, which generates a single prediction vector of various modalities, diluting its effectiveness in tasks that require a focused prediction on a singular modality.

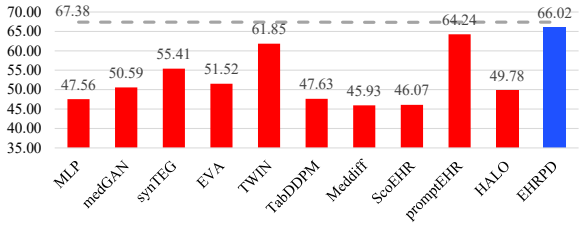

Figure 7: Illustration of adverse event prediction with synthetic datasets.

	Diagnosis		Drug		Lab Item		Procedure	
Metric	LPL	MPL	LPL	MPL	LPL	MPL	LPL	MPL
AS1	22.60	26.99	32.84	36.69	23.68	20.81	34.21	33.22
AS2	21.09	22.63	23.38	24.02	17.36	17.21	23.20	28.64
AS3	22.96	22.71	28.30	27.72	19.36	18.46	18.95	22.52
AS4	66.73	49.43	44.71	30.10	52.25	27.64	58.36	170.66
EHRPD	**15.97**	**17.95**	**20.53**	**19.15**	**15.11**	**13.99**	**14.53**	**18.89**

Table 6: Results of ablation study

4.6 Ablation Study

In this section, we remove some components of our model to assess each component's effectiveness towards the whole model, with LPL and MPL as evaluation metrics. All ablation experiments are described as follows:

- **AS 1:** removes the time aware visit embedding in Section 3.1 and replaces with a linear embedding layer.
- **AS 2:** removes the time interval estimation (Eq. 19) and time prediction loss (Eq. 20).
- **AS 3:** removes the demographic information embedding of catalyst representation in Section 3.3.
- **AS 4:** removes the self attention in Eq. (14), i.e., exclude catalyst representation entirely.

The experiment result is shown in Table 6. An analysis of the outcomes reveals that each component plays a significant role in enhancing the model's performance. Notably, the catalyst representation in EHRPD emerges as the most critical element, significantly influencing the model's performance.

5 Conclusion

In this paper, we present EHRPD, a diffusion-based EHR data generation model. By incorporating a time-aware visit embedding module and predicting the next visit with a novel predictive diffusion model, EHRPD is capable of capturing the complex temporal information of EHR data. Furthermore, EHRPD's ability to simultaneously estimate time intervals till the next visit sets it apart from existing methods, offering a significant improvement in the field of EHR data generation. To validate our claims, we conducted extensive experiments on publically available datasets, demonstrating EHRPD's superior performance from three comprehensive perspectives: utility, fidelity, and privacy.

Acknowledgements

The authors thank the anonymous referees for their valuable comments and helpful suggestions. This work is partially supported by the National Science Foundation under Grant No. 2238275 and the National Institutes of Health under Grant No. R01AG077016.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems* (2021), 17981–17993.

[2] Mariette Awad, Rahul Khanna, Mariette Awad, and Rahul Khanna. 2015. Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers* (2015), 67–80.

[3] Mrinal Kanti Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. 2019. Synthesizing electronic health records using improved generative adversarial networks. *JAMIA* (2019), 228–241.

[4] Siddharth Biswal, Soumya Ghosh, Jon Duke, Bradley Malin, Walter Stewart, Cao Xiao, and Jimeng Sun. 2021. EVA: Generating longitudinal electronic health records using conditional variational autoencoders. In *Machine Learning for Healthcare Conference*. 260–282.

[5] George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *J. Amer. Statist. Assoc.* (1970), 1509–1526.

[6] Zhengping Che, Yu Cheng, Shuangfei Zhai, Zhaonan Sun, and Yan Liu. 2017. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *IEEE ICDM*. 787–792.

[7] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems* (2016).

[8] Trisha Das, Zifeng Wang, and Jimeng Sun. 2023. Twin: Personalized clinical trial digital twin generation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 402–413.

[9] Yujuan Feng, Zhenxing Xu, Lin Gan, Ning Chen, Bin Yu, Ting Chen, and Fei Wang. 2019. Dcmn: Double core memory network for patient outcome prediction with multimodal data. In *International Conference on Data Mining*. 200–209.

[10] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *International Conference on Learning Representations*.

[11] Huan He, Shifan Zhao, Yuanzhe Xi, and Joyce C Ho. 2023. MedDiff: Generating electronic health records using accelerated denoising diffusion model. *arXiv preprint arXiv:2302.04355* (2023).

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* (2020), 6840–6851.

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

[14] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* (2016), 1–9.

[15] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*. 17564–17579.

[16] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. 2022. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing* (2022), 47–59.

[17] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems* (2022), 4328–4343.

[18] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. 2020. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 647–656.

[19] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1903–1911.

[20] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Association for the Advancement of Artificial Intelligence*. 825–832.

[21] Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems* (2017).

[22] Ahmed Ammar Naseer, Benjamin Walker, Christopher Landon, Andrew Ambrosy, Marat Fudim, Nicholas Wysham, Botros Toro, Sumanth Swaminathan, and Terry Lyons. 2023. ScoEHR: Generating Synthetic Electronic Health Records using Continuous-time Diffusion Models. In *Machine Learning for Healthcare Conference*. PMLR, 489–508.

[23] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning*. 16784–16804.

[24] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*. 8857–8868.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[26] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. 2022. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 4713–4726.

[27] Robert E Schapire. 2003. The boosting approach to machine learning: An overview. *Nonlinear estimation and classification* (2003), 149–171.

[28] Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael W Sjoding, and Jenna Wiens. 2020. Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association* (2020), 1921–1934.

[29] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems* (2021), 24804–24816.

[30] Brandon Theodorou, Cao Xiao, and Jimeng Sun. 2023. Synthesize high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. *Nature communications* (2023), 5305.

[31] Xiaochen Wang, Junyu Luo, Jiaqi Wang, Ziyi Yin, Suhan Cui, Yuan Zhong, Yaqing Wang, and Fenglong Ma. 2023. Hierarchical Pretraining on Multimodal Electronic Health Records. In *Empirical Methods in Natural Language Processing*.

[32] Zifeng Wang and Jimeng Sun. 2022. PromptEHR: Conditional Electronic Healthcare Records Generation with Prompt Learning. In *Conference on Empirical Methods in Natural Language Processing*.

[33] Greg Welch, Gary Bishop, et al. 1995. An introduction to the Kalman filter. (1995).

[34] Cao Xiao, Edward Choi, and Jimeng Sun. 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association* (2018), 1419–1428.

[35] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. 2018. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. 2565–2573.

[36] Ziqi Zhang, Chao Yan, Thomas A Lasko, Jimeng Sun, and Bradley A Malin. 2021. SynTEG: a framework for temporal structured electronic health data simulation. *Journal of the American Medical Informatics Association* (2021), 596–604.

[37] Yuan Zhong, Suhan Cui, Jiaqi Wang, Xiaochen Wang, Ziyi Yin, Yaqing Wang, Houping Xiao, Mengdi Huai, Ting Wang, and Fenglong Ma. 2024. MedDiffusion: Boosting Health Risk Prediction via Diffusion-based Data Augmentation. In *SIAM International Conference on Data Mining*.

[38] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi Nasrabadi, Evren Körpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2409–2419.

6 Appendix

6.1 Details of P-DDPM

In this section, we provide formula derivations for the theoretical foundation of P-DDPM.

6.1.1 Forward Noise Addition Process. In the forward diffusion process of P-DDPM, we gradually add noise to $\mathbf{v}_i$ according to the noise schedule β_s:

$$\begin{aligned} q(\mathbf{v}_i^{1:S}|\mathbf{v}_i^0) &= \prod_{s=1}^{S} q(\mathbf{v}_i^s|\mathbf{v}_i^{s-1}), \\ q(\mathbf{v}_i^s|\mathbf{v}_i^{s-1}) &= \mathcal{N}(\mathbf{v}_i^s; \sqrt{1-\beta_s}\mathbf{v}_i^{s-1}, \beta_s\mathbf{I}). \end{aligned} \tag{22}$$

Let $\alpha_s = 1 - \beta_s$ and $\bar{\alpha}_s = \prod_{j=1}^{s} \alpha_j$, we can re-parameterize the Gaussian step above with its mean and variance as:

$$\begin{aligned} \mathbf{v}_i^s &= \sqrt{\alpha_s}\mathbf{v}_i^{s-1} + \sqrt{1-\alpha_s}\epsilon_{s-1} \\ &= \sqrt{\alpha_s\alpha_{s-1}}\mathbf{v}_i^{s-2} + \sqrt{1-\alpha_s\alpha_{s-1}}\bar{\epsilon}_{s-2} \\ &= \cdots \\ &= \sqrt{\bar{\alpha}_s}\mathbf{v}_i^0 + \sqrt{1-\bar{\alpha}_s}\epsilon, \end{aligned} \tag{23}$$

where ϵ is the merged Gaussian noise term from $[\epsilon_1, \cdots, \epsilon_S]$ by the property of normal distribution.

6.1.2 Predictive Mapping Process. By Eq.(7), we construct a relationship between $\mathbf{v}^s_{i+1}$, $\mathbf{v}^s_i$, and $\mathbf{\Phi}_i$.

6.1.3 Backward Denoising Diffusion Process. Then in the backward diffusion process, we start with $\mathbf{v}^s_{i+1}$. We utilize Bayes Theorem to rewrite the backward diffusion step into a mixture of forward Gaussian steps as:

$$q(\mathbf{v}^{s-1}_{i+1}|\mathbf{v}^s_{i+1}, \mathbf{v}^0_{i+1}) = q(\mathbf{v}^s_{i+1}|\mathbf{v}^{s-1}_{i+1}, \mathbf{v}^0_{i+1}) \frac{q(\mathbf{v}^{s-1}_{i+1}|\mathbf{v}^0_{i+1})}{q(\mathbf{v}^s_{i+1}|\mathbf{v}^0_{i+1})}. \tag{24}$$

Then by the density function of normal distribution, the above equation is proportional to:

$$\begin{aligned}
&\propto \exp\{-\frac{1}{2}(\frac{(\mathbf{v}^s_{i+1} - \sqrt{\alpha_s}\mathbf{v}^{s-1}_{i+1})^2}{\beta_s} + \frac{(\mathbf{v}^{s-1}_{i+1} - \sqrt{\bar{\alpha}_{s-1}}\mathbf{v}^0_{i+1})^2}{1-\bar{\alpha}_{s-1}} \\
&\quad - \frac{(\mathbf{v}^s_{i+1} - \sqrt{\bar{\alpha}_s}\mathbf{v}^0_{i+1})^2}{1-\bar{\alpha}_s})\} \\
&= \exp\{-\frac{1}{2}(\frac{(\mathbf{v}^s_{i+1})^2 - 2\sqrt{\alpha_s}\mathbf{v}^s_{i+1}\mathbf{v}^{s-1}_{i+1} + \alpha_s(\mathbf{v}^{s-1}_{i+1})^2}{\beta_s} \\
&\quad + \frac{(\mathbf{v}^{s-1}_{i+1})^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{v}^0_{i+1}\mathbf{v}^{s-1}_{i+1} + \bar{\alpha}_{s-1}(\mathbf{v}^0_{i+1})^2}{1-\bar{\alpha}_{s-1}} \\
&\quad - \frac{(\mathbf{v}^s_{i+1} - \sqrt{\bar{\alpha}_s}\mathbf{v}^0_{i+1})^2}{1-\bar{\alpha}_s})\} \\
&= \exp\{-\frac{1}{2}((\frac{\alpha_s}{\beta_s} + \frac{1}{1-\bar{\alpha}_{s-1}})(\mathbf{v}^{s-1}_{i+1})^2 \\
&\quad - (\frac{2\sqrt{\alpha_s}}{\beta_s}\mathbf{v}^s_{i+1} + \frac{2\sqrt{\bar{\alpha}_{s-1}}}{1-\bar{\alpha}_{s-1}}\mathbf{v}^0_{i+1})\mathbf{v}^{s-1}_{i+1} + \text{Constant})\}
\end{aligned} \tag{25}$$

Then by inspection, we can derive the mean and variance of the above density function as:

$$\begin{aligned}
&\hat{\beta}_s = 1/(\frac{\alpha_s}{\beta_s} + \frac{1}{1-\bar{\alpha}_{s-1}}) = \frac{1-\bar{\alpha}_{s-1}}{1-\bar{\alpha}_s}\beta_s, \\
&\hat{\boldsymbol{\mu}}_s(\mathbf{v}^s_{i+1}, \mathbf{v}^0_{i+1}) \\
&= (\frac{\sqrt{\alpha_s}}{\beta_s}\mathbf{v}^s_{i+1} + \frac{\sqrt{\bar{\alpha}_{s-1}}}{1-\bar{\alpha}_{s-1}}\mathbf{v}^0_{i+1})/(\frac{\alpha_s}{\beta_s} + \frac{1}{1-\bar{\alpha}_{s-1}}) \\
&= (\frac{\sqrt{\alpha_s}}{\beta_s}\mathbf{v}^s_{i+1} + \frac{\sqrt{\bar{\alpha}_{s-1}}}{1-\bar{\alpha}_{s-1}}\mathbf{v}^0_{i+1})\frac{1-\bar{\alpha}_{s-1}}{1-\bar{\alpha}_s}\beta_s \\
&= \frac{\sqrt{\alpha_s}(1-\bar{\alpha}_{s-1})}{1-\bar{\alpha}_s}\mathbf{v}^s_{i+1} + \frac{\sqrt{\bar{\alpha}_{s-1}}\beta_s}{1-\bar{\alpha}_s}\mathbf{v}^0_{i+1}.
\end{aligned} \tag{26}$$

Substituting $\mathbf{v}^0_{i+1}$ with Eq.(6) by $\mathbf{v}^s_{i+1}$, we have:

$$\begin{aligned}
&\hat{\boldsymbol{\mu}}_s(\mathbf{v}^s_{i+1}, \mathbf{v}^0_{i+1}) \\
&= \frac{\sqrt{\alpha_s}(1-\bar{\alpha}_{s-1})}{1-\bar{\alpha}_s}\mathbf{v}^s_{i+1} + \frac{\sqrt{\bar{\alpha}_{s-1}}\beta_s}{1-\bar{\alpha}_s}\frac{1}{\sqrt{\bar{\alpha}_s}}(\mathbf{v}^s_{i+1} - \sqrt{1-\bar{\alpha}_s}\epsilon_s) \\
&= \frac{1}{\sqrt{\alpha_t}}(\mathbf{v}^s_{i+1} - \frac{1-\alpha_s}{\sqrt{1-\bar{\alpha}_t}}\epsilon_s)
\end{aligned} \tag{27}$$

And finally we have the closed-form solution that describes the cross-visit relation with Eq. (7) as follows:

$$\hat{\boldsymbol{\mu}}_s(\mathbf{v}^s_{i+1}, \mathbf{v}^0_{i+1}) = \frac{1}{\sqrt{\alpha}_s}(f(\mathbf{v}^s_i, \mathbf{\Phi}_i) - \frac{1-\alpha_s}{\sqrt{1-\bar{\alpha}_s}}\epsilon_s). \tag{28}$$

6.2 Details of PU-Net

In this section, we provide the detailed structure of our PU-Net, as in Figure 4.

6.2.1 Downsampling Path. Denoting the BatchNorm layer as BN, the downsampling path of the PU-Net utilizes Resnet Block (ResB) to refine features and downsamples with a 1-D convolutional layer as follows:

$$\begin{aligned}
\text{ResB}(\mathbf{v}^s_{i,l}) &= \text{ReLU}(\text{BN}(\text{Conv}(\mathbf{v}^s_{i,l}))) + \mathbf{v}^s_{i,l}, \\
\mathbf{v}^s_{i,l+1} &= \text{Conv1d}(\text{ResB}(\mathbf{v}^s_{i,l})).
\end{aligned} \tag{29}$$

6.2.2 Self-attention. In the skip connection, we utilize a self-attention to fuse l-th layer $\mathbf{\Phi}_{i,l}$ and $\mathbf{v}^s_{i,l}$:

$$\begin{aligned}
\bar{\mathbf{v}}^s_{i+1,l} &= \text{Softmax}\left(\frac{\mathbf{W}^Q_l(\mathbf{v}^s_{i,l}) \cdot \mathbf{W}^K_l(\mathbf{\Phi}_{i,l})}{\sqrt{d}}\right) \cdot \mathbf{W}^V_l(\mathbf{\Phi}_{i,l}), \\
\dot{\mathbf{v}}^s_{i+1,l} &= \text{MaxPooling}(\text{LayerNorm}(\mathbf{v}^s_{i,l}) + \bar{\mathbf{v}}^s_{i+1,l}),
\end{aligned} \tag{30}$$

where $\mathbf{W}^Q_l, \mathbf{W}^K_l, \mathbf{W}^V_l \in \mathbb{R}^{d_l * d_l}$.

6.2.3 Upsampling Path. With DeConv denoting the deconvolution layer, our upsampling path first upsamples the lower level feature $\mathbf{v}^s_{i+1,l+1}$ to $\ddot{\mathbf{v}}^s_{i+1,l}$:

$$\ddot{\mathbf{v}}^s_{i+1,l} = \text{ReLU}(\text{BN}(\text{DeConv1d}(\mathbf{v}^s_{i+1,l+1}))). \tag{31}$$

Then the feature from the skip connection is fused with the upsampled feature with a 1-D convolution layer as:

$$\tilde{\mathbf{v}}^s_{i+1,l} = \text{Conv1d}[\dot{\mathbf{v}}^s_{i+1,l}; \ddot{\mathbf{v}}^s_{i+1,l}]. \tag{32}$$

Lastly, we utilize a Resnet block to refine the learned feature:

$$\hat{\mathbf{v}}^s_{i+1,l} = \text{ResB}(\tilde{\mathbf{v}}^s_{i+1,l}). \tag{33}$$

6.3 Baseline EHR Generation Models

- **MLP** [13] integrates an LSTM with an MLP to learn relationships between patient visits.
- **medGAN** [3] uses a GAN to generate synthetic patient data, enhanced with an LSTM for temporal dynamics.
- **synTEG** [36] employs a Transformer for learning relationships in patient visit sequences and a Wasserstein GAN for generating EHR data sequences.
- **EVA** [4] utilizes a VAE to encode health records into latent vectors and generate synthetic records from the learned distribution.
- **TWIN** [8] combines a VAE for capturing data distribution with decoders for predicting current and next visit codes, focusing on cross-modality fusion and temporal dynamics.
- **TabDDPM** [15] generates tabular healthcare data, incorporating an LSTM for temporal learning.
- **Meddiff** [11] uses an accelerated DDPM to generate realistic synthetic EHR data, capturing temporal dependencies.
- **ScoEHR** [22] utilizes continuous-time diffusion models to generate synthetic EHR data with temporal dynamics.
- **PromptEHR** [32] uses a pre-trained BART to generate diverse longitudinal EHR data.
- **HALO** [30] uses transformer architecture to learn different modalities of EHR codes jointly.

Task	Model	F-LSTM			F-CNN			RAIM			DCMN		
	Metric	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa	AUPR	F1	Kappa
ARF	Orginal	0.9582	0.8969	0.7826	0.9550	0.8794	0.7590	0.9465	0.8698	0.7307	0.9471	0.8795	0.7439
	MLP	0.9577	0.8932	0.7810	0.9587	0.8886	0.7635	0.9494	0.8713	0.7419	0.9438	0.8756	0.7486
	medGAN	0.9518	0.8871	0.7610	0.9535	0.8873	0.7673	0.9538	0.8715	0.7408	0.9523	0.8754	0.7449
	synTEG	0.9590	0.8929	0.7722	0.9535	0.8812	0.7701	0.9445	0.8636	0.7273	0.9536	0.8871	0.7610
	EVA	0.9600	0.8980	0.7820	0.9526	0.8870	0.7623	0.9478	0.8761	0.7358	0.9530	0.8856	0.7663
	TWIN	0.9617	0.8997	0.7946	0.9537	0.8903	0.7728	0.9575	0.8820	0.7584	0.9541	0.8844	0.7629
	TabDDPM	0.9574	0.8952	0.7792	0.9567	0.8885	0.7720	0.9414	0.8706	0.7371	0.9518	0.8820	0.7559
	Meddiff	0.9542	0.8965	0.7840	0.9542	0.8837	0.7572	0.9435	0.8700	0.7222	0.9511	0.8824	0.7503
	ScoEHR	0.9559	0.9013	0.7933	0.9487	0.8719	0.7114	0.9487	0.8701	0.7260	0.9524	0.8802	0.7521
	PromptEHR	0.9530	0.8940	0.7706	0.9540	0.8797	0.7724	0.9562	0.8840	0.7443	0.9560	0.8894	0.7731
	HALO	**0.9645**	0.9020	0.7944	0.9546	0.8884	0.7657	0.9556	0.8776	0.7630	0.9542	0.8868	0.7657
	EHRPD	0.9628	**0.9031**	**0.7975**	**0.9616**	**0.8918**	**0.7737**	**0.9582**	**0.8862**	**0.7637**	**0.9562**	**0.8931**	**0.7785**
Shock	Orginal	0.8147	0.7092	0.5354	0.8110	0.7057	0.4669	0.8104	0.7455	0.5944	0.8012	0.7256	0.5726
	MLP	0.8274	0.7630	0.6298	0.8189	0.7500	0.5731	0.8101	0.7353	0.5868	0.8079	0.7345	0.5849
	medGAN	0.8379	0.7695	0.6322	0.8224	0.7579	0.5878	0.8010	0.7494	0.6005	0.8106	0.7399	0.5920
	synTEG	0.8391	0.7561	0.6155	0.8227	0.7513	0.6107	0.8262	0.7473	0.6053	0.8105	0.7449	0.5933
	EVA	0.8437	0.7638	0.6325	0.8324	0.7517	0.5934	0.8287	0.7406	0.6044	0.8192	0.7441	0.5908
	TWIN	0.8472	0.7639	0.6263	0.8382	0.7616	0.6282	0.8208	0.7374	0.5939	0.8280	0.7534	0.5968
	TabDDPM	0.8421	0.7684	0.6327	0.8247	0.7508	0.5882	0.8199	0.7477	0.6019	0.8085	0.7380	0.5824
	Meddiff	0.8437	0.7651	0.6283	0.8371	0.7651	0.6134	0.8138	0.7456	0.5943	0.8106	0.7423	0.5895
	ScoEHR	0.8477	0.7652	0.6400	0.8316	0.7638	0.6148	0.8153	0.7340	0.5826	0.8117	0.7434	0.5872
	PromptEHR	0.8427	0.7661	0.6268	0.8334	0.7672	0.6179	0.8282	0.7437	0.5938	0.8152	0.7518	0.6060
	HALO	**0.8563**	0.7709	**0.6419**	0.8253	0.7568	0.5923	0.8268	0.7538	0.6084	0.8212	0.7435	0.5926
	EHRPD	0.8507	**0.7722**	0.6353	**0.8421**	**0.7704**	**0.6369**	**0.8361**	**0.7791**	**0.6189**	**0.8376**	**0.7704**	**0.6118**

Table 7: Result evaluation via other two risk prediction tasks on multimodal EHR data.

Dataset	Modality	Metric	MLP	medGAN	synTEG	EVA	TWIN	TabDDPM	Meddiff	ScoEHR	PromptEHR	HALO	EHRPD
MIMIC-III	Diagnosis	LPL	325.55	242.30	36.87	29.62	26.28	108.79	664.54	685.17	126.23	149.66	**15.97**
		MPL	352.54	257.48	45.61	31.63	27.68	114.18	670.91	691.55	128.05	192.13	**17.95**
	Drug	LPL	553.63	403.02	83.38	43.79	40.94	179.22	936.28	934.87	167.48	166.11	**20.53**
		MPL	551.67	405.75	82.66	44.02	40.86	178.70	936.14	950.27	136.04	202.01	**19.15**
	Lab Item	LPL	168.25	77.10	26.80	20.05	17.47	54.69	413.41	432.11	107.22	322.51	**15.11**
		MPL	166.61	87.12	30.34	19.97	17.41	54.44	412.33	431.09	98.52	303.09	**13.99**
	Procedure	LPL	290.38	234.81	49.33	27.39	21.26	98.03	471.81	486.81	51.18	22.68	**14.53**
		MPL	286.53	245.28	44.00	30.49	24.26	102.72	479.96	499.89	31.13	39.04	**18.89**

Table 8: EHR data generation evaluation of different approaches on eICU dataset.

6.4 Multimodal Risk Prediction Models

Multimodal risk prediction models used in Section 4.5.2:

- **F-LSTM** [28] combines static demographic features with time-series features as input for an LSTM module.
- **F-CNN** [28] is similar to F-LSTM but with a CNN.
- **RAIM** [35] integrates attention mechanism with modality fusion and uses an LSTM for visit-wise relationship learning.
- **DCMN** [9] utilizes separate recursive learning modules for each modality with an attention mechanism.

6.5 Backbone Unimodal Risk Prediction Models

Unimodal risk prediction models used in Section 4.5.3:

- **AdaCare** [20] uses a CNN for feature extraction and GRU blocks for prediction.
- **Dipole** [19] combines a bidirectional GRU with an attention mechanism to analyze patient visit sequences.
- **HiTANet** [18] adopts a time-aware attention mechanism to capture evolving disease patterns.
- **LSTM** [13] learns the hidden state of each visit and performs risk prediction with an MLP.
- **Retain** [7] employs a reverse time attention mechanism to prioritize recent medical events.

6.6 Backbone Time Interval Prediction Models

Time interval prediction methods in Section 4.5.4:

- **ARIMA** [5] forecasts future values using past values and errors in a rolling window fashion.
- **KF** [33] estimates system states in linear dynamic systems.
- **SVR** [2] predicts continuous values by fitting a regression line within an error margin.
- **GBRT** [27] combines multiple decision trees to improve prediction accuracy through boosting.
- **LSTM** [13] outputs a single value for time prediction.

6.7 More Result on eICU Dataset and Multimodal Risk Prediction Task

Additional results on the eICU dataset and multimodal risk prediction task (Acute Respiratory Failure(ARF) and Shock) are shown in Table 8 and Table 7.