Computers in Biology and Medicine 163 (2023) 107115

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Diffusion-based conditional ECG generation with structured state space models

Juan Miguel Lopez Alcaraz, Nils Strodthoff *

The University of Oldenburg, 26129 Oldenburg, Germany

ARTICLE INFO

Keywords:
Cardiology
Electrocardiography
Signal processing
Synthetic data
Diffusion models
Time series

ABSTRACT

Generating synthetic data is a promising solution for addressing privacy concerns that arise when distributing sensitive health data. In recent years, diffusion models have become the new standard for generating various types of data, while structured state space models have emerged as a powerful approach for capturing long-term dependencies in time series. Our proposed solution, SSSD-ECG, combines these two technologies to generate synthetic 12-lead electrocardiograms (ECGs) based on over 70 ECG statements. As reliable baselines are lacking, we also propose conditional variants of two state-of-the-art unconditional generative models. We conducted a thorough evaluation of the quality of the generated samples by assessing pre-trained classifiers on the generated data and by measuring the performance of a classifier trained only on synthetic data. SSSD-ECG outperformed its GAN-based competitors. Our approach was further validated through experiments that included conditional class interpolation and a clinical Turing test, which demonstrated the high quality of SSSD-ECG samples across a wide range of conditions.

1. Introduction

Without a doubt, the vast amount of data generated worldwide has been a major catalyst for significant advancements in machine learning across a variety of data types and applications. However, acquiring and sharing data can present challenges, even among different departments within the same organization, due to privacy concerns. This is especially true in privacy-sensitive fields like healthcare. Considerable efforts have been made to enhance the security and privacy of healthcare data through the implementation of regulations such as GDPR or HIPAA [1,2], compliance measures [3–5], as well as the development of technical solutions such as blockchain technology or federated learning [6,7]. However, even technical solutions like federated learning alone do not guarantee complete privacy protection, as trained models can be reconstructed to reveal training samples through model inversion attacks [8]. Therefore, a combination of different privacy-enhancing techniques must be employed.

The ability to create digital replicas of raw data, known as digital twins, is a viable solution that preserves the statistical properties of the original data while removing personal patient information. Data augmentation techniques, such as statistical methods, provide limited privacy protection by disguising the original data. Therefore, the use of generative machine learning models has become increasingly popular. However, it is important to ensure that the recreated data is accurate, otherwise it may be biased [9] and negatively impact downstream tasks and decision-making, including interpretability [10]. This highlights the need for generative models that can produce high-quality samples based on different patient characteristics, as well as the development of objective benchmarking criteria to evaluate the quality of the generated samples.

Recently, diffusion models have exhibited remarkable outcomes for data synthesis, surpassing other models such as generative adversarial networks (GANs) or autoregressive models. These models have been shown to have biases towards high-density classes [11], as well as low fidelity dataset distributions [12,13], and are known to suffer from training instabilities [14]. On top of that, diffusion models have shown improvements on various downstream tasks [15], and have assisted in the identification of new pathologies [16].

In this study, our focus is on generating synthetic electrocardiogram data. The ECG is a commonly used medical procedure due to its non-invasive nature, simple and reliable technology, and high diagnostic value. It is an essential tool for the initial assessment of a patient's overall cardiac state. This importance is further highlighted by recent advances in AI-based ECG analysis, as discussed in [17]. However, many high-impact studies on ECG analysis have been conducted using private datasets, which poses a significant problem in terms of reproducibility and hinders progress in the research community as a whole. To enable the sharing of such datasets, high-quality digital twins of private datasets, which contain highly sensitive patient data and

* Corresponding author.
E-mail addresses: juan.lopez.alcaraz@uol.de (J.M.L. Alcaraz), nils.strodthoff@uol.de (N. Strodthoff).

https://doi.org/10.1016/j.compbiomed.2023.107115
Received 28 March 2023; Received in revised form 25 May 2023; Accepted 30 May 2023
Available online 7 June 2023

are subject to strict privacy requirements, need to be generated. We demonstrate this process by creating and evaluating synthetic copies of the publicly accessible PTB-XL dataset [18–20], which is a popular ECG dataset.

The main contributions of this work are the following: (1) We propose a diffusion model for generating short (10 s) 12-lead ECGs. This model uses a structured state space model as its internal component. (2) We introduce conditional variants of two state-of-the-art unconditional generative models for ECG data. (3) We generate synthetic versions of PTB-XL, a large publicly available ECG dataset, and evaluate the quality of the generated samples by training and testing classifiers on these datasets. (4) We demonstrate that our model has internalized domain knowledge in several ways: (a) by comparing generated samples using a beat-level aggregation across subgroups of samples with common pathologies, (b) by meaningfully interpolating between different sets of conditions, and (c) by conducting an expert assessment in the form of a clinical Turing test that confirms the high quality of the generated samples.

2. Materials & methods

2.1. Dataset and downstream task

The focus of this study is on short 12-lead ECGs that last 10 s. These ECGs are obtained from six limb leads and six precordial leads, which is the most commonly used method in clinical practice. The ECGs were sampled at a rate of 100 Hz, as previous research has shown that increasing the sampling rate to 500 Hz does not significantly improve the ability to classify ECGs [21].

PTB-XL dataset

Our experiments are based on the PTB-XL dataset [18–20], which is a publicly available collection of clinical 12-lead ECG data comprising 21,837 records from 18,885 patients. In order to train a high-quality generative model, it is important to have a dataset of sufficient size. For class-conditional generative models, which require sample-wise annotations for all samples in the dataset similar to supervised discriminative training, PTB-XL is a good choice as it provides annotations for each sample in terms of 71 ECG statements in a multi-label setting. These cover 44 diagnostic (organized into 24 sub-classes and 5 superclasses comprising normal, conduction disturbance, myocardial infarction, hypertrophy, ST/T changes), 19 form-related (5 of which also counted as diagnostic statements), such as abnormal QRS-complex, and 12 rhythm-related statements, such as atrial fibrillation. It is worth noting that this dataset represents a significant advancement in terms of complexity, as it includes a broad set of 71 ECG statements, which can be used in a multi-label setting. Most literature approaches have typically focused on a single condition, such as healthy samples, or on generating samples based on very limited sets of class labels. For further information about the dataset, please refer to Appendix A and the dataset descriptor [18].

Downstream task

As downstream task, we are examining the problem of predicting ECG statements at the most granular level, which involves classifying them into multiple labels. This task is a well-researched benchmark on the PTB-XL dataset, and we follow the established methodology outlined in previous work [22]. In particular, we make use of the proposed stratified folds for model training, where the first 8 folds are used for training, the 9th fold serves as validation set and the 10th fold as test set. Our evaluation metric for this task is the macro-averaged area under the receiver operating curve (macro AUROC) across all 71 labels on the PTB-XL test set. As model architecture, we use a XResNet1d50 model, which is a one-dimensional adaptation of a modern ResNet model [23], as proposed in [22]. We closely follow the training and evaluation methodology outlined in [22], with a binary crossentropy loss as our training objective, which is appropriate for a multi-label classification problem. We train the model on random crops of 2.5 s in length and during test time, we average seven overlapping predictions from the same sample to obtain the final prediction for the sample. To prevent overfitting, irrespective of whether we train on real or synthetic data, we always perform model selection based on a corresponding real/synthetic validation set score (macro AUC). For further information on the downstream classifier and training, please refer to Appendix B.

2.2. Background

Diffusion models

Diffusion models [24], which are a type of generative model, have shown state-of-the-art performance on a variety of data modalities, including audio data [25,26], image data [12,27–29], and video data [30]. Diffusion models consist of two processes: the forward process and the backward process. During the forward process, noise is incrementally introduced in a Markovian manner. In contrast, during the backward process, the model gradually removes the noise. The forward process is parameterized as

$$q(x_1, \dots, x_T|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{1}$$

where $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbb{1})$, β_t are (fixed or learnable) forward-process variances, which adjust the noise level and T is the number of diffusion steps. Equivalently, x_t can be expressed in closed form as $x_t = \sqrt{\alpha_t}x_0 + (1-\alpha_t)\epsilon$ for $\epsilon \sim \mathcal{N}(0, \mathbb{1})$, where $\alpha_t = \sum_{i=1}^{t}(1-\beta_t)$. The backward process is parameterized as in Eq. (2), where $x_T \sim \mathcal{N}0, \mathbb{1})$.

$$p_\theta(x_0, \dots, x_{t-1}|x_T) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{2}$$

Using a particular parameterization of $p_\theta(x_{t-1}|x_t)$, it was shown in [27] that the reverse process can be trained using

$$L = \min_\theta \mathbb{E}_{x_0 \sim D, \epsilon \sim \mathcal{N}(0,\mathbb{1}), t \sim \mathcal{U}(1,T)} \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + (1-\alpha_t)\epsilon, t)\|_2^2, \tag{3}$$

where $\epsilon_\theta(x_t, t)$ is parameterized using a neural network and D denotes the data distribution. This objective can be understood as a weighted variational bound on the negative log-likelihood that reduces the significance of terms at low t, i.e., at low noise levels. Class-conditional diffusion models can be realized by conditioning the backward process on desired set of labels c, i.e., using $\epsilon_\theta = \epsilon_\theta(x_t, t, c)$.

Structured state space models

In essence, structured state space models (SSSMs) rely on a linear state space transition equation that links a one-dimensional input sequence, denoted as $u(t)$, with a one-dimensional output sequence, denoted as $y(t)$, by way of a hidden state $x(t)$ that is of N dimensions,

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{4}$$

where A, B, C, D are transition matrices. In [31], it is discussed that the SSSM (Structured State Space Model) is a promising model for capturing long-term dependencies in time series. This is achieved by discretizing the input and output relations and representing them as a convolution operation. This operation can be effectively computed on modern GPUs using a custom kernel. The ability to capture long-term dependencies is closely linked to the initialization of the hidden-to-hidden transition matrix A, as discussed in [32]. By stacking multiple SSSM blocks, each with appropriate normalization and point-wise fully-connected layers, in a manner similar to a transformer layer, one can create a Structured State Space Sequence Model (S4). The S4 model has demonstrated outstanding performance on a variety of long-range-interaction benchmarks and sequence classification tasks, including 12-lead ECG classification, as shown in the most recent work [21].

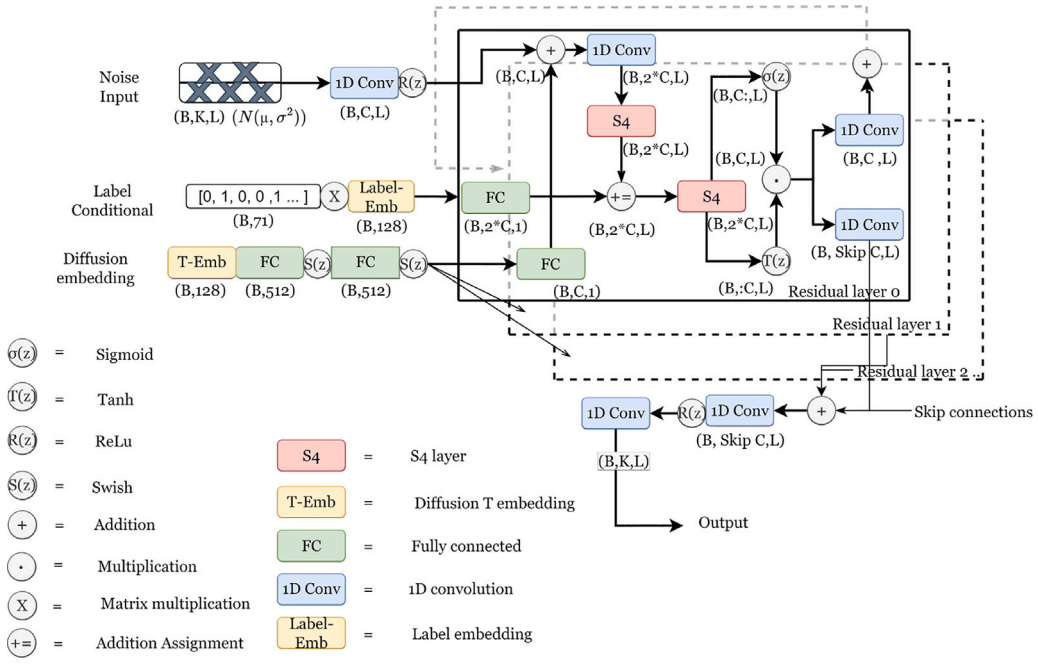

Fig. 1. Schematic representation of the *SSSD-ECG* model architecture.

Related work

Deep generative modeling for time series data is an emerging subfield in machine learning, driven mainly by the constant progress in the development of generative models, particularly in the field of imaging. While various algorithm architectures have been developed, such as variational autoencoders [33] and stacked denoising autoencoders [34], generative adversarial networks (GANs) remain the dominant technology in the field, especially with recurrent neural networks (RNNs) [35–38], transformers [39], or differential equations [40] as building blocks. Although some prior research has addressed conditional time-series generation [33,36,38,40,41], most of it is still limited to the unconditional setting [35,37,39,42,43].

During the final stage of the manuscript preparation, we became aware of [43], whose authors also applied a diffusion model for synthetic ECG generation, albeit using image representations rather than the time series directly and for the restricted case of the unconditional generation of single-channel ECGs. Interestingly, this approach was unable to outperform a GAN-based baseline according to several quality metrics. Also very recently, Chung et al. proposed a conditional generative model to generate synthetic ECGs from text reports [44]. Due to the different task setup, their results are not directly comparable to ours but represent an interesting direction for future research. Nevertheless, we see it as an import next step to establish proper baselines for the more controlled case of ECG generation based on structured labels first.

Apart from these very recent works, numerous literature approaches have attempted to address the issue of creating synthetic ECG signals, but these approaches are often subject to significant limitations. Firstly, many of these models are only able to generate short time series [37–39,42], which is why many of them restrict to the generation of single beats. Secondly, they are frequently trained on small sets of data from few patients [36] or with limited conditional labels [33,40,41]. Thirdly, many of these approaches require ECG beat segmentation as a preprocessing step, rather than working directly on continuous signals [33,38–40]. Moreover, many of these approaches suffer from different evaluation issues, such as being limited to patient-specific generation and classification [41] or lacking training data and software for filtering/feature extraction for the general public [35] (see Fig. 1).

2.3. Conditional generative models for ECG data

SSSD-ECG

Returning to the discussion of probabilistic diffusion in Section 2.2, the only remaining component that needs to be specified is the explicit parameterization of the backward process, denoted by $\epsilon_\theta = \epsilon_\theta(x_t, t, c)$. In this regard, we have implemented an adaptation of the recently proposed $SSSD^{S4}$ model [45],where SSSD stands for structured state space diffusion. The model is built on the DiffWave architecture [26], which was originally proposed for audio synthesis using dilated convolutions. However, in $SSSD^{S4}$, dilated convolutions are replaced by two S4 layers (as described in Section 2.2) to handle long-term dependencies better in time series. This has been shown to be effective for various time series imputation across different scenarios and forecasting tasks in [45]. *SSSD-ECG*, which is proposed in this work, builds on the $SSSD^{S4}$ model architecture but also deviates from it in several important aspects. The most important difference is due to the different conditional information that is provided for respective applications and the omission of task specific masking strategies. While $SSSD^{S4}$ receives an imputation mask and the remaining input signal as conditional information, *SSSD-ECG* is conditioned on the set of annotated ECG statements, which is in our case encoded as a binary vector of length 71. This vector is transformed into a continuous representation by multiplying it with a learnable weight matrix, transformed through a fully connected layer and is subsequently passed as conditional information to different $SSSD^{S4}$ layers. Figure Fig. 1 shows a schematic representation of the *SSSD-ECG* model architecture. For more details about the model's internals and hyperparameters, please refer to Appendix C.

As a further task-specific modification, it is worth mentioning that in a standard 12-lead ECG, only two out of the six limb leads are independent. This means that any set of limb leads can be reconstructed using any two given limb leads, based on the defining relationships III = II-I, aVL = (I-III)/2, aVF = (II+III)/2, and -aVR = (I+II)/2. To ensure that the ECGs we generate satisfy these relationships, we use generative models to synthesize only 8 leads — the 6 precordial leads and 2 limb leads (I and aVF in our case). We then reconstruct the remaining 4 leads by sampling from the two limb leads (I and aVF). This approach

is similar to the one used by [35] and is also applied to all baseline models described below.

We believe that using a publicly available ECG dataset for training is a critical step towards measurable progress. This allows for improvements in model architecture and training schedules to be disentangled from improvements resulting solely from larger or more comprehensive training datasets. In contrast to previous limitations, the *SSSD-ECG* model generates long sequences of 1000 time steps (for a 10-second ECG at 100 Hz), is trained on a large dataset of over 18,000 patients, and is conditioned on a rich set of 71 ECG statements. Additionally, it generates full samples without the need for prior segmentation or any other preprocessing and builds on publicly available code [46].

Baselines: WaveGAN and Pulse2Pulse*

Our primary contribution is the *SSSD-ECG* model, but we also present conditional versions of two existing generative models for ECG data, namely WaveGAN and Pulse2Pulse [35]. These models use Generative Adversarial Networks (GANs) [47]. We make these models class-conditional by incorporating batch normalization layers into their architecture and converting them into conditional batch normalization layers [48]. This involves making the layer's internal shift and scaling parameters dependent on the class. We follow a similar approach to map the binary label vector into a continuous representation using a learned weight matrix. Further details about the model configurations and training hyperparameters are provided in Appendix D.

Regrettably, we were unsuccessful in training effective generative models for our particular case using publicly available implementations, except for WaveGAN and Pulse2Pulse as mentioned earlier. Our attempts included creating generative models for time series generation, such as TTS-GAN [39] and well-established methods like TimeGAN [42], but these models were likely challenged by input lengths of 1000 time steps. We were also unable to generate samples at 250 time steps. Additionally, we were not successful in training a class-conditional model using the cVAE_ECG [33] approach, as described in Appendix D.3.

2.4. Performance measures for generative models

In this section, our objective is to propose measures to quantify the quality of the generated samples. We can achieve this by training classifiers on either real or synthetic training sets and testing them on either real or synthetic test sets. It is worth noting that we only need to train a single classifier on real data, which we will refer to as the reference classifier, which is kept fixed for all the remaining experiments. From it, one can infer three different performance measures:

(1) The primary criterion for evaluating the quality of synthetic data is its *capacity to replace real data*. This evaluation involves testing a classifier that is trained on a synthetic training set with a real test set. The ranking of various algorithms based on this measure is widely regarded as the gold standard for assessing the quality of synthetic data models, as mentioned in [49]. However, it is also valuable to compare the absolute performance of the synthetic classifier with that of the reference classifier trained on real data.

(2) The second metric, which complements the first, aims to determine *the realism of the synthetic data by evaluating it using a reference classifier*. This involves using the reference model, which was trained on real data, to assess the performance of the synthetic test set. The expected decrease in predictive performance, as compared to the evaluation using the real test set, is due to the inherent difference between the distribution of the real training data and that of the synthetic test data. This decrease can serve as a second measure of the performance of generative models.

(3) The third performance indicator takes a distinct approach by evaluating the *internal consistency*. It assesses how well a classifier trained on synthetic training data generalizes to unseen synthetic test data from the same distribution. To evaluate this, a classifier is trained on a synthetic training set and tested on its corresponding synthetic test set. However, this criterion does not reference the real data directly or indirectly through a reference classifier trained on real data, which is why it is considered less informative than the first two criteria.

We would like to draw attention to recent research such as [49] that focuses on evaluating the performance of generative models, specifically in producing output that matches a predetermined standard. However, during our attempts to replicate the findings, we encountered problems with instability when training one-class embedding as a necessary step in [49], which proved to significantly affect the results. Therefore, we have chosen not to include those corresponding results in our paper.

3. Experiments and results

We trained three generative models – *SSSD-ECG*, WaveGAN*, and Pulse2Pulse – based on the eight PTB-XL training folds. Each model was conditioned on the respective sample annotations in the dataset. After training each model, we generated a synthetic copy of the PTB-XL dataset as follows: For each sample in PTB-XL, we generated a new synthetic sample by conditioning on the ECG annotation of that sample. For each of the three generative models, this resulted in a synthetic ECG dataset that matches PTB-XL in terms of size and label distribution. In this way, we obtain synthetic training, validation and test sets, whose label distributions match exactly the corresponding PTB-XL sets. We used these synthetic datasets for both qualitative and quantitative assessments of the quality of the generated samples.

3.1. Qualitative assessment

To conduct a qualitative assessment, we chose two commonly occurring conditions, namely normal ECGs (NORM) and ECGs with left-ventricular hypertrophy (LVH) In order to be able to compare across samples, we perform beat-level segmentation across 250 generated samples for each condition by identifying R-peaks and cropping from 300 ms before the R-peak until 500 ms after the R-peak. Now, we plot the median across all beats along with the corresponding 0.25 and 0.75 quantiles visualized through a shaded band. This allows for a direct visual comparison of the main characteristics of the generated samples, both across different models and in comparison to the real samples from PTB-XL.

Fig. 2 presents a qualitative evaluation of healthy (NORM) generated samples compared to real samples. The WaveGAN* model produces signals with a significant amount of interference and numerous ECG features that differ from those found in real signals, such as larger R-peaks and absent P- and T-waves in most leads due to signal inference. The Pulse2Pulse model produces more consistent results, with less variability across quantiles. However, there are some mismatched features compared to real samples, including the absence of an S-peak in lead I, a larger S-peak in lead II, and an opposite (upward) T-wave in lead V1. The *SSSD-ECG* model is the one that closely resembles real samples, with a higher level of confidence as the quantile bands closely approximate the median in all features. This model correctly replicates the P- and T-waves in leads I and II, as well as the downward T-wave in lead V1. In addition, the features in lead V2, such as the R- and S-peaks and P- and T-waves, are balanced.

Fig. 3 shows a comparison of synthetic samples with left-ventricular hypertrophy (LVH) generated by different proposed models to real samples, based on a qualitative assessment. The WaveGAN* model has a lot of signal interference, which makes it difficult to observe certain

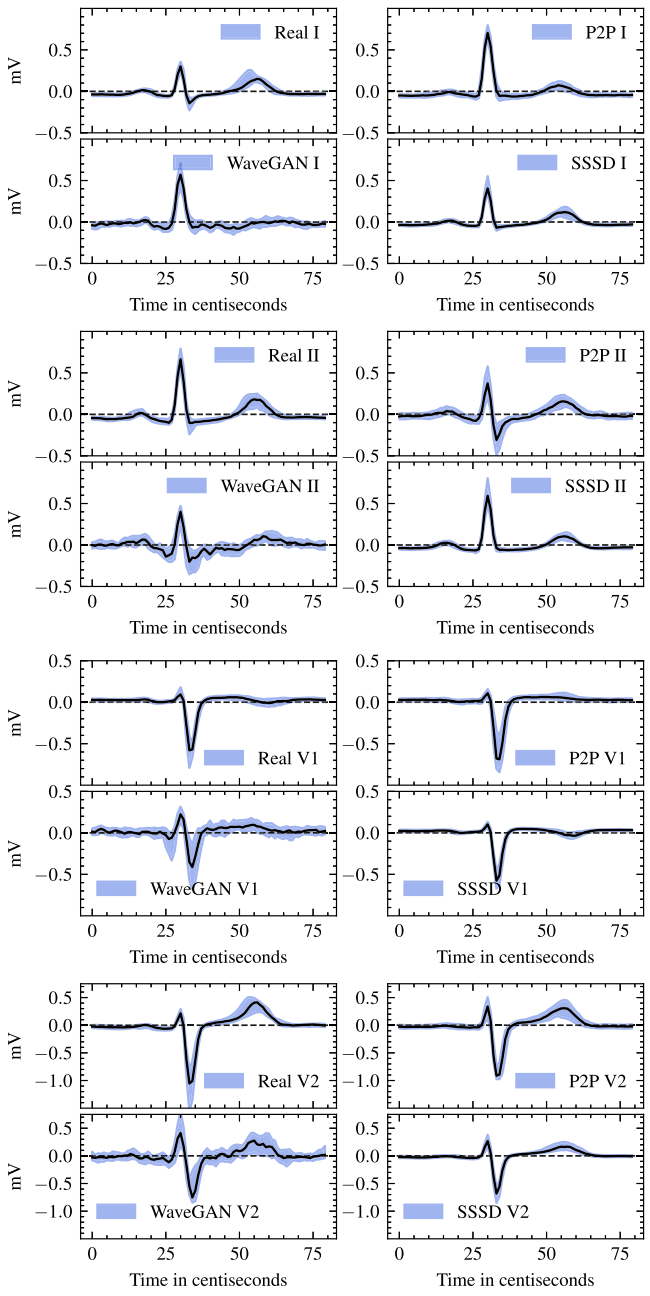

Fig. 2. From top to bottom four grids of leads I, II, V1, and V2 respectively, where for each grid from top to bottom and left to right real, WaveGAN*, Pulse2Pulse, and *SSSD-ECG* Median (black line) and 0.25–0.75 quantiles (blue shaded area) segmented beats for healthy (NORM) samples.

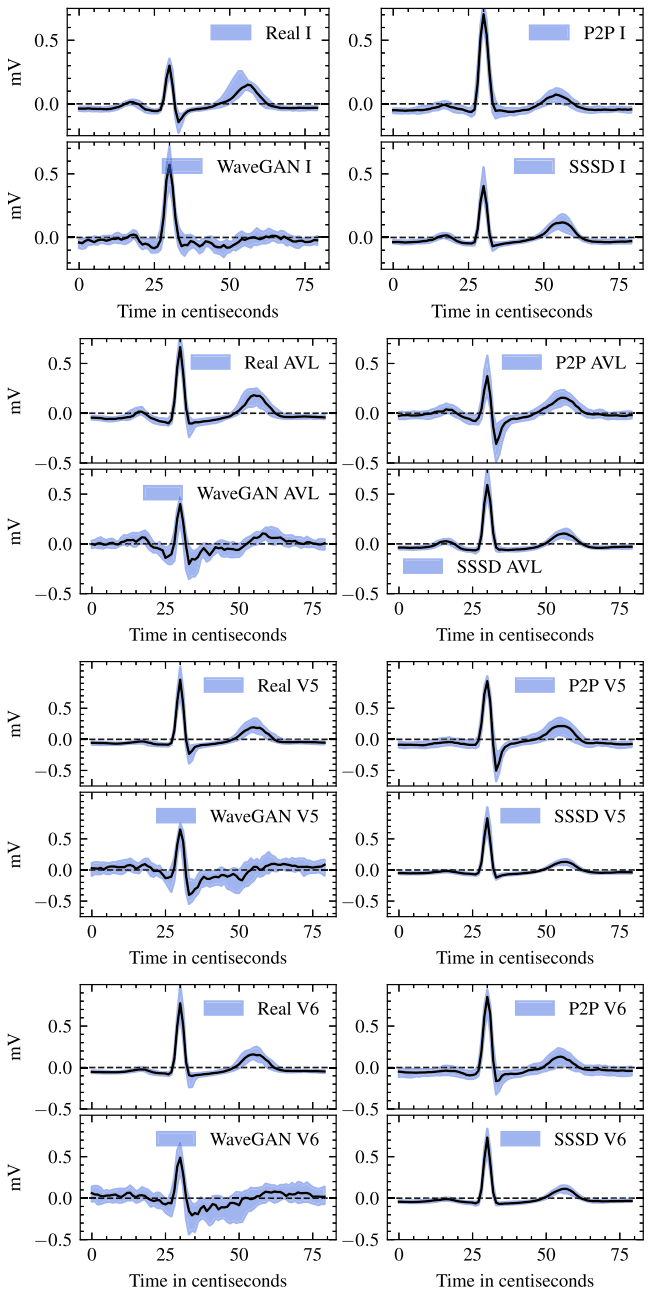

Fig. 3. From top to bottom four grids of leads I, AVL, V5, and V6 respectively, where for each grid from top to bottom and left to right real, WaveGAN*, Pulse2Pulse, and *SSSD-ECG* Median (black line) and 0.25–0.75 quantiles (blue shaded area) segmented beats for left-ventricular hypertrophy (LVH) samples.

features such as P- and T-waves, although the R-peak seems to be correctly shaped. In contrast, the Pulse2Pulse model fails to accurately generate the R-peak, which appears curved towards the top in all leads, and also fails to generate many other features, such as a straight T-wave in V5 and a downward T-wave in V6, with a high degree of uncertainty. Finally, the *SSSD-ECG* plots are the most closely aligned with the real samples, generating a correct R-peak with relatively short P-waves and balanced T-waves for all beats, with consistent confidence intervals across all leads.

In summary, this initial qualitative evaluation provides indications of the superiority of the *SSSD-ECG* samples over its competitors and is consistent with the characteristics of the corresponding real samples from PTB-XL.

3.2. Quantitative assessment

In Table 1, we present a quantitative comparison of the three proposed generative models based on the performance metrics introduced in Section 2.4. The first metric shows that *SSSD-ECG* outperforms its competitors with a score of 0.84, compared to only 0.60 for Pulse2Pulse and 0.58 for WaveGAN*. One possible explanation for this result is that Pulse2Pulse and WaveGAN* are both GAN-based approaches, which tend to focus on selected nodes rather than covering the full distribution. The second metric also shows a similar pattern, with *SSSD-ECG* clearly outperforming Pulse2Pulse (0.71) and WaveGAN* (0.64) with a score of 0.94. According to the third metric, all three approaches achieve scores of 0.98 or higher, indicating a high degree of internal

Table 1
Classification performance of XResNet50 models trained/evaluated on different combinations of real/synthetic data.

Model	AUROC	
WaveGAN*	Test real	Test synth.
Train real	0.9317	0.6489
Train synth.	0.5816	0.9793
Pulse2Pulse	Test real	Test synth.
Train real	0.9317	0.7082
Train synth.	0.5968	**0.9950**
SSSD-ECG	Test real	Test synth.
Train real	0.9317	**0.9434**
Train synth.	**0.8402**	0.9822

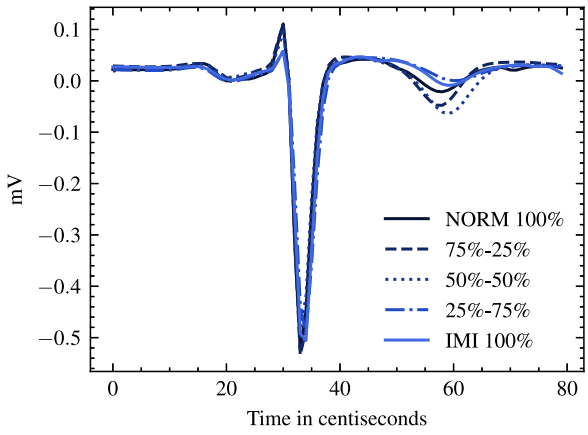

Fig. 4. *SSSD-ECG* interpolation between a healthy (NORM+SR) and an inferior myocardial infarction (IMI+SR+ABQRS) signal in lead V1, where 5 signals of 100% and 0%, 75%, and 25%, 50% and 50%, 25% and 75%, and 0% and 100% of healthy and inferior myocardial infarction, respectively.

consistency among the models. In summary, the results of our experiment demonstrate a clear quantitative advantage of *SSSD-ECG* over its GAN-based competitors. We will now reflect in more detail on the quantitative performance of *SSSD-ECG* and its implications:

Remarkably, on one hand, *SSSD-ECG* even achieves a slighter better score (0.94) assessed through the reference classifier compared to evaluating the reference classifier on real samples. This is an encouraging prospect for auditing ECG analysis algorithms, as pre-screening with synthetic data may be conducted before evaluating them on high-quality private test data.

On the flip side, it is important to acknowledge that the *SSSD-ECG* performance during training with synthetic data and testing with real data is notably lower compared to the reference classifier trained on actual data (0.84 vs. 0.93). It is important to reiterate the difficulty of the generation task at hand. This work is the first attempt at constructing a generative model that is dependent on a diverse range of 71 ECG statements. Additionally, one must not underestimate the fact that these 71 statements are employed in a multi-label environment, where the number of unique label combinations (including co-occurring conditions) is significantly higher, which is a reflection of the complex reality of co-existing diseases and disease states. Some of the ECG statements are already sparsely populated with less than 100 occurrences throughout the entire dataset, and this is even more pronounced in the case of co-occurring label combinations. We consider it to be a challenging but worthwhile objective for the research community to devise methods that will bridge the gap in the "train on synthetic, test on real" situation, ultimately allowing synthetic data to be used essentially interchangeably with real data.

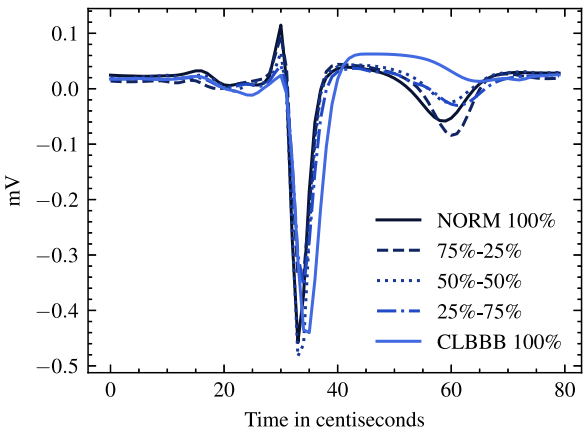

Fig. 5. *SSSD-ECG* interpolation between a healthy (NORM+SR) and a complete left bundle branch block signal (CLBBB+SR) in lead V1, where 5 signals of 100% and 0%, 75%, and 25%, 50% and 50%, 25% and 75%, and 0% and 100% of healthy and complete left bundle branch block, respectively.

3.3. Conditional class interpolation

In order to show that the *SSSD-ECG* model has gained valuable knowledge in the specific field of ECG analysis, we conducted various class interpolation experiments. Unlike other models, our model is not limited to using only binary vectors as conditional information. Instead, we can use any real-valued vectors with values ranging from 0 to 1. By using a convex combination of two binary annotation vectors A and B, specified by $\alpha a + (1 - \alpha)b$ for $\alpha \in [0, 1]$, we can interpolate between the two conditions. The parameter α determines the weight given to condition A. The sample's initialization is kept constant throughout the process. By varying α, we can smoothly transition between the two conditions. To better illustrate this, we divided the generated samples based on R-peaks and only presented median beats extracted from the signal. This makes it easier to observe signal changes as we move from one condition to another. This not only serves as an interesting consistency check, but it also opens up possibilities for more complex generative models that can incorporate non-binary disease states.

Inferior myocardial infarction

Fig. 4 shows the interpolation between a healthy normal sample (NORM) and a signal from an inferior myocardial infarction (IMI) for lead V1. The labels for the healthy samples are based on their occurrence in the training sample and represent NORM and sinus rhythm (SR). The labels for the IMI disease include SR, IMI, and abnormal QRS complex (ABQRS). It can be observed that the generated IMI signals and their interpolations exhibit an early Q-wave formation, whereas the normal signal has a shorter and downward shape.

Complete left bundle branch block

Fig. 5 displays an interpolation between a healthy signal and a complete left bundle branch block (CLBBB) signal as observed in lead V1. The labels assigned to the signals are based on the occurrence of training samples for healthy signals, which are labeled as NORM and SR, and for CLBBB signals, which are labeled as SR and CLBBB. In an electrocardiogram (ECG), various main features are observed that are more representative of the CLBBB disease. Firstly, as the disease increases, the QRS complex widens, which is clearly visible in the 75% and 100% CLBBB signals. Secondly, on all signals, the RSR feature that is characteristic of CLBBB becomes larger as the disease progresses. Lastly, an upward trend on the T-wave can also be observed as the disease becomes more severe.

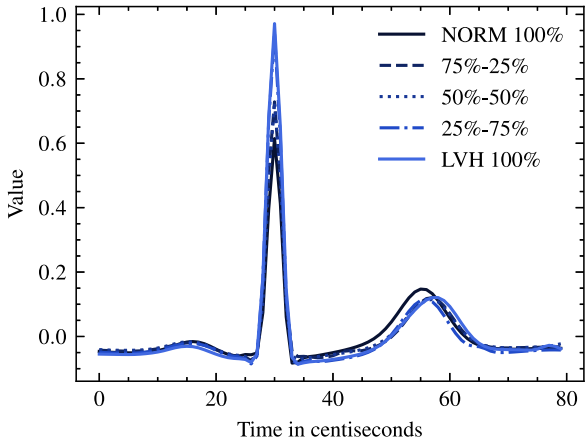

Fig. 6. *SSSD-ECG* interpolation between a healthy (NORM+SR) and a left ventricular hypertrophy signal (LVH+SR+VCLVH) in lead V5, where 5 signals of 100% and 0%, 75%, and 25%, 50% and 50%, 25% and 75%, and 0% and 100% of healthy and left ventricular hypertrophy, respectively.

Left ventricular hypertrophy

Fig. 6 displays an interpolation between a healthy signal and a signal with left ventricular hypertrophy (LVH), as observed from lead V5. The labels used for healthy samples are NORM, SR, and for LVH samples, the labels used are SR, LVH and voltage criteria (QRS) for left ventricular hypertrophy (VCLVH), based on training sample occurrence. An important ECG feature of LVH can be observed, which is the enlargement of the R-peak in one of the left-side leads (V5). As the disease progresses, the R-peak also increases in size, and there is also a downward trend in the T-wave as the disease worsens.

In summary, these case interpolation studies, conducted across three different domains and aligned with domain knowledge on the considered conditions, suggest that *SSSD-ECG* has gained a significant understanding of how different label combinations relate to specific signal features.

3.4. Expert evaluation

As final component of our evaluation of the sample quality, we presented the generated samples to an expert clinical and interventional cardiologist for qualitative assessment.

Generative diagnosis on normal samples

To perform this task, we provided the expert with four 10-second 12-lead ECGs, including one real ECG and one sample from each of the generative models. These four complete ECG recordings offer a detailed visual representation of the generated samples. The expert was then asked to distinguish between real and synthetic signals.

Fig. 7 depicts a 10-second real NORM signal that was evaluated by the medical expert. The diagnosis was a synthetic sample because of changes in voltage in leads I, III, and AVF, and a slow first vector (R-wave) progression in the precordial leads V1-V6. Fig. 8 shows a 10-second WaveGAN* NORM signal that was also evaluated by a medical expert, and the diagnosis was a synthetic sample due to the absence of a sinus signal, unclear P-waves, high interference, no symmetry on RR intervals, and changes in voltage on the same trace. Fig. 9 depicts a 10-second Pulse2Pulse NORM signal that was also evaluated by a medical expert, and the diagnosis was a synthetic sample because of the variability in P-waves, no symmetry on RR intervals, high interference, and morphological changes in the same trace within the same lead. Lastly, Fig. 10 depicts a 10-second *SSSD-ECG* NORM signal that was diagnosed as a real sample by the medical expert. The diagnosis was due to the well-defined P-wave and RR interval, and although there was a bit of tachycardia, the signal contained clear isodiphasic patterns, particularly in the I and AVF leads.

Fig. 7. Real (PTB-XL) NORM sample.

Fig. 8. Synthetic WaveGAN* NORM sample.

Clinical turing test

In this section, we present the results of a Turing test conducted with a medical professional previously mentioned. The task consisted of presenting 22 pairs of 12-leads ECGs (a total of 44), where one is synthetic (a total of 22) and the other is real (also a total of 22). The cardiologist was asked to perform two tasks. The first task was to decide if the sample was consistent with the provided set of ECG statements, which was either used to select a corresponding normal sample from PTB-XL or used to generate the synthetic sample. The second task was to determine if any of the ECGs in the given pair appeared synthetic. In Appendix E, we provide a detailed breakdown of the various combinations involved, which correspond to the most frequent label combinations in PTB-XL, along with specific results for each pair. However, in this section, we will only report simple descriptive statistics. The original samples will be made available as part of the code repository [46].

In the first part of the assignment, which involved evaluating diagnosis samples, a total of 44 samples were assessed. Among these, the cardiologist incorrectly diagnosed 19 (43.18%) and correctly diagnosed 25 (56.81%). Specifically, out of the 22 synthetic samples, 7 (31.81%)

Fig. 9. Synthetic Pulse2Pulse NORM sample.

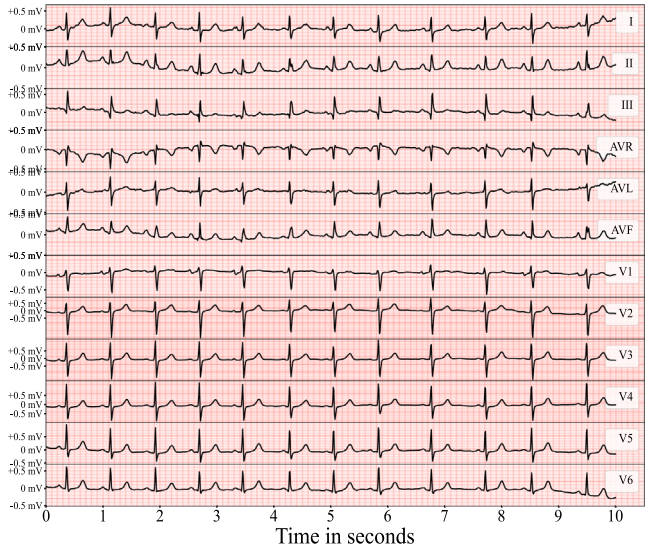

Fig. 10. Synthetic *SSSD-ECG* NORM sample.

were diagnosed incorrectly and 15 (68.18%) were diagnosed correctly. Similarly, out of the 22 real samples, the cardiologist diagnosed 12 (54.54%) incorrectly and 10 (45.45%) correctly. The higher confirmation rate of 68% for the synthetic samples, compared to 45% for the real samples, indicates that the synthetic ECGs are very accurate and match their designated labels very well.

In the second part of the assignment, which involved identifying synthetic samples, a total of 44 samples were considered. Among these, the cardiologist identified 10 as synthetic, which represents 22.72%, and 34 as real, which represents 77.27%. Out of the 22 synthetic samples, the cardiologist identified 5 as synthetic (22.72%) and 17 as real (77.27%). Similarly, out of the 22 real samples, the cardiologist identified 5 as synthetic (22.72%) and 17 as real (77.27%). The fact that 5 samples were identified as synthetic in both cases confirms that it is difficult for a medical expert to distinguish between real and synthetic samples. It is worth emphasizing that this conclusion applies to a diverse set of 22 different medical conditions, which underscores the high quality of the synthetic samples generated.

Table A.2
PTB-XL dataset details.

Description	Value
Train set size	17,441
Validation set size	2193
Test set size	2203
Sample length	1000
Sample feature	12
Labels	71

4. Conclusion

In our study, we proposed a novel approach called *SSSD-ECG* for generating electrocardiogram (ECG) data using diffusion-based techniques. We conditioned our model on a diverse set of 71 ECG statements in a multi-label setting, which is a highly complex task. The generated samples excelled in different context from qualitative over quantitative evaluation to conditional label interpolation and a human expert evaluation, clearly outperforming the two GAN-based competitors also proposed in this work. However, we observed that the generated *SSSD-ECG* samples were not entirely capable of compensating for real samples when training a classifier on them. We believe that bridging this gap would be a significant achievement and a measurable sign of progress in the field in the near future. To encourage further research in this field, we are releasing the source code used in our investigations [46], as well as the trained models and a synthetic copy of the PTB-XL dataset generated by *SSSD-ECG* [50].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to extend their gratitude to Erick Davila Zaragoza for conducting a meticulous evaluation of the samples that were generated. Mr. Zaragoza is a certified clinical and interventional cardiologist accredited by the Mexican National Council of Cardiology as a general practitioner from the University of Guadalajara (UDG), and as a cardiologist from the National Autonomous University of Mexico (UNAM).

Appendix A. PTB-XL dataset

Table A.2 provides detailed information on the PTB-XL dataset [18–20], which is a collection of 21,837 clinical 12-lead electrocardiograms (ECGs) taken from 18,885 patients. Each ECG recording lasts for 10 s, and all signals were collected at a sampling rate of 100 Hz, i.e., corresponding 1000 time steps per sample. During the experiments, a set of 71 possible labels was used in a multi-label setting to provide conditional information.

Appendix B. XResNet1d50 details

Table B.3 presents the XResNet1d(50) architecture details proposed in [22], along with the corresponding training hyperparameters. The architecture comprises four blocks with 3, 4, 6, and 3 layers, respectively, of one-dimensional convolutions with an expansion of four and strides of 1. The training procedure employed the Adam optimizer with a learning rate and weight decay of 1×10^{-3} for 100 epochs and a batch size of 64. Unlike the generated ECG data, the classifier was trained on cropped samples of 250 time steps each. During inference, the mean output probabilities were computed across seven different crops obtained by shifting the window through the signal using a stride of 125 time steps. The mean of the output probabilities was then used as the prediction for the entire sample.

Table B.3
XResNet1d50 hyperparameters.

Hyperparameter	Value
Block of layers	4
Layers in each block	[3,4,6,3]
Expansion	4
Stride	1
Optimizer	Adam
Learning rate	1×10^{-3}
Weight decay	1×10^{-3}
Batch size	64
Epochs	100

Table C.4
SSSD-ECG hyperparameters.

Hyperparameter	Value
Residual layers	36
Residual channels	256
Skip channels	256
Diffusion embedding dim. 1	128
Diffusion embedding dim. 2	512
Diffusion embedding dim. 3	512
Schedule	Linear
Diffusion steps T	200
B_0	0.0001
B_1	0.02
Optimizer	Adam
Loss function	MSE
Learning rate	2×10^{-4}
Batch size	4

Appendix C. *SSSD-ECG* details

Table C.4 provides detailed information on the architecture of the *SSSD-ECG* diffusion model, which consists of a network of 36 stacked residual layers with 256 residual and skip channels. For a comprehensive discussion of the model architecture, we refer the reader to [45]. The *SSSD-ECG* model uses a swish activation function over the second and third levels and has a three-level diffusion embedding in 128, 256, and 256 dimensions. To compute the initial S4 diffusion, we constructed a convolutional layer after the diffusion embedding to double the input's residual channel dimension. We then employed a second S4 layer that increased the conditional information and its inclusion in the input. The output was routed through a gated-tanh non-linearity, and a convolutional layer was used to project residual channels back to the channel dimensionality. Regarding the hyperparameters, we used 200 time steps on a linear schedule for diffusion setup, with a beta value ranging from 0.0001 to 0.02. We employed Adam as an optimizer with a learning rate of $2 \cdot 10^{-4}$. To learn the temporal dependencies of series in both directions for the S4 model, we used a single bidirectional S4 layer. Based on prior research [31], we used layer normalization and an internal state dimensionality of $N = 64$.

Appendix D. Baseline details

D.1. Wavegan* details

Table D.5 displays the architecture and training hyperparameters utilized in the implementation of WaveGAN*. The generator takes in a one-dimensional vector of 1000 data points sampled from a uniform distribution and passes it through five deconvolution blocks to generate an output signal consisting of 1000 time steps and 8 ECG leads. Each deconvolution block is composed of four layers: an upsampling layer, a padding layer, a 1D-convolution layer, and a ReLU activation function, in that order. The discriminator and generator models have a size of 50 each. In the conditional setting, we incorporated a conditional batch normalization at every convolution in the generator. This batch normalization takes in a batch of labels to condition, which is passed through an embedding layer with an embedding dimension size twice that of the output channel dimensions, and then added to the convolution output.

Table D.5
WaveGAN* hyperparameters.

Hyperparameter	Value
Generator model size	50
Generator deconvolutional blocks	5
Generator latent dimensions	1000
Discriminator model size	50
Discriminator convolutional blocks	6
Optimizer	Adam
Loss function	MSE
learning rate	0.0001
Training epochs	3000
Batch size	32

Table D.6
Pulse2Pulse hyperparameters.

Hyperparameter	Value
Generator model size	50
Generator deconvolutional blocks	5
Discriminator model size	50
Discriminator convolutional blocks	6
Optimizer	Adam
Loss function	MSE
learning rate	0.0001
Training epochs	3000
Batch size	32

D.2. Pulse2Pulse details

Table D.6 presents the architecture and training hyperparameters utilized in the Pulse2Pulse implementation. Pulse2Pulse is an architecture that is akin to a U-Net, but differs in its use of one-dimensional convolutional layers for generating electrocardiogram (ECG) signals. The generator takes an input consisting of 1000 time steps and 8 channels sampled from a uniform distribution. This input undergoes five downsampling blocks and five upsampling blocks, where the upsampling technique used is similar to that of WaveGAN*. The downsampling process involves a one-dimensional convolution and a Leaky ReLU activation function. For the conditional setting, we added a conditional batch normalization to every convolution in the generator. This batch normalization takes in a batch of labels to condition the output, which is then passed through a word embedding with an embedding dimension size of twice the output channel dimensions. Finally, the result of the word embedding is added to the convolution output.

D.3. Other baselines

In this final appendix, we would like to comment on several baseline models that we have tested for our use case. Specifically, we were unable to train functional generative models using TTS-GAN, TimeGAN, and cVAE_ECG.

The authors presented a GAN model called TTS-GAN [39], as an unconditional generative model for time series data. They conducted various experiments to test the model, including one where they generated electrocardiogram (ECG) data. We implemented TTS-GAN with varying depths of layers for the generator and discriminator, including 4, 6, 12, and 24 layers. Additionally, we used different latent dimensions for the generator, such as 128, 256, and 1000. Since our sequence length was 1000, we mainly used two settings. The first setting had a patch size of 200 and an embedding dimension of 5, while the second setting had values of 100 and 10, respectively. The training process consisted of 200 epochs with a batch size of 4 and a learning rate of 0.0001 and 0.0003 for the generator and discriminator, respectively.

In a similar vein, TimeGAN [42] is a well-known GAN model used for generating unconditional time series data. Its creators conducted various experiments to test its efficacy, albeit mostly on short signal lengths. We attempted to train the TimeGAN model using the authors' recommended default hyperparameters, including a 3-layer GRU

Table E.7
Turing test diagnosis evaluation.

Set	Labels	A(P)	B(P)
1	NORM, SR	True	False
2	NDT, SR	False	True
3	ABQRS, IMI, SR	True	False
4	NORM, SARRH	True	True
5	LAFB SR	False	True
6	NORM SBRAD	True	True
7	PACE	True	True
8	CLBBB, SR	True	False
9	LVH, SR, VCLVH	False	False
10	NORM	True	False
11	IRBBB, SR	True	False
12	ABQRS, NORM, SR	True	True
13	IRBBB, NORM, SR	True	False
14	NORM, STACH	False	True
15	IMI, SR	False	True
16	ISC_, LVH, SR	True	True
17	ABQRS, ASMI, SR	True	False
18	NST_, SR	False	True
19	NDT, NT_, SR	True	False
20	ABQRS, ASMI, IMI, SR	False	True
21	LVH, SR	False	False
22	AFIB, NST_	True	False

Table E.8
Turing test synthetic evaluation.

Set	Labels	A(T)	B(T)	A(P)	B(P)
1	NORM, SR	False	True	False	True
2	NDT, SR	False	True	False	False
3	ABQRS, IMI, SR	True	False	False	True
4	NORM, SARRH	True	False	False	False
5	LAFB SR	False	True	True	False
6	NORM SBRAD	True	False	False	True
7	PACE	False	True	False	True
8	CLBBB, SR	False	True	False	True
9	LVH, SR, VCLVH	False	True	False	True
10	NORM	True	False	False	False
11	IRBBB, SR	True	False	False	True
12	ABQRS, NORM, SR	True	False	False	False
13	IRBBB, NORM, SR	False	True	False	False
14	NORM, STACH	False	True	False	False
15	IMI, SR	True	False	False	False
16	ISC_, LVH, SR	False	True	False	False
17	ABQRS, ASMI, SR	True	False	False	False
18	NST_, SR	True	False	False	False
19	NDT, NT_, SR	True	False	False	False
20	ABQRS, ASMI, IMI, SR	False	True	False	True
21	LVH, SR	False	True	False	False
22	AFIB, NST_	True	False	False	True

module with 24 hidden dimensions, a batch size of 4, and 2000 training iterations. However, we were unable to observe any meaningful generated samples.

Variational autoencoders (VAEs) are a popular framework for generative modeling. cVAE_ECG [33] is a conditional generative model specifically designed for ECG data. However, it differs from our approach in that it requires beat segmentation as a preprocessing step, rather than generating continuous signals. Additionally, the label set space of their implemented dataset is relatively small. In our work, we used a learning rate of 0.001, a batch size of 4, and various convolutional filter dimensions, including 3, 5, 8, and 10, as well as convolutional kernels at length dimensions of 10, 50, 100, 500, and 1000. We also experimented with latent spaces of 10, 20, 50, 100, and 1000, with a conditional dimensionality of 71 given the dataset labels. However, despite these efforts, we were unable to produce reasonable reconstructions using the given model architecture.

Appendix E. Clinical turing test: Detailed results breakdown

Table E.7 presents information about the initial phase of the Turing test, which involved evaluating diagnostic predictions. The predictions for sample A and sample B were represented by A(P) and B(P), respectively. The objective was to determine whether the given set of labels accurately matched the represented sample. True was used to denote correct diagnoses, while false indicated incorrect diagnoses.

Table E.8 presents details regarding the synthetic evaluation that was conducted as the second part of the Turing test. In the table, A(T), B(T), A(P), and B(P) denote the true label for sample A or B, as well as the predicted labels for sample A or B, respectively. The value "true" indicates a synthetic sample, while "false" indicates a real sample.

References

[1] M. Tzanou, Health data privacy under the GDPR: Big data challenges and regulatory responses, in: Routledge Research in the Law of Emerging Technologies, Taylor & Francis, 2020.

[2] J. Sullivan, American Bar Association Health Law Section, HIPAA: A Practical Guide To the Privacy and Security of Health Data, in: Hein's ABA Archive Microfiche Collection, Health Law Section, American Bar Association, 2004.

[3] A. Abbas, S.U. Khan, A review on the state-of-the-art privacy-preserving approaches in the e-health clouds, IEEE J. Biomed. Health Inf. 18 (2014) 1431–1441.

[4] J.L. Fernández-Alemán, I.C. Señor, P.Á.O. Lozoya, A. Toval, Security and privacy in electronic health records: A systematic literature review, J. Biomed. Inform. 46 (3) (2013) 541–562.

[5] M.M. Farooqi, M.A. Shah, A. Wahid, A. Akhunzada, F. Khan, N. ul Amin, I. Ali, Big data in healthcare: A survey, in: Applications of Intelligent Technologies in Healthcare, Springer International Publishing, Cham, 2019, pp. 143–152.

[6] R. Kumar, W. Wang, J. Kumar, T. Yang, A. Khan, W. Ali, I. Ali, An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals, Comput. Med. Imaging Graph. 87 (2021) 101812.

[7] J. Xu, B.S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, J. Healthc. Inform. Res. 5 (1) (2020) 1–19.

[8] H. Yin, A. Mallya, A. Vahdat, J.M. Alvarez, J. Kautz, P. Molchanov, See through gradients: Image batch recovery via gradinversion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16337–16346.

[9] P. Madley-Dowd, R. Hughes, K. Tilling, J. Heron, The proportion of missing data should not be used to guide decisions on multiple imputation, J. Clin. Epidemiol. 110 (2019) 63–73.

[10] T. Shadbahr, M. Roberts, J. Stanczuk, J. Gilbey, P. Teare, S. Dittmer, M. Thorpe, R.V. Torne, E. Sala, P. Lio, M. Patel, A.-C. Collaboration, J.H.F. Rudd, T. Mirtti, A. Rannikko, J.A.D. Aston, J. Tang, C.-B. Schönlieb, Classification of datasets with imputed missing values: does imputation quality matter? 2022, arXiv preprint 2206.08478.

[11] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, 2018, arXiv preprint 1809.11096.

[12] P. Dhariwal, A. Nichol, Diffusion models beat GANs on image synthesis, in: Advances in Neural Information Processing Systems, Vol. 34, 2021, pp. 8780–8794.

[13] R. Child, S. Gray, A. Radford, I. Sutskever, Generating long sequences with sparse transformers, 2019, arXiv preprint 1904.1050ß.

[14] A.M. Delaney, E. Brophy, T.E. Ward, Synthesis of realistic ECG using generative adversarial networks, 2019, arXiv preprint 1909.09150.

[15] M. Seibold, A. Hoch, M. Farshad, N. Navab, P. Fürnstahl, Conditional generative data augmentation for clinical audio datasets, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VII, Springer, 2022, pp. 345–354.

[16] K. Falahkheirkhah, S. Tiwari, K. Yeh, S. Gupta, L. Herrera-Hernandez, M.R. McCarthy, R.E. Jimenez, J.C. Cheville, R. Bhargava, Deepfake histologic images for enhancing digital pathology, Laboratory Investigation 103 (1) (2023) 100006.

[17] E.J. Topol, What's lurking in your electrocardiogram? Lancet 397 (10276) (2021) 785.

[18] P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F.I. Lunze, W. Samek, T. Schaeffter, PTB-XL, a large publicly available electrocardiography dataset, Sci. Data 7 (1) (2020) 154.

[19] P. Wagner, N. Strodthoff, R.-D. Bousseljot, W. Samek, T. Schaeffter, PTB-XL, a large publicly available electrocardiography dataset, 2020.

[20] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet, Circulation 101 (23) (2000) e215–e220.

[21] T. Mehari, N. Strodthoff, Advancing the state-of-the-art for ECG analysis through structured state space models, 2022, arXiv:2211.07579, extended abstract.

[22] N. Strodthoff, P. Wagner, T. Schaeffter, W. Samek, Deep learning for ECG analysis: Benchmarks and insights from PTB-XL, IEEE J. Biomed. Health Inf. 25 (5) (2021) 1519–1528.

[23] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, M. Li, Bag of tricks for image classification with convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 558–567.
[24] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: Proceedings of the 32nd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, Vol. 37, 2015, pp. 2256–2265.
[25] N. Chen, Y. Zhang, H. Zen, R.J. Weiss, M. Norouzi, W. Chan, WaveGrad: Estimating gradients for waveform generation, in: International Conference on Learning Representations, 2020.
[26] Z. Kong, W. Ping, J. Huang, K. Zhao, B. Catanzaro, DiffWave: A versatile diffusion model for audio synthesis, in: 9th International Conference on Learning Representations, ICLR 2021, 2021.
[27] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 6840–6851.
[28] J. Ho, C. Saharia, W. Chan, D. Fleet, M. Norouzi, T. Salimans, Cascaded diffusion models for high fidelity image generation, J. Mach. Learn. Res. 23 (2022) 47:1–47:33.
[29] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
[30] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, D.J. Fleet, Video diffusion models, 2022, arXiv preprint 2204.03458.
[31] A. Gu, K. Goel, C. Ré, Efficiently modeling long sequences with structured state spaces, in: International Conference on Learning Representations, 2022, arXiv:2111.00396.
[32] A. Gu, T. Dao, S. Ermon, A. Rudra, C. Ré, HiPPO: Recurrent memory with optimal polynomial projections, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 1474–1487.
[33] Y. Sang, M. Beetz, V. Grau, Generation of 12-lead electrocardiogram with subject-specific, image-derived characteristics using a conditional variational autoencoder, in: 2022 IEEE 19th International Symposium on Biomedical Imaging, ISBI, 2022, pp. 1–5, http://dx.doi.org/10.1109/ISBI52829.2022.9761431.
[34] M.A. Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, R. Yager, Deep learning approach for active classification of electrocardiogram signals, Inform. Sci. 345 (2016) 340–354.
[35] V. Thambawita, J.L. Isaksen, S.A. Hicks, J. Ghouse, G. Ahlberg, A. Linneberg, N. Grarup, C. Ellervik, M.S. Olesen, T. Hansen, C. Graff, N.-H. Holstein-Rathlou, I. Strümke, H.L. Hammer, M.M. Maleckar, P. Halvorsen, M.A. Riegler, J.K. Kanters, DeepFake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine, Sci. Rep. 11 (1) (2021) 21896.
[36] F. Zhu, F. Ye, Y. Fu, Q. Liu, B. Shen, Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network, Sci. Rep. 9 (2019).
[37] A.M. Delaney, E. Brophy, T.E. Ward, Synthesis of realistic ECG using generative adversarial networks, 2019, arXiv preprint 1909.09150.
[38] T. Golany, G. Lavee, S. Tejman Yarden, K. Radinsky, Improving ECG classification using generative adversarial networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, (08) 2020, pp. 13280–13285, http://dx.doi.org/10.1609/aaai.v34i08.7037.
[39] X. Li, V. Metsis, H. Wang, A.H.H. Ngu, TTS-gan: A transformer-based time-series generative adversarial network, in: M. Michalowski, S.S.R. Abidi, S. Abidi (Eds.), Artificial Intelligence in Medicine, Springer International Publishing, Cham, 2022, pp. 133–143.
[40] T. Golany, K. Radinsky, D. Freedman, SimGANs: Simulator-based generative adversarial networks for ECG synthesis to improve deep ECG classification, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, Vol. 119, PMLR, 2020, pp. 3597–3606.
[41] T. Golany, K. Radinsky, PGANs: Personalized generative adversarial networks for ECG synthesis to improve patient-specific deep ECG classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, (01) 2019, pp. 557–564, http://dx.doi.org/10.1609/aaai.v33i01.3301557.
[42] J. Yoon, D. Jarrett, M. van der Schaar, Time-series generative adversarial networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.
[43] E. Adib, A. Fernandez, F. Afghah, J.J. Prevost, Synthetic ECG signal generation using probabilistic diffusion models, 2023, arXiv:2303.02475.
[44] H. Chung, J. Kim, J.-m. Kwon, K.-H. Jeon, M.S. Lee, E. Choi, Text-to-ecg: 12-lead electrocardiogram synthesis conditioned on clinical text reports, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023, pp. 1–5.
[45] J.L. Alcaraz, N. Strodthoff, Diffusion-based time series imputation and forecasting with structured state space models, Trans. Mach. Learn. Res. (2022).
[46] J.M.L. Alcaraz, N. Strodthoff, SSSD-ECG public code repository, https://zenodo.org/account/settings/github/repository/AI4HealthUOL/SSSD-ECG, (Accessed: 2022-12-31).
[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[48] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, A.C. Courville, Modulating early visual processing by language, Adv. Neural Inf. Process. Syst. 30 (2017).
[49] A. Alaa, B. Van Breugel, E.S. Saveliev, M. van der Schaar, How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, Vol. 162, PMLR, 2022, pp. 290–306.
[50] J.M.L. Alcaraz, N. Strodthoff, SSSD-ECG data repository, https://figshare.com/s/43df16e4a50e4dd0a0c5, (Accessed: 2022-01-19).