Privacy-Preserving Generative Modeling and Clinical Validation of Longitudinal Health Records for Chronic Disease

Benjamin D. Ballyk[1,2] benjamin.ballyk@eng.ox.ac.uk
Ankit Gupta[1] ankit.m.gupta17@gmail.com
Sujay Konda[1] skonda1@terpmail.umd.edu
Kavitha Subramanian[4] kavithas@stanford.edu
Chris Landon[5] chris.landon@ventura.org
Ahmed Ammar Naseer[1] aanaseer22@gmail.com
Georg Maierhofer[2,3] gam37@cam.ac.uk
Sumanth Swaminathan[1,2] sswami@vironix.ai
Vasudevan Venkateshwaran[1] vvenkate@vironix.ai

[1] *Vironix Health Inc, Austin, TX, USA*
[2] *University of Oxford, Oxford, UK*
[3] *University of Cambridge, Cambridge, UK*
[4] *Stanford University, Stanford, CA, USA*
[5] *University of Southern California, Los Angeles, CA, USA*

Abstract

Data privacy is a critical challenge in modern medical workflows as the adoption of electronic patient records has grown rapidly. Stringent data protection regulations limit access to clinical records for training and integrating machine learning models that have shown promise in improving diagnostic accuracy and personalized care outcomes. Synthetic data offers a promising alternative; however, current generative models either struggle with time-series data or lack formal privacy guaranties. In this paper, we enhance a state-of-the-art time-series generative model to better handle longitudinal clinical data while incorporating quantifiable privacy safeguards. Using real data from chronic kidney disease and ICU patients, we evaluate our method through statistical tests, a Train-on-Synthetic-Test-on-Real (TSTR) setup, and expert clinical review. Our non-private model (Augmented TimeGAN) outperforms transformer- and flow-based models on statistical metrics in several datasets, while our private model (DP-TimeGAN) maintains a mean authenticity of 0.778 on the CKD dataset, outperforming existing state-of-the-art models on the privacy-utility frontier. Both models achieve performance comparable to real data in clinician evaluations, providing robust input data necessary for developing models for complex chronic conditions without compromising data privacy.

Keywords: Time-series modeling, chronic disease, generative adversarial network

Data and Code Availability The datasets used in the study are publicly available. The time-series eICU dataset originates from the eICU Collaborative Research Database created by the Philips eICU Research Institute, and is available after required PhysioNet credentialing (Pollard et al., 2017, 2018). The chronic kidney disease dataset is pulled from the CKD-ROUTE study, which monitored patient prognosis over a three-year window (Iimori et al., 2018a,b). The code is available at https://github.com/Vironix-Science/ppehcrgen.

Institutional Review Board (IRB) No IRB approval was necessary for this project, as the data used is de-identified and publicly available.

1. Introduction

Recent advancements in the utility and breadth of machine learning (ML) have unlocked several applications for enhancing and streamlining medical workflows. In particular, risk prediction, triage, disease progression modeling, and early detection are among

© 2025 B.D. Ballyk[1,2] *et al.*

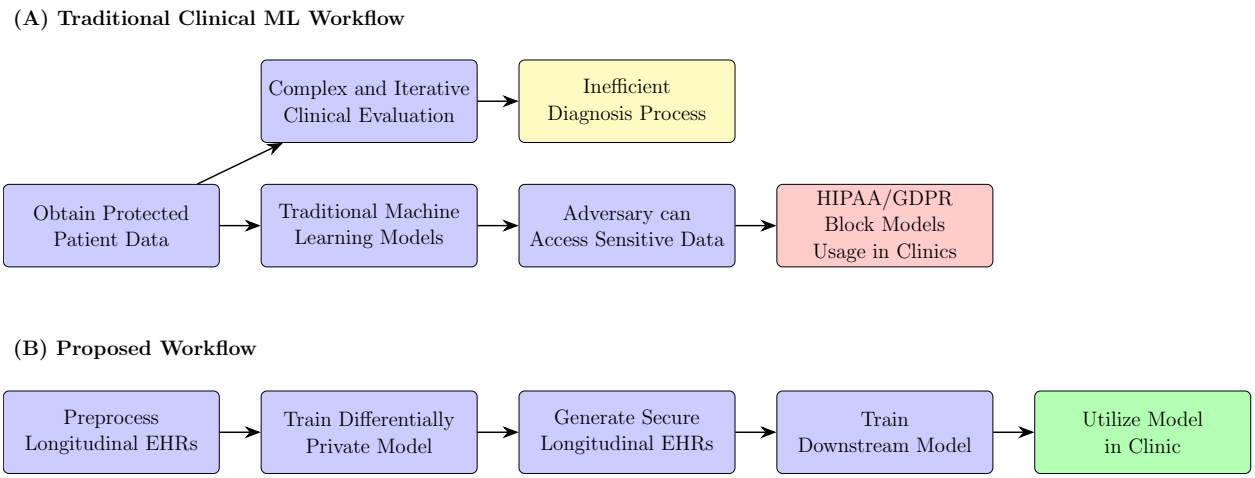


Figure 1: (A) Current workflow when handling protected patient data within the clinic. (B) Proposed downstream model pipeline for generic secure patient evaluation with machine learning models.

many clinical tasks that have proven to be conducive to ML techniques (Law et al., 2019; Mienye et al., 2020; Swaminathan et al., 2017). However, the integration of large-scale predictive and diagnostic models in clinical settings has been constrained by stringent privacy regulations.

Broad data privacy laws such as the Health Insurance Portability Accountability Act (HIPAA) in the United States and the General Data Protection Regulation (GDPR) in Europe are designed to protect medical patients from fraud and promote the adoption of electronic health records (EHRs) across medical institutions. Both HIPAA and GDPR mandate that hospitals minimize the quantity of data released, and often require explicit consent from patients prior to data disclosure (European Parliament and Council of the European Union, 2025; U.S. Health and Human Services, 2024). Consequently, procuring EHRs has become costly and time-consuming for researchers and private stakeholders. Synthetic health records have recently gained traction as an avenue to address these challenges.

The notion of synthetic data first emerged in the early 1990s as a statistical method to enable meaningful analysis without compromising individual privacy (Rubin, 1993). Unfortunately, early efforts to extend these concepts into the clinical domain were hindered by insufficient computing power, inconsistent data standards, and the low utility of synthetic outputs (Gonzales et al., 2023).

Over the past decade, advances in deep generative models and the widespread adoption of electronic health records (EHRs) have revitalized the practice of synthetic data generation (Chen et al., 2021; van Breugel et al., 2024; Ktena et al., 2024). Typically, diagnostic pipelines rely on either a complicated physical diagnosis process or ML models that use protected patient data for training. Recently, with the surge of synthetic data, the flow of clinical data begins by passing records to a generative model to be used either for training or direct modification. Then, the output may be used to train downstream ML models for medical prediction or classification tasks. (cf. Figure 1).

While deep generative models have empirically excelled at generating static snapshots of clinical information (Foraker et al., 2021), they have struggled to accomodate time-dependent (longitudinal) records necessary for the development of forward-predictive disease progression models. Moreover, the generative models available for time-series applications fail to produce quantifiable privacy controls to protect patient information.

In this paper, we introduce the Differentially Private TimeGAN (DP-TimeGAN) model, which incorporates differential privacy into the training processes of a generative adversarial network for quantifiable patient data security.

2. Related work and contributions

Generating realistic longitudinal EHRs is challenging due to their high dimensionality, long sequence

lengths, and frequent discontinuities. To address these challenges, recent work on generative models for time-series has explored several architectures with potential for synthesizing longitudinal EHRs. Temporal variational autoencoders (TimeVAEs) were designed to stabilize training, but struggle to capture the abrupt changes common in longitudinal EHRs (Desai et al., 2021; Kingma and Welling, 2022). Conversely, temporal fusion transformers (TFTs) can represent long, irregular sequences (Lim et al., 2021), but their architectural backbone infamously suffers from time scaling quadratically with sequence length, which is prohibitive for quickly generating long longitudinal records (Sommers et al., 2024; Vaswani et al., 2017). Diffusion models have been adapted to quickly produce time-series data; however, they currently produce lower-fidelity results relative to other methods (Lin et al., 2023; Sohl-Dickstein et al., 2015).

Generative adversarial networks (GANs) have shown to balance fidelity and computational efficiency, requiring relatively few parameters and generating results quickly in a single forward pass (Goodfellow et al., 2014). Our investigation begins with TimeGAN, a widely used baseline for time-series synthesis (Yoon et al., 2019), which, in its default state, does not enforce privacy guarantees. We benchmark against the recent SeriesGAN (Eskandari Nasab et al., 2024), TransFusion (Sikder et al., 2025), and TimeDiff (Tian et al., 2024) models.

Recent literature has also tested several strategies to enforce privacy requirements within deep generative models. Heuristic approaches, such as the identifiability loss in ADS-GAN, embed privacy into training, but lack reproducible, provable bounds (Yoon et al., 2020). We instead emphasize differential privacy, which introduces noise and gradient clipping during training to track and bound cumulative privacy loss (Abadi et al., 2016; Dwork et al., 2006). As a differentially private baseline, we also benchmark against Differentially Private Normalizing Flows (DP Normalizing Flows) (Lee et al., 2022).

3. Methodology

Time series health records are characterized by chronologically ordered observations $\mathbf{x}_{1:T} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where each data vector may exhibit a dependence on previous observations. Given $\{\mathbf{x}_{1:T}\}_{i=1}^{N} \sim \mathcal{D}$ for some unknown data distribution, an effective generative model must approximate $\mathcal{D}$ for sampling without purely replicating data samples. We accomplish this by augmenting a powerful recurrent generative adversarial network, and incorporating differentially private training (cf. Figure 2).

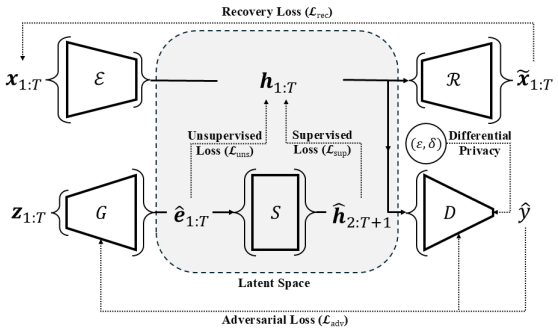


Figure 2: Model architecture for DP-TimeGAN. The model consists of five recurrent networks: embedding ($\mathcal{E}$), recovery ($\mathcal{R}$), supervisor ($\mathcal{S}$), generator (G), and discriminator (D). Real sequences $\mathbf{x}_{1:T}$ are mapped to latent space as $\mathbf{h}_{1:T} = \mathcal{E}(\mathbf{x}_{1:T})$. The generator produces latent sequences $\hat{\mathbf{e}}_{1:T} = G(\mathbf{z}_{1:T})$ from random noise, which are refined by the supervisor into supervised embeddings $\hat{\mathbf{h}}_{2:T+1} = \mathcal{S}(\hat{\mathbf{e}}_{1:T})$. The recovery network maps latent sequences back to data space, yielding $\tilde{\mathbf{x}}_{1:T} = \mathcal{R}(\mathbf{h}_{1:T})$, and the discriminator outputs $\hat{y} \in [0, 1]$ as the classification of latent sequences for adversarial training.

3.1. Time-series Generative Adversarial Networks

Our method for longitudinal EHR generation starts with the Time-series Generative Adversarial Network (TimeGAN), which comprises five recurrent neural networks (RNNs), working together to learn temporal dynamics in a latent space: the embedding ($\mathcal{E}$), recovery ($\mathcal{R}$), supervisor ($\mathcal{S}$), generator (G) and discriminator (D) networks. Real data sequences are denoted $\mathbf{x}_{1:T}$, and random noise sequences by $\mathbf{z}_{1:T}$. Synthetic data is obtained from $\mathbf{z}_{1:T}$ by passing the supervised latent sequences through the recovery network: $\tilde{\mathbf{x}}_{1:T} = \mathcal{R}(\mathcal{S}(G(\mathbf{z}_{1:T})))$.

TimeGAN is trained in three steps. Firstly, $\mathcal{E}$ and $\mathcal{R}$ are trained to compress raw data into a lower-dimensional latent space and reconstruct them back to feature space. This is achieved by minimizing the reconstruction loss, $\mathcal{L}_{\text{rec}} = \mathbb{E}\left[\|\mathbf{x}_{1:T} - \mathcal{R}(\mathcal{E}(\mathbf{x}_{1:T}))\|^2\right]$.

Next, S is trained to learn temporal dynamics in the latent space by performing next-step prediction: $\mathcal{L}_{\text{sup}} = \mathbb{E}\left[\|\mathcal{E}(\mathbf{x}_{2:T+1}) - S(\mathcal{E}(\mathbf{x}_{1:T}))\|^2\right]$. This means

S enforces temporal consistency in the latent space. Notably, supervisory loss information continues to backpropagate through $\mathcal{E}$.

Finally, the generator (G) and discriminator (D) networks are trained adversarially using the typical min-max GAN objective,

$$\mathcal{L}_{\text{adv}} = \min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)} \left[\log D(\mathbf{x}_{1:T})\right] + \mathbb{E}_{z\sim p_z(z)} \left[\log\left(1 - D(S(G(\mathbf{z}_{1:T})))\right)\right]$$

while $\mathcal{E}$ and S continue to train. The adversarial loss is also computed on unsupervised embeddings, creating the unsupervised loss ($\mathcal{L}_{\text{uns}}$). Gated recurrent units (GRUs) are used as the default RNN architecture for all internal networks (cf. Cho et al., 2014). A schematic overview of TimeGAN is shown in Figure 2.

Below, we describe two further modifications to TimeGAN's training protocol which have been experimented with to improve the generation of synthetic EHRs while stabilizing training; we refer to the version with the highest performance as the 'Augmented TimeGAN', and is used for further comparisons.

3.1.1. Discriminator Noise Injection

In practice, adversarial training in TimeGAN is often unstable as D quickly outperforms G in early training. This imbalance is particularly pronounced in EHRs due to the complex temporal dependencies and sparse observations.

To optimize generator expressivity, we inject Gaussian noise into discriminator ground-truth inputs,

$$\begin{aligned} \hat{y}_{\text{real}} &= D(\mathbf{h}_{1:T} + \mathbf{n}_{1:T}), \\ \mathbf{n}_t &\sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \quad t = 1, \ldots, T. \end{aligned} \tag{1}$$

Here, $\hat{y}_{\text{real}}$ are ground-truth discriminator outputs, $\mathbf{h}$ are real embeddings, $\mathbf{n}$ is the injected noise vector sequence, and σ is the standard deviation of noise. This process regularizes D, slows early dominance, and ensures that stronger gradient signals reach G, leading to more realistic synthetic EHR sequences.

Figure Appendix 4 shows an example of the synthetic results from training augmented TimeGAN on the sinusoidal dataset. We observe in Figures Appendix 4(a) and Appendix 4(b) that synthetic temporal paths capture the dynamics of real data, and that their distributions are roughly aligned, as evidenced by the PCA and t-SNE visualizations shown in Figures Appendix 4(c) and Appendix 4(d).

3.1.2. Extended Long Short-Term Memory Blocks

Longitudinal EHRs often contain extended sequences of observations. While GRUs in TimeGAN are effective for processing and replicating shorter sequences, their performance deteriorates on longer sequences (Bai et al., 2018). To address this limitation, we experiment with Extended Long Short-Term Memory (xLSTM) blocks in G, which capture long-range dependencies efficiently without costly autoregression (Beck et al., 2024). However, in our ablation study (Table 2), the 1:1 mLSTM : sLSTM blocks configuration did not improve generation quality, suggesting that standard GRUs suffice for the sequence lengths in our datasets. Therefore, the xLSTM block was not used in G for the Augmented TimeGAN.

3.2. Incorporating Differential Privacy

Differential privacy (DP) quantifies the extent to which a transformation mechanism releases information about individual records in a dataset. A mechanism $\mathcal{M}$ is said to be (ε, δ)-differentially private if for any two adjacent datasets, d and d',

$$P[\mathcal{M}(d) \in S] \leq e^{\varepsilon} P[\mathcal{M}(d') \in S] + \delta, \tag{2}$$

where, $S \subseteq \text{Range}(\mathcal{M})$. Here, the parameter $\varepsilon \geq 0$ limits the maximum change in the output probabilities; smaller values provide stronger privacy. The parameter δ is the tail risk, which represents the probability that the ε privacy guarantee may be violated completely (Dwork et al., 2006).

In EHR synthesis, differential privacy (DP) provides a quantifiable measure of privacy loss from real patient data. In all experiments, we set $\epsilon \in [10, 20]$ and $\delta = 10^{-5}$, which is consistent with other large-scale government and personal data releases (Table Appendix 8).

DP in machine learning relies on three core mechanisms to limit privacy leakage (Abadi et al., 2016): (i) gradient clipping, which bounds the contribution of any individual sample; (ii) noise injection, which obscures aggregated gradients to further reduce single-sample influence and facilitates privacy accounting; and (iii) random batch sampling, which selects samples independently in each batch, preventing correlations between patient records from being learned.

As training proceeds, privacy loss accumulates with each optimization step. This is tracked via a privacy accounting framework, summing per-epoch ϵ to ensure it remains below the predefined budget. In our

experiments, DP was implemented in the discriminator (D) using Opacus (Xie et al., 2018; Yousefpour et al., 2022) with Renyi differential privacy employed for accounting (Mironov, 2017). This incorporation of DP into the Augmented TimeGAN thus creates the DP-TimeGAN, selected for consistency of outputs.

4. Evaluation

Evaluating synthetic longitudinal data is challenging, as multivariate sequences do not readily lend themselves to traditional cross-sectional statistical analyses (Alaa et al., 2022; Dankar et al., 2022). To address this, we assess data quality using four key characteristics, complemented by end-use evaluations to link quantitative results to clinical relevance.

4.1. Fidelity, Diversity, and Privacy Metrics

Fidelity measures the plausibility of synthetic data relative to real patient EHRs. We measure fidelity using three metrics in our experiments. First, **Maximum mean discrepancy (MMD)** measures the distributional distance between real and synthetic data using Gaussian kernels (Xu et al., 2018). An additional measure of fidelity is **α-precision**, which is based on minimum volume sets, and evaluates the overlap between the majority of real data and the synthetic distribution, discounting outliers (Alaa et al., 2022). Finally, we also measure fidelity via a **discriminative score (DS)**, which measures the accuracy of a post-hoc GRU-based discriminator network in distinguishing real from synthetic EHRs (Yoon et al., 2019).

Diversity assesses whether synthetic data capture the full variability of the real dataset. We measure diversity with two strategies: First, the **β-recall**, the overlap between the minimum volume set of synthetic data and the full real data distribution (Alaa et al., 2022). Secondly, we use **Principal Component Analysis (PCA)** and **t-Stochastic Neighbor Embedding (t-SNE)** plots, which are dimensionality reduction techniques that allow visual assessment of distributional alignment of high-dimensional data using low-dimensional projections (Pareek and Jacob, 2021; van der Maaten and Hinton, 2008).

Privacy metrics are necessary to validate that patient data remains quantifiably secure. The **authenticity metric** evaluates the fraction of synthetic samples that are not close to any real training sample, thereby indicating the model's generalization capability and mitigating the risk of overfitting (Alaa et al., 2022).

Further details on our metrics and their mathematical formulations are included in Appendix C.

4.2. Downstream Utility

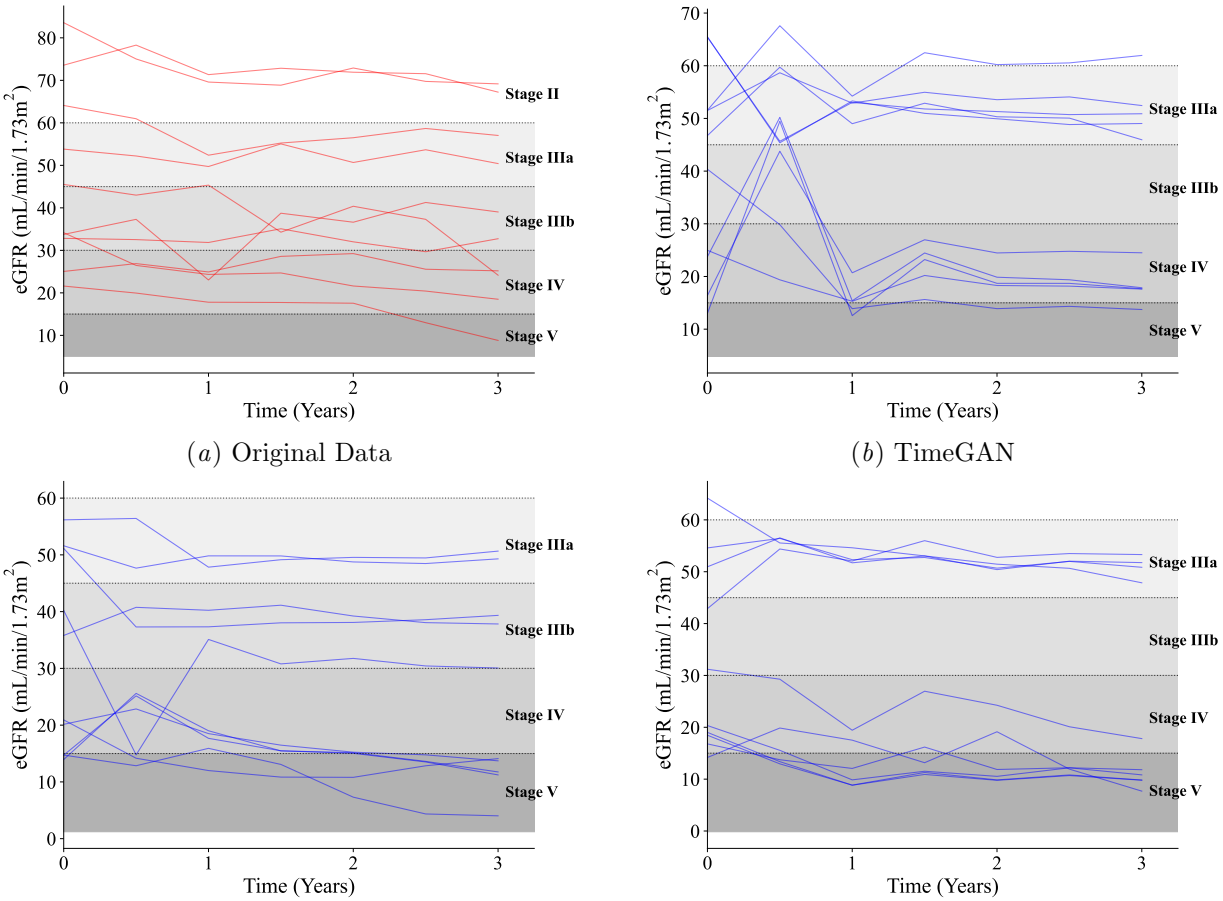
Downstream utility quantifies the practical usefulness of synthetic data in real predictive tasks, evaluated using a "Train on Synthetic, Test on Real" (TSTR) framework. We measure utility with the **predictive score (PS)**, defined as the mean absolute error of a post-hoc GRU predictor trained on synthetic data to forecast the next time step of real EHR sequences (Yoon et al., 2019).

Additionally, to assess applicability for chronic disease modeling, we also implement a **downstream classification task** using synthetic EHRs from the CKD dataset. A GRU-based classifier is trained on diabetes flags in synthetic data and evaluated on real patient data under the TSTR setup, with performance quantified by the **Area Under the Receiver Operating Characteristic curve (AUC-ROC)**. Further details are provided in Appendix C.4.

4.3. Blinded Clinician Validation

Beyond statistical similarity, synthetic longitudinal EHRs must exhibit clinically credible trajectories. For blinded validation, we randomly select 25 CKD patient profiles, comprising a mix of real data and synthetic outputs of each generative model. Each profile includes: (i) patient age and gender, (ii) baseline measurements of body mass index (BMI), hemoglobin (Hb), albumin (Alb), creatinine (Cr), and urinary protein-to-creatinine ratio (UPCR), and (iii) a three-year sequence of estimated glomerular filtration rate (eGFR) recorded every 6 months. Example profiles are shown in (Appendix D). Profiles were evaluated by five CKD specialists, who answered three evaluation questions for each profile.

From responses to question 1, we calculate two realism metrics: **Relaxed R/U**, where a sample was deemed realistic if at least one clinician labeled it realistic, and **Strict R/U**, where a sample was deemed unrealistic if at least one clinician labeled it unrealistic. Responses to questions 2–3 were aggregated into mean clinician-perceived fidelity scores. Finally, we defined the **Deception Rate** as the fraction of synthetic cases judged to be real.



(a) Original Data

(b) TimeGAN

(c) Augmented TimeGAN

(d) DP-TimeGAN

Figure 3: Real and synthetic eGFR trajectories for patients with chronic kidney disease (CKD). CKD stages are shaded in order of severity, labeled on the right. Data has shape $(N, T, C) = (421, 7, 7)$; (b), (c), and (d) use parameters: #epochs = 10000, #layers = 3, latent-dim = 24, $\gamma = 1$. For DP, $(\varepsilon, \delta) = (10, 10^{-5})$.

5. Clinical Datasets

We evaluate generative performance on three datasets. As a benchmark, we use a synthetic sine dataset where sequences are generated as $\sin(\eta t + \theta)$ with $\eta \sim \mathcal{U}[0, 0.1]$ and $\theta \sim \mathcal{U}[0, 0.1]$ (Yoon et al., 2019). For clinical data, we first construct longitudinal EHRs from the eICU Collaborative Research Database, containing time-varying vital signs from intensive care unit (ICU) patients (Pollard et al., 2017, 2018). Finally, to evaluate applicability in chronic disease, we use a chronic kidney disease (CKD) dataset with longitudinal estimated glomerular filtration rate (eGFR) trajectories, collected by Iimori et al. (2018a,b). Table 5 details the shape and makeup of each dataset.

5.1. Data Preprocessing

Each clinical dataset requires task-specific preprocessing to form consistent time-series tensor inputs for model training. For the eICU dataset, patient measurements ware resampled to a uniform interval of one observation per hour, and patients with incomplete or insufficient sequence lengths were removed. Remaining sequences were truncated or padded to a fixed length and reshaped into the standard RNN input format (Esteban et al., 2017).

For the CKD dataset, longitudinal trajectories were constructed from patient time-series measurements, reshaped for RNN compatibility, and filtered to exclude incomplete sequences. Finally, both datasets were normalized using MinMax scaling.

Table 1: Statistical metrics for synthetic data performance on the sines, eICU and CKD datasets using the following training parameters: #epochs = 7000, #layers = 3, latent-dim = 24, noise-SD = 0.2. DP runs use $\varepsilon = 15$ for sines, and $\varepsilon = 20$ for eICU and CKD, alongside $\delta = 10^{-5}$ for all runs. Benchmark models replicate the hyperparameters from their respective publications. All metrics are averaged over three training runs, and are listed as Mean ± S.D.

Model	**MMD (↓)**	**DS (↓)**	**α-precision (↑)**	**β-recall (↑)**	**Authenticity (↑)**
Sines Dataset					
Augmented TimeGAN	$\mathbf{0.002 \pm 0.002}$	$\mathbf{0.089 \pm 0.061}$	$\mathbf{0.951 \pm 0.016}$	$\mathbf{0.963 \pm 0.012}$	0.549 ± 0.019
DP-TimeGAN	0.010 ± 0.004	0.213 ± 0.056	0.929 ± 0.044	0.918 ± 0.022	0.583 ± 0.020
SeriesGAN	0.016 ± 0.010	0.203 ± 0.104	0.807 ± 0.111	0.799 ± 0.086	0.537 ± 0.069
DP Normalizing Flows	0.020 ± 0.011	0.105 ± 0.081	0.602 ± 0.106	0.506 ± 0.078	$\mathbf{0.598 \pm 0.114}$
TransFusion	0.007 ± 0.004	0.257 ± 0.070	0.862 ± 0.036	0.865 ± 0.031	0.540 ± 0.042
TimeDiff	0.018 ± 0.010	0.270 ± 0.095	0.657 ± 0.134	0.549 ± 0.093	0.576 ± 0.103
eICU Dataset					
Augmented TimeGAN	$\mathbf{0.012 \pm 0.009}$	0.053 ± 0.016	$\mathbf{0.951 \pm 0.038}$	0.941 ± 0.032	0.415 ± 0.112
DP-TimeGAN	0.019 ± 0.006	0.145 ± 0.072	0.894 ± 0.057	0.920 ± 0.029	0.581 ± 0.030
SeriesGAN	0.102 ± 0.048	0.240 ± 0.061	0.866 ± 0.073	0.788 ± 0.040	0.467 ± 0.039
DP Normalizing Flows	0.020 ± 0.013	0.167 ± 0.064	0.776 ± 0.079	0.637 ± 0.044	$\mathbf{0.684 \pm 0.068}$
TransFusion	0.014 ± 0.008	$\mathbf{0.032 \pm 0.012}$	0.942 ± 0.027	$\mathbf{0.964 \pm 0.035}$	0.574 ± 0.053
TimeDiff	0.018 ± 0.009	0.071 ± 0.030	0.698 ± 0.104	0.717 ± 0.092	0.532 ± 0.135
CKD Dataset					
Augmented TimeGAN	$\mathbf{0.049 \pm 0.013}$	$\mathbf{0.231 \pm 0.049}$	$\mathbf{0.925 \pm 0.030}$	$\mathbf{0.936 \pm 0.019}$	0.604 ± 0.083
DP-TimeGAN	0.091 ± 0.046	0.312 ± 0.053	0.844 ± 0.035	0.904 ± 0.047	$\mathbf{0.778 \pm 0.053}$
SeriesGAN	0.125 ± 0.062	0.335 ± 0.060	0.880 ± 0.021	0.840 ± 0.030	0.569 ± 0.104
DP Normalizing Flows	0.253 ± 0.055	0.323 ± 0.071	0.701 ± 0.042	0.718 ± 0.041	0.723 ± 0.062
TransFusion	0.201 ± 0.059	0.344 ± 0.082	0.327 ± 0.079	0.488 ± 0.092	0.767 ± 0.106
TimeDiff	0.073 ± 0.040	0.235 ± 0.037	0.494 ± 0.065	0.772 ± 0.101	0.703 ± 0.086

6. Results

Figure 3 shows a sampling of the real and resultant synthetic CKD progression eGFR pathways from the TimeGAN, Augmented TimeGAN, and DP-TimeGAN models. We observe that the synthetic sequences capture transitions between disease stages, even in relatively early progression of CKD, where training data is scarce. Of the three models, the Augmented TimeGAN displays the strongest preservation of the original data distribution, while the original TimeGAN model struggles to capture transitions between CKD stages. Additionally, Figure Appendix 3(d) validates the notion that the differentially private variant of the model sacrifices fidelity and diversity as compared to the augmented TimeGAN model. To quantify model performance and practicality in a clinical context, we use both statistical and clinician-evaluated measures.

6.1. Statistical Performance Measures

We first evaluate synthetic samples using the statistical metrics described in Sections 4.1 and 4.2, enabling reproducible benchmarking without requiring clinician input. Results are reported in Tables 1 and 4.

DP-TimeGAN achieves the strongest authenticity scores on both the sine and CKD datasets but is surpassed by DP Normalizing Flows on eICU authenticity, though at the cost of reduced sample quality. Conversely, DP-TimeGAN underperforms in fidelity and diversity, where Augmented TimeGAN demonstrates superior results, which demonstrates the fidelity tradeoff of differentially private training (Esteban et al., 2017). TransFusion performs comparably to Augmented TimeGAN on the eICU dataset but suffers on others. While DP Normalizing Flows outperforms Augmented TimeGAN on downstream AUC-ROC, it fails to produce a strong Predictive Score, limiting the utility of outputs for longitudinal prediction tasks. These quantitative findings only build a partial picture, and motivate further validation through blinded clinician review.

6.2. Clinical Validation

We complement statistical metrics by conducting blinded expert clinician evaluations of CKD trajec-

Table 2: Ablation study of the different unique portions of the Augmented and DP-TimeGAN on the sines dataset. TimeGAN models utilize the same parameters as mentioned in the caption of Table 1. xLSTM-specific parameters include: #heads = 4, #blocks = 4, sLSTM positions = 1, 3, 1D convolution kernel size = 4, QKV block size = 4, projection factor = 1.3 and activation function = GeLU. All metrics are calculated from three separated training runs, and are listed as Mean ± S.D.

Modifications	**MMD**	**DS**	**α-precision**	**β-recall**	**Authenticity**
Augmented TimeGAN					
None	0.008 ± 0.004	0.269 ± 0.044	0.648 ± 0.157	0.657 ± 0.157	0.531 ± 0.040
xLSTM	0.012 ± 0.011	0.289 ± 0.042	0.549 ± 0.279	0.546 ± 0.223	0.451 ± 0.095
Noise Injection	$\mathbf{0.002 \pm 0.002}$	$\mathbf{0.089 \pm 0.061}$	$\mathbf{0.951 \pm 0.016}$	$\mathbf{0.963 \pm 0.012}$	$\mathbf{0.549 \pm 0.019}$
xLSTM & Noise Injection	0.009 ± 0.011	0.290 ± 0.053	0.856 ± 0.025	0.916 ± 0.049	0.495 ± 0.119
DP-TimeGAN					
None	0.015 ± 0.010	0.237 ± 0.154	0.664 ± 0.233	0.837 ± 0.096	$\mathbf{0.682 \pm 0.124}$
xLSTM	0.012 ± 0.002	0.304 ± 0.088	0.897 ± 0.044	0.837 ± 0.074	0.546 ± 0.128
Noise Injection	$\mathbf{0.010 \pm 0.004}$	$\mathbf{0.213 \pm 0.056}$	$\mathbf{0.929 \pm 0.044}$	$\mathbf{0.918 \pm 0.022}$	0.583 ± 0.020
xLSTM & Noise Injection	0.020 ± 0.010	0.257 ± 0.056	0.704 ± 0.195	0.817 ± 0.213	0.629 ± 0.034

Table 3: Clinician validation of 25 random patient longitudinal EHRs from the real dataset and the models. The TimeGAN models utilize the same parameters as mentioned in Table 1.

Model	**Relaxed R/U**	**Strict R/U**	**Q2 Mean**	**Q3 Mean**	**Deception Rate**
Real Data	0.857	0.143	3.286	3.036	-
Regular TimeGAN	1.000	0.250	3.813	3.438	0.750
Augmeted TimeGAN	1.000	0.800	3.700	3.250	**0.960**
DP-TimeGAN	1.000	1.000	4.354	4.229	0.950

Table 4: Utility metrics for generative models trained on the CKD dataset. TimeGAN-based models use the parameters shown in Table 1. Benchmark models replicate the parameters from their respective publications. All metrics are averaged over three training runs, and are listed as Mean ± S.D.

Model	**Predictive Score**	**Downstream AUC-ROC**
Real Data	0.289 ± 0.016	0.730 ± 0.117
Augmented TimeGAN	0.381 ± 0.050	0.615 ± 0.046
DP-TimeGAN	$\mathbf{0.370 \pm 0.033}$	0.549 ± 0.063
SeriesGAN	0.448 ± 0.052	0.535 ± 0.050
DP Normalizing Flows	0.497 ± 0.062	$\mathbf{0.648 \pm 0.008}$
TransFusion	0.395 ± 0.051	0.450 ± 0.104
TimeDiff	0.565 ± 0.122	0.540 ± 0.140

tories (see Section 4.3). Using the Relaxed R/U criterion, 86% of real patients and 100% of synthetic patients were judged realistic. Under the Strict R/U criterion, 14% of real patients and 50% of synthetic patients were labeled realistic. We learn that high-fidelity synthetic records at are practically imperceptable at a high level, even to a subset of expert clinicians. A detailed breakdown by model is provided in Table 3. Overall, results suggest that synthetic data achieves parity with real CKD trajectories in the most conservative evaluation setting.

7. Discussion

DP-TimeGAN provides a secure framework for generating synthetic longitudinal EHRs, offering stronger privacy protection than baseline models, as evidenced by improved authenticity scores. The model also produces clinically useful data, as demonstrated by its AUC-ROC performance on diabetes prediction in CKD patients and by clinician assessments rating its trajectories as realistic. To our knowledge, this is the first approach to combine formal privacy guarantees with demonstrated clinical realism and downstream utility in chronic disease EHRs.

The choice to incorporate formal privacy-preserving mechanisms supports compliance with HIPAA and GDPR, enabling a safer integration of ML tools into clinical workflows. Furthermore,

we see this strategy as an opportunity to expand data accessibility for CKD research and pave the way toward broader applications in chronic disease modeling.

Limitations We acknowledge several limitations that should be overcome to generalize our results into broader applicability. First, the CKD dataset contains a limited set of features measured over a fixed three-year window, where only eGFR is longitudinal, while all other measurements are held static at the initial observation. This constrains the diversity and expressivity of CKD trajectories available for training, and was highlighted by clinicians as a drawback during expert classification.

Moreover, we draw attention to the inherent privacy–utility trade-off introduced with differential privacy. As shown in Appendix F, DP-TimeGAN sacrifices some data quality to enforce quantifiable privacy guarantees, which may reduce its effectiveness for downstream tasks. The choice of privacy budget remains underexplored, and further work is needed to systematically characterize trade-offs among privacy, utility, and computational cost.

Finally, we pose as future work that model performance may be improved by incorporating alternative DP mechanisms, architectural refinements such as state-space model (SSM) backbones, or training stabilization techniques including spectral normalization or a Wasserstein loss (Gulrajani et al., 2017; Miyato et al., 2018; Zhang et al., 2023).

8. Conclusion

We introduce DP-TimeGAN, a privacy-preserving generative model for synthesizing longitudinal electronic health records (EHRs) in chronic disease contexts. DP-TimeGAN demonstrates stronger formal privacy guarantees than baseline models while maintaining near-state-of-the-art performance on downstream predictive tasks and blinded clinical evaluations for CKD. Clinicians rate both Augmented TimeGAN and DP-TimeGAN outputs as clinically realistic, aligning with patterns observed in practice.

When combined with disease progression models, DP-TimeGAN has direct applications in disease modeling research, mitigating barriers to longitudinal EHR access, accelerating medical software testing, and informing healthcare delivery economics.

Acknowledgments

We gratefully acknowledge the essential contributions of Vironix Health, colleagues at Stanford University, and the Nephrology Division at Unity Health Toronto, whose collective expertise provided the clinical foundation and expert data evaluation within this paper.

We thank the following clinicians for providing the review discussed in Section 6.2: Dr. Indira Chevru (Georgia Kidney Associates, Marietta, GA), Dr. Vartan Papazian (Modern Nephrology, Antelope Valley, CA), and Dr. Nicholas Wysham (Vancouver Clinic, Vancouver, WA).

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, Vienna Austria, October 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978318. URL https://dl.acm.org/doi/10.1145/2976749.2978318.

Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models. In *Proceedings of the 39th International Conference on Machine Learning*, pages 290–306. PMLR, June 2022. URL https://proceedings.mlr.press/v162/alaa22a.html. ISSN: 2640-3498.

Apple Machine Learning Research. Learning with privacy at scale. https://machinelearning.apple.com/research/learning-with-privacy-at-scale, 2021. Accessed: 2024-08-24.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended Long Short-Term Memory, May 2024.

URL http://arxiv.org/abs/2405.04517. arXiv:2405.04517.

Richard J. Chen, Ming Y. Lu, Tiffany Y. Chen, Drew F. K. Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497, June 2021. ISSN 2157-846X. doi: 10.1038/s41551-021-00751-8. URL https://www.nature.com/articles/s41551-021-00751-8.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, September 2014. URL http://arxiv.org/abs/1406.1078. arXiv:1406.1078 [cs].

Fida K. Dankar, Mahmoud K. Ibrahim, and Leila Ismail. A Multi-Dimensional Evaluation of Synthetic Data Generators. *IEEE Access*, 10:11147–11158, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3144765. URL https://ieeexplore.ieee.org/document/9686689/.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. TimeVAE: A Variational Auto-Encoder for Multivariate Time Series Generation, December 2021. URL http://arxiv.org/abs/2111.08095. arXiv:2111.08095 [cs].

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004, pages 486–503. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-34546-6 978-3-540-34547-3. doi: 10.1007/11761679_29. URL http://link.springer.com/10.1007/11761679_29. Series Title: Lecture Notes in Computer Science.

Mohammad Reza Eskandari Nasab, Shah Muhammad Hamdi, and Soukaina Filali Boubrahimi. SeriesGAN: Time Series Generation via Adversarial and Autoregressive Learning. In *2024 IEEE International Conference on Big Data (BigData)*, pages 860–869, Washington, DC, USA, December 2024. IEEE. ISBN 979-8-3503-6248-0. doi: 10.1109/BigData62323.2024.10825115. URL https://ieeexplore.ieee.org/document/10825115/.

Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs, December 2017. URL http://arxiv.org/abs/1706.02633. arXiv:1706.02633.

European Parliament and Council of the European Union. General Data Protection Regulation: Principles Relating to Processing of Personal Data, April 2025. URL https://www.legislation.gov.uk/eur/2016/679/article/5.

Facebook Research. Protecting privacy in facebook mobility data during the covid-19 response. https://tinyurl.com/4u4rhueu, 2020. Accessed: 2024-08-24.

Randi E Foraker, Sean C Yu, Aditi Gupta, Andrew P Michelson, Jose A Pineda Soto, Ryan Colvin, Francis Loh, Marin H Kollef, Thomas Maddox, Bradley Evanoff, Hovav Dror, Noa Zamstein, Albert M Lai, and Philip R O Payne. Spot the difference: comparing results of analyses from real patient data and synthetic derivatives. *JAMIA Open*, 3(4):557–566, February 2021. ISSN 2574-2531. doi: 10.1093/jamiaopen/ooaa060. URL https://academic.oup.com/jamiaopen/article/3/4/557/6032922.

Aldren Gonzales, Guruprabha Guruswamy, and Scott R. Smith. Synthetic data in health care: A narrative review. *PLOS Digital Health*, 2(1):e0000082, January 2023. ISSN 2767-3170. doi: 10.1371/journal.pdig.0000082. URL https://dx.plos.org/10.1371/journal.pdig.0000082.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014. URL http://arxiv.org/abs/1406.2661. arXiv:1406.2661 [stat].

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs, December 2017. URL http://arxiv.org/abs/1704.00028. arXiv:1704.00028 [cs].

Soichiro Iimori, Shotaro Naito, Yumi Noda, Hidehiko Sato, Naohiro Nomura, Eisei Sohara, Tomokazu Okado, Sei Sasaki, Shinichi Uchida, and Tatemitsu Rai. Data from: Prognosis of chronic kidney disease with normal-range proteinuria: The CKD-ROUTE study, December

2018a. URL https://datadryad.org/stash/dataset/doi:10.5061/dryad.kq23s. Artwork Size: 439534 bytes Pages: 439534 bytes.

Soichiro Iimori, Shotaro Naito, Yumi Noda, Hidehiko Sato, Naohiro Nomura, Eisei Sohara, Tomokazu Okado, Sei Sasaki, Shinichi Uchida, and Tatemitsu Rai. Prognosis of chronic kidney disease with normal-range proteinuria: The CKD-ROUTE study. *PLOS ONE*, 13(1):e0190493, January 2018b. ISSN 1932-6203. doi: 10.1371/journal.pone.0190493. URL https://dx.plos.org/10.1371/journal.pone.0190493.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic data–what, why and how? *arXiv preprint arXiv:2205.03257*, 2022.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022. URL http://arxiv.org/abs/1312.6114. arXiv:1312.6114 [stat].

Ira Ktena, Olivia Wiles, Isabela Albuquerque, Sylvestre-Alvise Rebuffi, Ryutaro Tanno, Abhijit Guha Roy, Shekoofeh Azizi, Danielle Belgrave, Pushmeet Kohli, Taylan Cemgil, Alan Karthikesalingam, and Sven Gowal. Generative models improve fairness of medical classifiers under distribution shifts. *Nature Medicine*, 30(4):1166–1173, April 2024. ISSN 1078-8956, 1546-170X. doi: 10.1038/s41591-024-02838-6. URL https://www.nature.com/articles/s41591-024-02838-6.

Marco Tk Law, Anthony L Traboulsee, David Kb Li, Robert L Carruthers, Mark S Freedman, Shannon H Kolind, and Roger Tam. Machine learning in secondary progressive multiple sclerosis: an improved predictive model for short-term disability progression. *Multiple Sclerosis Journal - Experimental, Translational and Clinical*, 5(4):2055217319885983, October 2019. ISSN 2055-2173, 2055-2173. doi: 10.1177/2055217319885983. URL https://journals.sagepub.com/doi/10.1177/2055217319885983.

Jaewoo Lee, Minjung Kim, Yonghyun Jeong, and Youngmin Ro. Differentially Private Normalizing Flows for Synthetic Tabular Data Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7345–7353, June 2022. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v36i7.20697. URL https://ojs.aaai.org/index.php/AAAI/article/view/20697.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion Models for Time Series Applications: A Survey, May 2023. URL http://arxiv.org/abs/2305.00624. arXiv:2305.00624 [cs].

Ibomoiye Domor Mienye, Yanxia Sun, and Zenghui Wang. An improved ensemble learning approach for the prediction of heart disease risk. *Informatics in Medicine Unlocked*, 20:100402, 2020. ISSN 23529148. doi: 10.1016/j.imu.2020.100402. URL https://linkinghub.elsevier.com/retrieve/pii/S2352914820304184.

Ilya Mironov. Renyi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, August 2017. doi: 10.1109/CSF.2017.11. URL http://arxiv.org/abs/1702.07476. arXiv:1702.07476 [cs].

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks, February 2018. URL http://arxiv.org/abs/1802.05957. arXiv:1802.05957 [cs].

Joseph P Near, David Darais, Naomi Lefkovitz, and Gary S Howarth. Guidelines for evaluating differential privacy guarantees. Technical Report NIST SP 800-226, National Institute of Standards and Technology (U.S.), Gaithersburg, MD, March 2025. URL https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-226.pdf.

Jyoti Pareek and Joel Jacob. Data Compression and Visualization Using PCA and T-SNE. In Vishal Goar, Manoj Kuri, Rajesh Kumar, and Tomonobu Senjyu, editors, *Advances in Information Communication Technology and Computing*, volume 135, pages 327–337. Springer Singapore, Singapore, 2021. ISBN 978-981-15-5420-9 978-981-15-5421-6. doi: 10.1007/978-981-15-5421-6_34. URL http://link.springer.com/10.1007/

978-981-15-5421-6_34. Series Title: Lecture Notes in Networks and Systems.

Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1): 180178, September 2018. ISSN 2052-4463. doi: 10.1038/sdata.2018.178. URL https://www.nature.com/articles/sdata2018178.

Tom Joseph Pollard, Alistair Edward William Johnson, Jesse Raffa, and Omar Badawi. The eICU Collaborative Research Database, 2017. URL https://physionet.org/content/eicu-crd/.

Ryan Rogers, Subbu Subramaniam, Sean Peng, David Durfee, Seunghyun Lee, Santosh Kumar Kancha, Shraddha Sahay, and Parvez Ahammad. Linkedin's audience engagements api: A privacy preserving data analytics system at scale. *arXiv preprint arXiv:2002.05839*, 2020.

Donald B Rubin. Statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468, 1993.

Md Fahim Sikder, Resmi Ramachandranpillai, and Fredrik Heintz. TransFusion: Generating long, high fidelity time series using diffusion models with transformers. *Machine Learning with Applications*, 20:100652, June 2025. ISSN 26668270. doi: 10.1016/j.mlwa.2025.100652. URL https://linkinghub.elsevier.com/retrieve/pii/S2666827025000350.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015. URL http://arxiv.org/abs/1503.03585. arXiv:1503.03585 [cs].

Alexander Sommers, Logan Cummins, Sudip Mittal, Shahram Rahimi, Maria Seale, Joseph Jaboure, and Thomas Arnold. A Survey of Transformer Enabled Time Series Synthesis. In *2024 IEEE 10th International Conference on Collaboration and Internet Computing (CIC)*, pages 60–69, Washington, DC, USA, October 2024. IEEE. ISBN 979-8-3503-8670-7. doi: 10.1109/CIC62241.2024.00018. URL https://ieeexplore.ieee.org/document/10835781/.

Sumanth Swaminathan, Klajdi Qirko, Ted Smith, Ethan Corcoran, Nicholas G. Wysham, Gaurav Bazaz, George Kappel, and Anthony N. Gerber. A machine learning approach to triaging patients with chronic obstructive pulmonary disease. *PLOS ONE*, 12(11):e0188532, November 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0188532. URL https://dx.plos.org/10.1371/journal.pone.0188532.

Muhang Tian, Bernie Chen, Allan Guo, Shiyi Jiang, and Anru R Zhang. Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models. *Journal of the American Medical Informatics Association*, 31(11):2529–2539, November 2024. ISSN 1067-5027, 1527-974X. doi: 10.1093/jamia/ocae229. URL https://academic.oup.com/jamia/article/31/11/2529/7747780.

U.S. Census Bureau. 2020 census key parameters announced. https://www.census.gov/newsroom/press-releases/2021/2020-census-key-parameters.html, 2021. Accessed: 2024-08-24.

U.S. Health and Human Services. The HIPAA Privacy Rule, September 2024. URL https://www.hhs.gov/hipaa/for-professionals/privacy/index.html#:~:text=The%20HIPAA%20Privacy%20Rule%20establishes,care%20providers%20that%20conduct%20certain.

Boris van Breugel, Tennison Liu, Dino Oglic, and Mihaela van der Schaar. Synthetic data in biomedicine via generative artificial intelligence. *Nature Reviews Bioengineering*, 2(12):991–1004, October 2024. ISSN 2731-6092. doi: 10.1038/s44222-024-00245-7. URL https://www.nature.com/articles/s44222-024-00245-7.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, June 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially Private Generative Ad-

versarial Network, 2018. URL https://arxiv.org/abs/1802.06739. Version Number: 1.

Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks, August 2018. URL http://arxiv.org/abs/1806.07755. arXiv:1806.07755 [cs].

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html.

Jinsung Yoon, Lydia N. Drumright, and Mihaela van der Schaar. Anonymization Through Data Synthesis Using Generative Adversarial Networks (ADS-GAN). *IEEE Journal of Biomedical and Health Informatics*, 24(8):2378–2388, August 2020. ISSN 2168-2208. doi: 10.1109/JBHI.2020.2980262. URL https://ieeexplore.ieee.org/document/9034117. Conference Name: IEEE Journal of Biomedical and Health Informatics.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-Friendly Differential Privacy Library in PyTorch, August 2022. URL http://arxiv.org/abs/2109.12298. arXiv:2109.12298 [cs].

Michael Zhang, Khaled K. Saab, Michael Poli, Tri Dao, Karan Goel, and Christopher Ré. Effectively Modeling Time Series with Simple Discrete State Spaces, 2023. URL https://arxiv.org/abs/2303.09489. Version Number: 1.

Appendix A. Augmented TimeGAN Ablation Study

To measure the effects of each modification introduced into the Augmented TimeGAN, we performed an ablation study with the sines dataset, as shown in Table 2. Figure 4 also illustrates the capabilities of Augmented TimeGAN on the sine dataset. The Augmented TimeGAN version that had the best performance was including the noise injection only. Furthermore, the Augmented TimeGAN outperformed the original TimeGAN on the eICU and CKD datasets, as shown in Tables 6 and 7. Therefore, when creating DP-TimeGAN, we chose to only integrate the noise injection for the best possible performance before integrating differential privacy.

Appendix B. Differential Privacy in Data Releases

To justify our (ϵ, δ) choices when training DP-TimeGAN, we refer to Table 8. Maintaining $\epsilon \in [10, 20]$ and $\delta = 10^{-5}$ fits within the various data releases conducted in governmental and private agencies, meaning that they are acceptable values for DP-TimeGAN experiments. These data releases are also the best precedent for DP parameters that would be legally allowed, as the best method to choose ϵ is not clear and is still an open question based on government reports. Furthermore, a typical requirement is that $\delta < \frac{1}{n}$, for a dataset containing n patients. This is maintained as per the values in Table 5 (Near et al., 2025).

Appendix C. Metric Calculation Details

C.1. Fidelity Metrics

For fidelity metrics, we consider the (i) maximum mean discrepancy, (ii) discriminative score, and (iii) α-precision, each summarized below.

Maximum mean discrepancy (MMD) is a multivariate distributional distance metric that transforms data using the Gaussian kernel:

$$K(x, y) = \exp\left\{-\frac{\|x - y\|^2}{2\sigma^2}\right\}$$

to compute a distance between two distributions. Here, σ is a length scale parameter that we take to equal 1 in our tests. For real and synthetic datasets, $\mathbf{x}$ and $\hat{\mathbf{x}}$ consisting of N and M sequences, respectively, we first obtain cross-sectional latent codes.

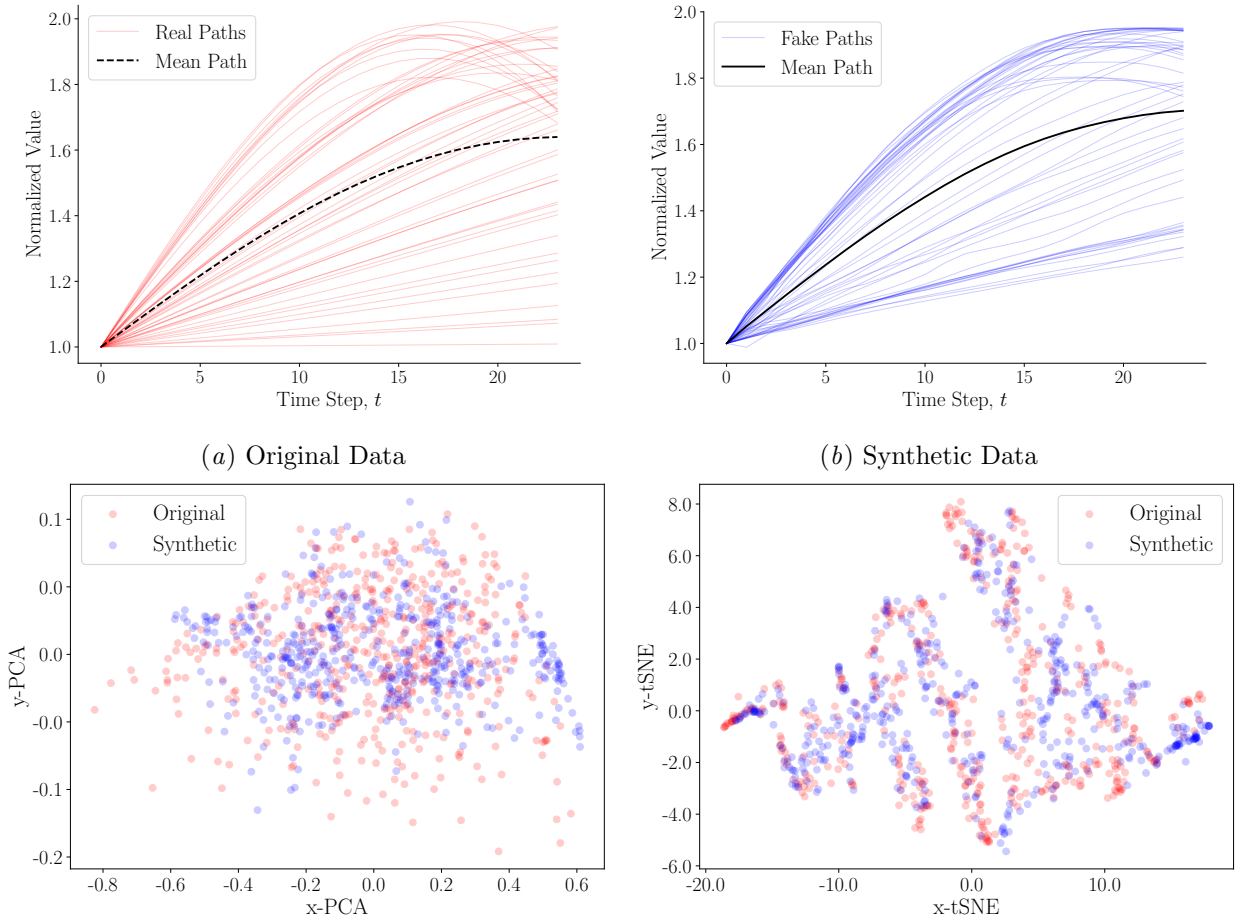
$$\mathbf{h}^{(n)} = \mathcal{E}(\mathbf{x}^{(n)}) \qquad \text{for } n = 1, \ldots, N,$$
$$\hat{\mathbf{h}}^{(n)} = \mathcal{E}(\hat{\mathbf{x}}^{(n)}), \qquad \text{for } n = 1, \ldots, M.$$

Using the cross-sectional data from the above two equations, the MMD is calculated as:

Table 5: Feature choices for datasets being used in DP-TimeGAN evaluation.

Dataset	**Feature Choices**	**Dataset shape***
Sines dataset (Yoon et al., 2019)	Synthetically generated sine waves	(500, 24, 5)
eICU Collaborative Research Database (Pollard et al., 2017)	Body temperature, oxygen saturation, heart rate, mean blood pressure, respiration rate	(750, 12, 5)
Chronic kidney disease dataset (Iimori et al., 2018a)	Age, body mass index, estimated glomerular filtration rate (eGFR), albumin, hemoglobin, creatinine, urine protein-to-creatinine ratio	(421, 7, 7)

**Shapes are taken after cleaning.*



(a) Original Data

(b) Synthetic Data

(c) PCA visualization

(d) t-SNE visualization

Figure 4: Comparison of real and synthetic sinusoidal data from the Augmented TimeGAN. Real data uses $(N, T, C) = (700, 24, 5)$, where each feature is a randomly generated sine wave; training parameters are: #epochs = 6000, #layers = 3 and latent-dim = 24. All data is normalized to a starting value of 1 prior to plotting for clarity. Plots (a) and (b) isolate one feature of real and synthetic sine waves, respectively; plots (c) and (d) compare the PCA and t-SNE results, respectively, for the two datasets.

Table 6: Regular and Augmented TimeGAN statistical metrics for synthetic data performance on the eICU and CKD datasets. TimeGAN models utilize the same parameters as mentioned in Table 1. All metrics are calculated from three separated training runs, and are listed as Mean ± S.D.

Model	**MMD**	**DS**	**α-precision**	**β-recall**	**Authenticity**
eICU Dataset					
Original TimeGAN	0.161 ± 0.130	0.153 ± 0.096	0.792 ± 0.078	0.746 ± 0.016	$\mathbf{0.446 \pm 0.104}$
Augmented TimeGAN	$\mathbf{0.012 \pm 0.009}$	$\mathbf{0.053 \pm 0.016}$	$\mathbf{0.951 \pm 0.038}$	$\mathbf{0.941 \pm 0.032}$	0.415 ± 0.112
CKD Dataset					
Original TimeGAN	0.061 ± 0.015	0.341 ± 0.112	0.848 ± 0.028	0.841 ± 0.020	$\mathbf{0.613 \pm 0.083}$
Augmented TimeGAN	$\mathbf{0.049 \pm 0.013}$	$\mathbf{0.231 \pm 0.049}$	$\mathbf{0.925 \pm 0.030}$	$\mathbf{0.936 \pm 0.019}$	0.604 ± 0.083

Table 7: Utility metrics for Regular and Augmented TimeGAN trained on the CKD dataset. TimeGAN models utilize the same parameters as mentioned in Table 1. All metrics are calculated from three separated training runs, and are listed as Mean ± S.D.

Model	**Predictive Score**	**Downstream AUC-ROC**
Original TimeGAN	0.443 ± 0.048	0.564 ± 0.052
Augmented TimeGAN	$\mathbf{0.381 \pm 0.050}$	$\mathbf{0.615 \pm 0.046}$

$$\begin{aligned}\mathrm{MMD}^2(\mathbf{h}, \hat{\mathbf{h}}) = \mathbb{E}_{\mathbf{h},\mathbf{h}'}\left[K(\mathbf{h}, \mathbf{h}')\right] \\ + \mathbb{E}_{\hat{\mathbf{h}},\hat{\mathbf{h}}'}\left[K(\hat{\mathbf{h}}, \hat{\mathbf{h}}')\right] \\ - 2\mathbb{E}_{\mathbf{h},\hat{\mathbf{h}}}\left[K(\mathbf{h}, \hat{\mathbf{h}})\right]\end{aligned}$$

where abstractly, $\mathbb{E}_{\mathbf{u},\mathbf{v}}[K(\mathbf{u}, \mathbf{v})]$ represents the expectation of the Gaussian kernel K over all vector pairs $(\mathbf{u}, \mathbf{v})$ sampled from the respective datasets; $\mathbf{h}'$ and $\hat{\mathbf{h}}'$ are used when the comparison is occuring within the same dataset. We choose this metric because it is directly applicable to multivariate distributions, and does not require kernel density estimation, enabling superior reproducibility.

Discriminative score trains a separate recurrent neural network to classify real and synthetic sequences. The training dataset is composed of a sample of entries from the real and synthetic data which have been labeled according to their validity. After the classifier has been trained for a fixed number of epochs, the model is tested on an unseen sample of data, and the discriminative score is calculated as

$$\mathrm{DS} = \left| 0.5 - \frac{N_{correct}}{N_{total}} \right|$$

Where N_{total} is the total number of sequences in the testing dataset, and $N_{correct}$ is the number of sequences correctly classified by the recurrent classifier. Intuitively, discriminative score measures the extent to which the classifier's accuracy matches that of a random guess (lower is better from perspective of generated data fidelity). For additional details on this metric, please see Yoon et al. (2019).

α-precision measures the probability that a sample from the synthetic data resides within the α-support of the real data distribution.

$$P_\alpha \triangleq \mathbb{P}(\hat{\mathbf{x}}^{(i)} \in \mathcal{S}_r^\alpha), \text{ for } \alpha \in [0, 1].$$

where $\mathcal{S}_r$ is the distribution of the real data. For additional details on this metric, please see Alaa et al. (2022).

C.2. Diversity Metrics

Principal Component Analysis (PCA) is a synthetic data visualization tool that is produced by first centering each real and synthetic sequences about their temporal means, then aggregating the centered vectors, $\mathbf{c}_t \in \mathbb{R}^n$, into a matrix, $\mathbf{C} \in \mathbb{R}^{T \times n}$:

$$\mathbf{c}_t = \mathbf{x}_t - \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t, \quad \text{for } t = 1, \ldots, T. \tag{3}$$

$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T]^\top, \tag{4}$$

Finally computing a singular value decomposition of $\mathbf{C}$, we may use the first two right singular vectors

Table 8: (ε, δ) values for differential privacy in various data releases.

Use Case	Data Released	(ε, δ) Values
US Census Bureau (2020 Census) (U.S. Census Bureau, 2021)	Population data, demographic characteristics	(19.6, 1e-5)
Meta (Pandemic Motility) (Facebook Research, 2020)	User motility data during COVID-19 pandemic	(2.0, 0)
Apple (QuickType) (Apple Machine Learning Research, 2021)	User vocabulary on iOS keyboards	(4.0, 0)
LinkedIn (Audience Engagement) (Rogers et al., 2020)	User activity and content engagement trends	(0.15, 1e-10)

to project the centralized data:

$$\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{5}$$

$$\mathbf{C}_{\text{PCA}} = \mathbf{C}\left[\mathbf{v}_1 \,|\, \mathbf{v}_2\right]. \tag{6}$$

Here, $\mathbf{v}_1$, and $\mathbf{v}_2$ are the first two right singular vectors, extracted from $\mathbf{V}$. The result, $\mathbf{C}_{\text{PCA}} \in \mathbb{R}^{n\times 2}$, has rows which indicate (x, y) coordinates in the projected space, that may be used for 2D visualization. We perform (3)-(6) to each sequence in $\{\mathbf{x}_{1:T}^{(i)}\}_{i=1}^N$ and $\{\hat{\mathbf{x}}_{1:T}^{(i)}\}_{i=1}^M$, visualizing both results to highlight whether the span and clusters in the datasets are in alignment, from which we draw insights about diversity.

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique that works by converting pairwise Euclidean distances between high-dimensional points into conditional probabilities representing similarities. The algorithm then finds a lower dimensional embedding that best preserves these similarities using a gradient descent method. The result in our case is a two-dimensional map where points that were nearby in the original high-dimensional space remain close, making it an effective tool for visualizing complex local structures such as clusters in synthetic and real datasets. For additional details on this metric, including the mathematical formulation, please see van der Maaten and Hinton (2008).

β-recall measures the probability that a sample from the real data resides within the β-support of the synthetic data distribution.

$$R_\beta \triangleq \mathbb{P}(\mathbf{x}^{(i)} \in \mathcal{S}_g^\beta), \text{ for } \beta \in [0, 1].$$

Table 9: Hyperparameters for the downstream GRU classifier

Hyperparameter	Value
Data normalization	Standard scaler
Train-test split	60:40
Batch size	32
Optimizer	Adam
Learning rate	10^{-3}
Hidden dimension	32
Epochs	1500

where $\mathcal{S}_g$ is the distribution of the synthetic data. For additional details on this metric, please see Alaa et al. (2022).

C.3. Privacy Metric

Authenticity measures the probability of a generative model synthesizing unique samples rather than copies of training data that are slightly shifted.

$$\mathbb{P}_g = A \cdot \mathbb{P}'_g + (1 - A) \cdot \delta_{g,\epsilon}$$

where $\delta_{g,e} = \delta_g * \mathcal{N}(0, \epsilon^2)$ and δ_g is a specified probability mass function for the training data. For additional details on this metric, please see Alaa et al. (2022).

C.4. Utility Metrics

Predictive score follows the ubiquitous "Train on synthetic, test on real" principle by using synthetic data to train downstream models, as we require that models trained on generated data may be readily applied to real circumstances without a substantial loss of efficacy (Jordon et al., 2022). The predictive score

metric creates a new post-hoc predictive recurrent network, $\mathcal{P}$, which learns to predict the next observation in a sequence using normalized synthetic training data, $\{\hat{\mathbf{x}}_{1:T}^{(i)}\}_{i=1}^{M}$. Once the model is trained, it may then be tested using real sequences from $\{\mathbf{x}_{1:T}^{(i)}\}_{i=1}^{N}$, and the predictive score is evaluated based on the mean absolute error (MAE) over component prediction

$$\text{Predictive Score} = \frac{1}{N}\sum_{i=1}^{N} \text{MAE}\left(\mathbf{x}_{1:T}^{(i)}, \mathcal{P}\right), \tag{7}$$

where,

$$\text{MAE}(\mathbf{x}_{1:T}, \mathcal{P}) = \frac{1}{nT}\sum_{t=1}^{T} \|\mathbf{x}_t - \mathcal{P}(\mathbf{x}_{1:t-1})\|_1 . \tag{8}$$

Here, $\|\cdot\|_1$ is the L1 norm, and n is the dimensionality of sequence observations. We may observe from (7) that lower is better for predictive scores. For additional details on this metric, please see Yoon et al. (2019).

The **downstream AUC-ROC** from the synthetic CKD data is calculated by considering the performance of a simple GRU classifier on predicting if a CKD patient has diabetes based on the features presented in Table 5.

After a train-test split, chosen to accommodate for the high variability in CKD expression, the synthetic data was passed through one GRU layer and one fully-connected layer, followed by a sigmoid activation. The hyperparameters for the downstream model are stated in Table 9.

Appendix D. Clinician Evaluation Details

Figure Appendix 5 shows a typical patient profile that was shared with the clinician along with the three questions that were used to evaluate the data.

We then provided the profiles to five CKD specialists, each of whom responded to the following three questions for each patient:

Q1 Does this patient follow a realistic CKD trajectory? (Y/N)

Q2 On a scale from 1 (least) to 5 (most), how likely do you think it is that this is a real patient?

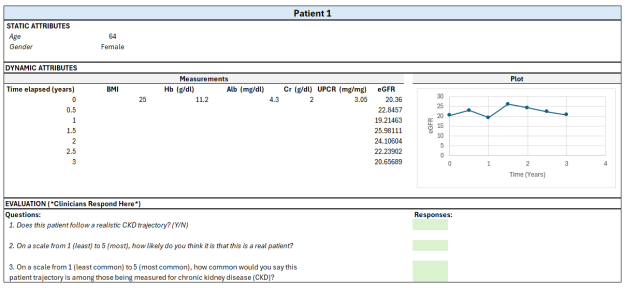


Figure 5: Sample test patient from blinded clinician evaluation.

Q3 On a scale from 1 (least common) to 5 (most common), how common would you say this patient trajectory is among those being measured for chronic kidney disease (CKD)?

This judged the realism and frequency of each synthetic sample in the clinic, providing a basis for how relevant the CKD sample would be. In addition to filling out the answers to the questions within the form, clinicians provided additional written feedback.

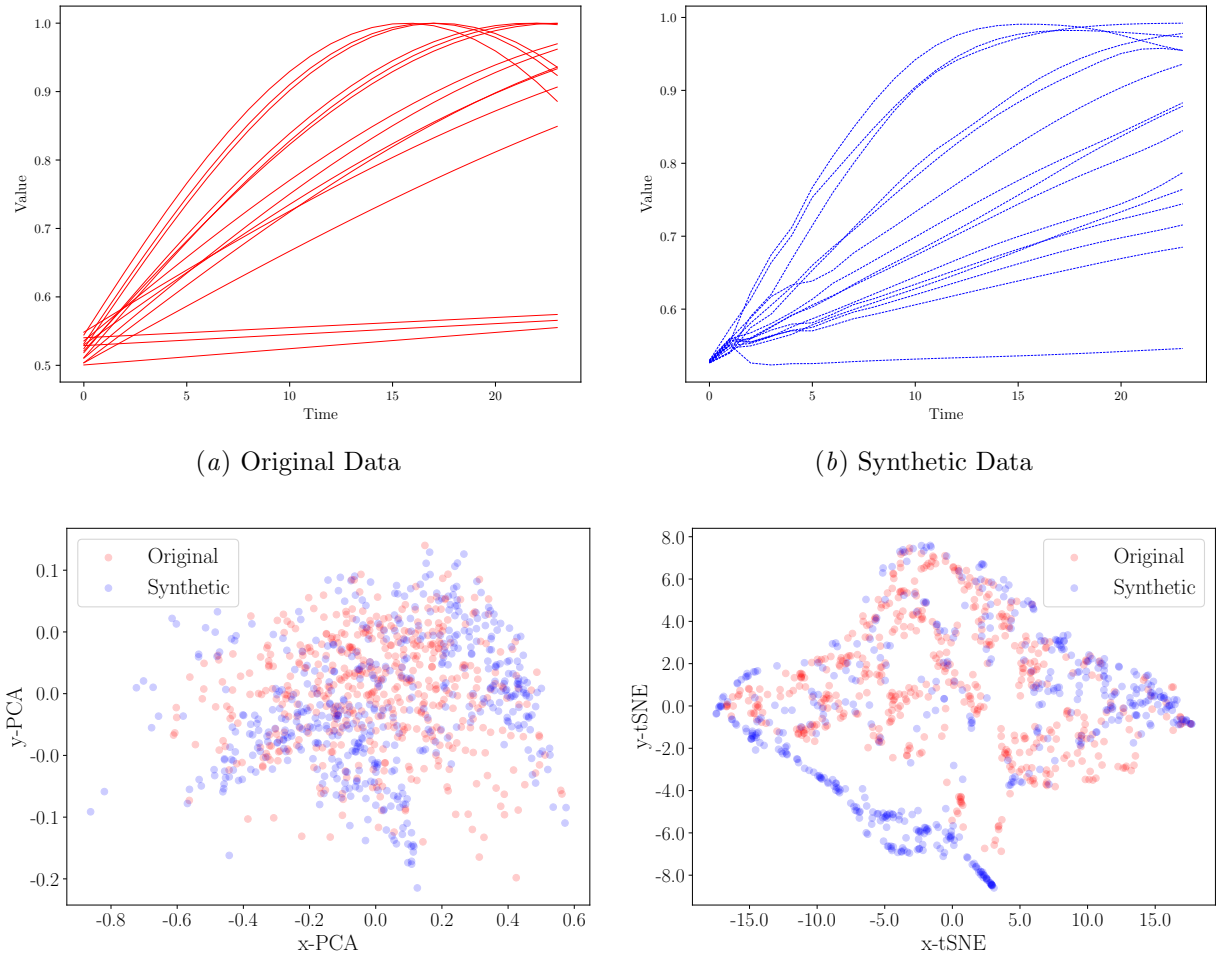
Appendix E. Generative Model Visualizations

In this appendix, we provide visualizations of data generated using DP-TimeGAN. To begin with, we exhibit a common test for generative time series models, which is to synthesize sinusoidal data, as it exhibits seasonality patterns that are often challenging to handle in traditional auto-regressive models. Here, we randomly sample phase and frequency values from a uniform distribution, which we use to construct a dataset of real sine waves, as described in Section 5. Figure 6 shows an example of the synthetic results from training DP-TimeGAN on the sinusoidal dataset.

As a second example, we utilize patient data from the eICU dataset. In Figures 7, 8, and 9, we exhibit the shape of synthetic data generated using our novel Augmented-TimeGAN and DP-TimeGAN as well as the baseline models. From these graphs it is clear that visually, our novel model achieves comparable performance to TransFusion and DP Normalizing Flows for the eICU dataset. Augmented TimeGAN also performs substantially better than other benchmarks, while the DP version is not far behind.

Appendix F. Privacy-Utility and Fidelity Tradeoff

To further describe the privacy-utility tradeoff seen with DP-TimeGAN, as mentioned in the Limitations, we measured utility and fidelity metrics with changing ϵ values. The results of this are shown in Table 10. As shown, with increasing ϵ values, the AUC-ROC increases, and fidelity measures also become closer to optimal. Therefore, if DP-TimeGAN is used in a clinical setting, the tuning of ϵ values is necessary in order to maximize privacy while still retaining high downstream potential and accurate chronic disease deterioration.



(a) Original Data

(b) Synthetic Data

(c) PCA visualization

(d) t-SNE visualization

Figure 6: Comparison of real and synthetic sinusoidal data from DP-TimeGAN. Real data uses (N, T, C) = (500, 24, 5), where each feature is a randomly generated sine wave; training parameters are: #epochs = 7000, #layers = 3, latent-dim = 24, $\gamma = 1$. Plots (a) and (b) isolate one feature of real and synthetic sine waves, respectively; plots (c) and (d) compare the PCA and t-SNE results, respectively, for the two datasets.

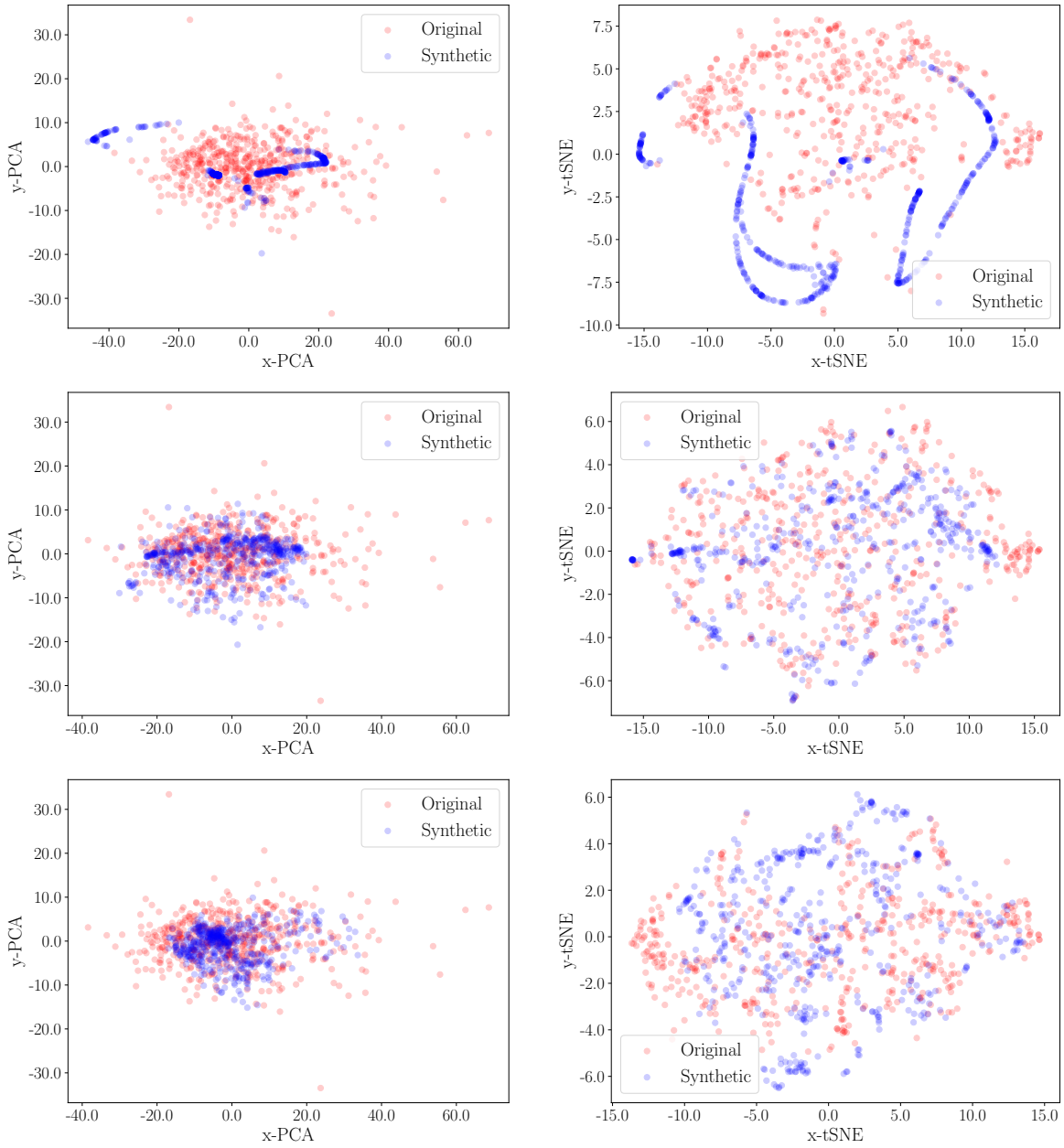


Figure 7: Comparison of real and synthetic eICU data from the Regular TimeGAN (Row 1), Augmented TimeGAN (Row 2), and DP-TimeGAN (Row 3) models. TimeGAN models utilize the same parameters as mentioned in Table 1. Plots in the first and second columns from the left compare the PCA and t-SNE results, respectively.

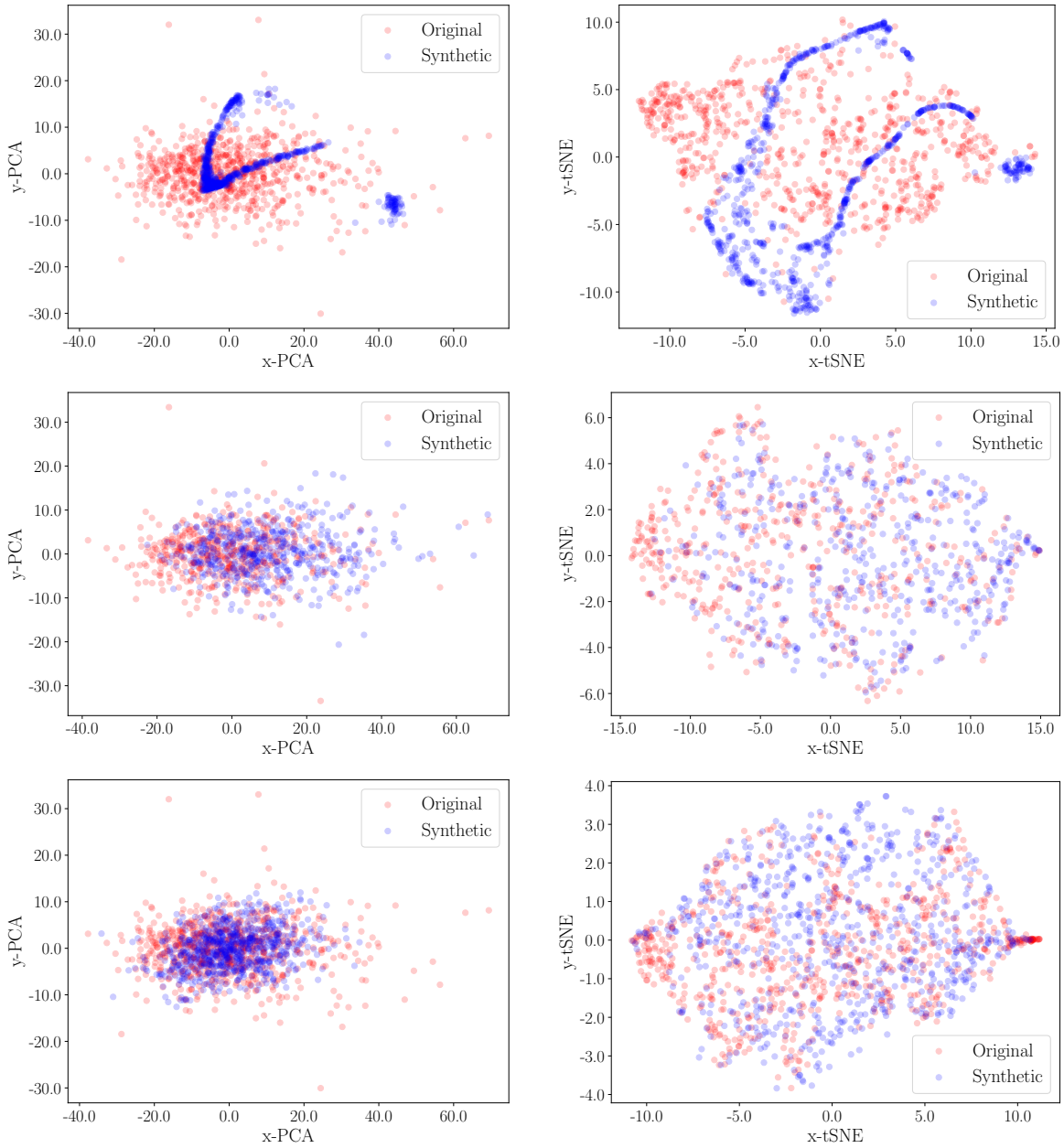


Figure 8: Continued comparison of real and synthetic eICU data from the SeriesGAN (Row 1), TransFusion (Row 2), and DP Normalizing Flows (Row 3) models. Benchmark models replicate the hyperparameters from their respective publications. Plots in the first and second columns from the left compare the PCA and t-SNE results, respectively.

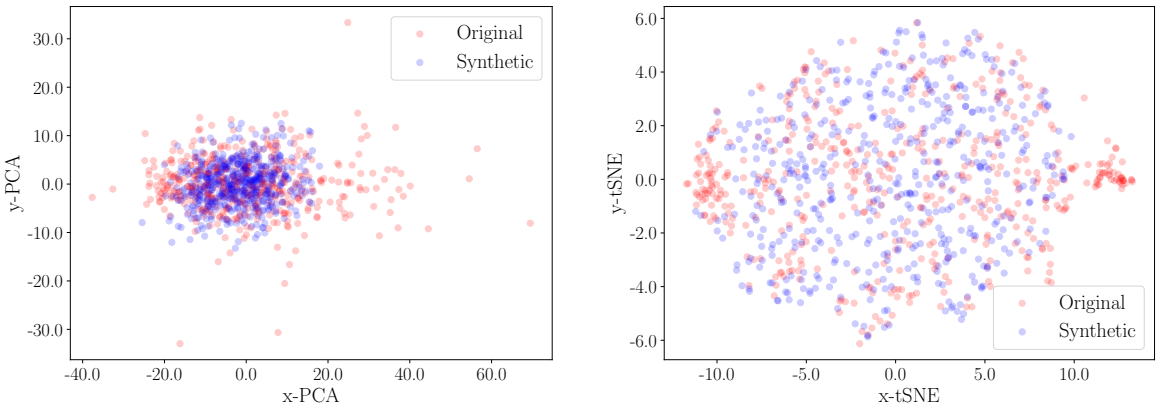


Figure 9: Continued comparison of real and synthetic eICU data from the TimeDiff model. Benchmark models replicate the hyperparameters from their respective publications. Plots in the first and second columns from the left compare the PCA and t-SNE results, respectively.

Table 10: Privacy-fidelity and -utility study of DP-TimeGAN on the CKD dataset. DP-TimeGAN utilizes the same parameters as mentioned in Table 1.

Epsilon	**MMD**	**DS**	**α-precision**	**Downstream AUC-ROC**
10	0.122 ± 0.058	0.380 ± 0.040	0.852 ± 0.051	0.501 ± 0.025
20	0.091 ± 0.046	0.312 ± 0.053	0.844 ± 0.035	0.549 ± 0.063
30	0.069 ± 0.002	0.320 ± 0.076	0.840 ± 0.079	0.553 ± 0.074
40	0.079 ± 0.050	0.321 ± 0.057	0.870 ± 0.032	0.553 ± 0.019
50	0.064 ± 0.002	0.297 ± 0.030	0.894 ± 0.066	0.577 ± 0.026