Text-to-ECG: a Framework to Generate 12-Lead ECG from Text Reports

Fabrizio Cattozzo[1], Sara Battiston[1], Massimo W. Rivolta[1] and Roberto Sassi[1]

***Abstract*— Synthetic electrocardiograms (ECGs), obtained with Generative Artificial Intelligence (GenAI), are currently used to support the training of other AI algorithms, most often decision support systems, by augmenting the dataset for the minority classes. In this study, we proposed a Text-to-ECG (T2ECG) framework, which could generate synthetic ECG beats from textual data. The framework made use of two components. The first, Bio_ClinicalBERT, produced an embedded vector from the input text (e.g., "left ventricular hypertrophy"). Then, a second component leveraged such representation to generate a 12-lead ECG heartbeat by means of a Wasserstein Generative Adversarial Network with gradient penalty. The training was performed on the PTB-XL dataset, freely available on Physionet. The framework was designed to generate five different diagnostic classes: i) normal sinus rhythm; ii) inferior myocardial infarction (IMI); iii) antero-septal myocardial infarction (ASMI); iv) left anterior fascicular block (LAFB); and v) left ventricular hypertrophy (LVH). The realism of the generated signals was assessed through three different methodologies, involving both visual inspection and quantitative analyses. Our results show that the T2ECG framework was able to generate heartbeats of sufficient quality, except for the the ASMI class. In conclusion, the framework proposed does not only support data augmentation but also facilitates the creation of ECGs by non-technical users, offering a textual interface to the GenAI model.**

I. INTRODUCTION

Generative Artificial Intelligence (GenAI) is widely employed for chatbots and image or video generation, with an impact that quickly affected not only the scientific field of computational intelligence, but also many applications at global scale. However, in the context of the generation of synthetic electrocardiograms (ECGs), GenAI is being currently used to support the creation of other AI algorithms (*e.g.*, [1], [2], most often decision support systems), with the primary scope of augmenting the size of the training set for poorly represented diagnostic cases [3]. Therefore, GenAI in the context of synthetic ECGs still needs to find its applicative scope to reach non-technical end-users.

Recently, Chung *et al.* [4] proposed a Deep Learning (DL) technique able to generate synthetic ECGs from medical text reports. Despite the motivation of the work was to mitigate the lack of annotated ECG datasets, we believe that such Text-to-ECG (T2ECG) applications could open possibilities not currently explored and make accessible synthetic ECG generation to non-technical users. For example, T2ECG may foster the education of the future generation of cardiologists, or facilitate the clinical practice by retrieving information about specific ECG morphologies.

Most GenAI methodologies for ECG synthesis are based on Generative Adversarial Networks (GANs) [5]. GANs rely on two neural networks that compete in an adversarial process. The two networks are a *generator*, whose goal is to generate data as similar as possible to the real data from a noise vector, and a *discriminator*, trained to distinguish real from synthetic data. However, the training of GANs is often complicated due to vanishing gradients, mode collapse and the adversarial training itself. To address this issue, Wasserstein GANs (WGANs) [6] and WGANs with gradient penalty (WGAN-GP) [7] were recently introduced. This new framework maintains the generator-discriminator architecture of GANs, while leading to a more reliable training.

In this study, we investigated the use of a WGAN-GP for designing a T2ECG pipeline able to synthesize 12-lead ECG heartbeats conditioned on textual data. We focused on the generation of signal of 5 different diagnostic classes, each impacting the morphology of the QRTS complex.

II. METHODS

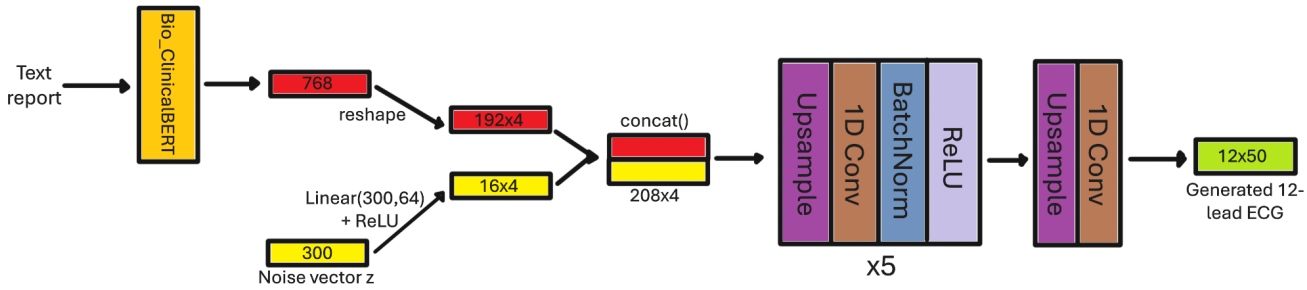
A. Dataset and preprocessing

The data employed in this study were taken from the PTB-XL dataset [8], [9], one of the largest freely accessible clinical 12-lead ECG collections on Physionet. It comprises 21837 10s diagnostic ECGs from 18885 patients, with 71 different diagnoses. Two versions of the same ECG were available, one sampled at 500 Hz, and one downsampled at 100 Hz. For the purpose of this study, we considered only the ECG signals downsampled at 100 Hz. Furthermore, we selected patients with the following diagnoses: i) normal sinus rhythm (NORM); ii) inferior Myocardial Infarction (IMI); iii) antero-Septal Myocardial Infarction (ASMI); iv) left anterior fascicular block (LAFB); and v) left ventricular hypertrophy (LVH).

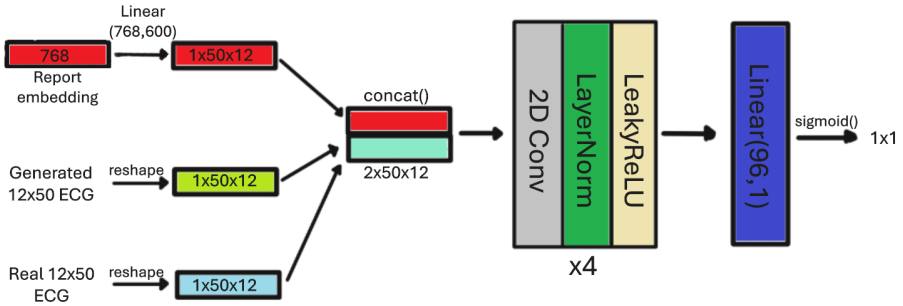
Signals were band-pass filtered with a zero-phase fourth-order Butterworth filter with cut-off frequencies of 0.5-40 Hz. Then, the *gqrs_detect* function of the *wfdb* [10] Python package was used to detect the location of the QRS complexes. Each signal was segmented into "heartbeats" (ECG segments of 50 samples or 500 ms, extracted from 150 ms before to 350 ms after the R peak, corresponding approximately to the entire PQRTS duration). We then randomly selected a subset of the patients for each class to obtain a dataset approximately balanced. The final dataset

This work was supported by the project FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence).

[1]Authors are all affiliated with the Department of Computer Science, Università degli Studi di Milano, Via Celoria 18, 20133, Milan, Italy. Corresponding author: massimo.rivolta@unimi.it



(a)



(b)

Fig. 1: Proposed AI pipeline for Text-to-ECG generation. (a) Full model architecture from text report to synthetic ECG. Convolutional parameters of the generative model are reported in Table II. (b) Model architecture of the "critic" in the WGAN-GP methodology.

contained 29,391 12-lead heartbeats across the five classes stored into 12×50 matrices (see Table I for details).

Due to the large presence of non-English text reports in the PTB-XL dataset, original reports were replaced with English ones based on the diagnostic class. To do so, we manually created a list of approximately 25 equivalent possible reports for each class by translating or rephrasing into English language some of the existing reports, and by consulting online medical repositories. An example of a text report was "left ventricular hypertrophy". The 75th percentile of the length of the report was 47 characters, while the longest had 108 characters. All heartbeats in a given diagnostic class was then assigned a randomly selected report from the corresponding list for that class.

B. AI Pipeline

The AI pipeline of our method is depicted in figure 1 and further detailed in table II; it involved mainly two components. The first one was a DL model meant to embed the input text into a single vector. The second component was instead a conditional generator taking in input the embedded text and a random noise to generate the output ECG. The text reports were embedded through the Bio_ClinicalBERT model [11], *i.e.*, a well-known DL model pretrained on the notes of the MIMIC-III dataset [12], which comprised also ECG reports. With Bio_ClinicalBERT, all text reports were embedded to vector of fixed size (768 dimensions). Bio_ClinicalBERT was used *as-is*, without any further fine-tuning on the reports of our dataset.

The embedded report, together with a noise vector z sampled from a uniform distribution of size 300 (selected empirically), served as inputs to the generative model, which was based on convolutional layers (see figure 1a). The output of the model was a 12×50 matrix representing the generated ECG for each of the 12 leads.

TABLE I: Number of heartbeats for each class of the dataset.

Diagnostic class	No. of heartbeats
NORM	5856
IMI	6000
ASMI	5640
LVH	6096
LAFB	5799

C. Training

The training of the generator was based on a WGAN-GP approach. It required the use of a model called "critic" which goal was to identify real signals from generated ones. Its architecture is reported in figure 1b and additional details are reported in Table III. Specifically, the critic was meant to minimize the following loss, where D and G are the differentiable functions associated to the critic and generator respectively, λ is a regularization constant, x is the 12 lead

ECG segment, y is the vector embedding the text report, p_z is the noise distribution (uniform in our case) and p_{data} is the distribution of the dataset:

$$L = \mathbb{E}_{\substack{(x,y)\sim p_{data} \\ z\sim p_z \\ u\sim U(0,1)}} \Big[D(G(z|y)|y) - D(x|y) + \lambda \Big(||\nabla D(v|y)||_2 - 1 \Big)^2 \Big]. \quad (1)$$

Also, $\nabla D(v|y)$ is, with a little abuse of notation, the gradient of D with respect to all its inputs then evaluated at v. The vector v was defined as a random sample extracted from the line connecting the vector x and the generated vector $G(z|y)$. Formally, $v = [G(z|y) + (x - G(z|y)]u$ with $u \sim U(0, 1)$. The generator was instead trained to minimize:

$$L = -\mathbb{E}_{z\sim p_z} \Big[D(G(z|y)|y) \Big]. \quad (2)$$

The weights of both the generator and critic were initialized using values drawn from a normal distribution.

After extensive experiments to train both networks, the final training was carried out with a batch size of 1024 and over 83 epochs. The critic was updated 100 times each epoch to reach its optimal state. In our experiment, the optimal value of the λ constant was found to be 10. Both the generator and critic were optimized with the Adam optimizer [13] with a learning rate of 0.0001 and β values of 0.9 and 0.999, respectively. The hyperparameters of the batch normalization layers were set to $\epsilon = 0.0001$ and $\text{momentum} = 0.001$. To prevent the critic from overpowering the generator during training, inspired by [14], Gaussian noise with a standard deviation of 0.001 mV was added to both the fake and real signals fed into the critic. The training process took approximately 200 minutes to complete on a NVIDIA L40S GPU.

D. Assessment of the generated ECGs

The generated signals were evaluated both visually and quantitatively to assess their quality and diversity using three different approaches, as similarly performed in [15].

UMAP algorithm [16]: for each diagnostic class, the ECG signals in the real and generated datasets were projected to a bi-dimensional plane by using the well-known UMAP algorithm trained on all classes together. The probability density function for the projections of the real signals for each diagnostic class was obtained using kernel density estimation and plotted using heatmaps. The scatterplot of the synthetic ECG projections of the corresponding class was plotted over each heatmap. In this way, we visually checked whether the two distributions on the 2D plane were overlapping.

GAN-train and GAN-test [17]: this technique provided two scores called GAN-train and GAN-test. Briefly, for GAN-train we trained an ECG classifier on a synthetic dataset generated by our T2ECG generator and tested it on both synthetic and real data, comparing the performances between them. GAN-test trained instead the same classifier but on a real dataset; the classifier was then assessed on a test set comparing performance between real and generated signals. The GAN-train score is supposed to provide a measure of how diverse the generated data are, while the GAN-test score indicates how close the generated data are to the original data manifold. The classifier adopted in this work used the same architecture in [18].

ECG signal distribution plot: we computed the 90% confidence interval for each time sample of the 12-lead ECG signals for both real and synthetic ECGs, and plotted them together for each diagnostic class. The test was meant to verify whether there was overlapping between the real and generated data. In addition, we computed the intersection over union (IoU) of these two bands to quantify such overlap. Areas were quantified with the rectangular rule.

Details about the training of the GAN-test/GAN-train classifiers and UMAP are reported in the Appendix.

III. RESULTS AND DISCUSSIONS

An example of an ECG signal generated by T2ECG with the input text "antero-septal myocardial infarction" is reported in Fig. 2. The signals resembled quite nicely the overall morphology of a single heartbeat in this diagnostic class.

The results obtained using UMAP are displayed in Fig. 3. In the bi-dimensional plane, for NORM (Fig. 3a) most projections of generated and real signals shared the same area, except few generated signals which behaved differently. A possible explanation for the latter is the loss of information due to dimensionality reduction or due to inaccuracies

TABLE II: Parameters for each of the five repeated block (composed of Upsample, 1D Conv, BatchNorm and ReLU) in the generator. Each row corresponds to one block in the generator, detailing the upsample size, number of 1D convolution filters, and convolutional layer parameters (kernel size, stride, padding). The 6th row corresponds to the block following the repeated ones.

Block	Upsample size	N° of Conv filters	Conv parameters
1st	7	128	(3, 1, 1)
2nd	17	64	(5, 1, 2)
3rd	25	32	(5, 1, 2)
4th	37	24	(5, 1, 2)
5th	47	16	(3, 1, 1)
-	50	12	(3, 1, 1)

TABLE III: Parameters for the 2D convolution in the critic implementation. Each row corresponds to one block in the critic, detailing, respectively, the number of filter, the 2D kernel size, the horizontal and vertical stride and the padding on the two axis

Block n#	n# of filters	(k_x, k_y)	(s_x, s_y)	(p_x, p_y)
1st	4	$(4, 4)$	$(2, 2)$	$(2, 2)$
2nd	8	$(5, 3)$	$(2, 1)$	$(1, 0)$
3rd	16	$(4, 3)$	$(2, 1)$	$(1, 0)$
4th	32	$(3, 3)$	$(2, 1)$	$(1, 0)$

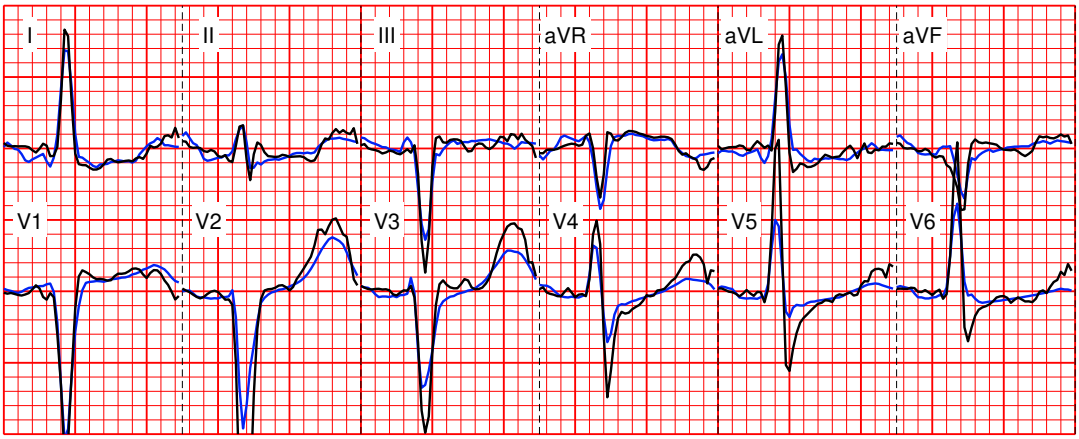


Fig. 2: Example of 12-lead ECG generated from the text "antero-septal myocardial infarction" (black) and a real one from the same class selected from the dataset (blue). For visualization reason, we selected, through crosscorrelation, the real ECG which most closely matched the generated one. Thick horizontal lines are spaced 0.2 s while thick vertical lines 0.5 mV.

introduced by the generative framework. For LAFB (Fig. 3b) and LVH (Fig. 3e) the projections of the generated signals laid again on the same areas of the real signals, but with less variability. For IMI, the synthetic signals projections did not consistently lay in the expected region. This occurred for the ASMI class too, when the generated ECG signals projections were wrongly located where the IMI signals should be (Fig. 3c vs 3d).

The GAN-train and GAN-test scores of all experiments are reported in Table IV, along with the number of heartbeats used. The GAN-test results (bottom) indicated that the generator, overall, produced ECG signals closely resembling real data, as evidenced by a minimal accuracy drop when tested on generated versus real signals. In contrast, the larger accuracy drop observed in GAN-train (top) suggests that the generator samples from a narrower distribution than the real data distribution, resulting in reduced signal diversity. In particular, the generator seemed to approximate the real distributions of the NORM and LAFB classes well, while the poor GAN-train results for the LVH class suggest that it was reasonably captured but with smaller diversity. The IMI and ASMI classes, however, were approximated by distributions that did not completely match the real ones and had smaller variability because of their poor GAN-train and GAN-test scores. However, it is worth noting that interpreting GAN-train and GAN-test scores when the classifier performance is poor (trained and tested on real data) still requires further investigation.

Figure 4 shows the 90% confidence bands for the ASMI diagnostic class. In most leads, the generated signals laid within the expected bands, as supported by UMAP and GAN-train scores, and displayed a mean ($\pm$ std) IoU across leads of 0.76 ± 0.05. Similar results were obtained for other diagnostic classes. Table V reports IoU across leads for each class.

To the best of our knowledge, the only other Text-to-ECG framework available in the literature is the one proposed by Chung *et al.* [4]. Three major differences are present between their approach and ours. First, they designed a DL model able to synthesize 10 s ECGs rather than single heartbeats. Second, they trained their model on the entire PTB-XL and also on a private dataset. Despite this task was more challenging than what we aimed to tackle, in particular during the training phase given that the dataset is highly imbalanced for certain diagnosis, they eventually evaluated the quality of generated ECGs only for 6 different diagnostic classes, which was then similar to what done in our study. Finally, their method was based on the temporal quantization of the ECGs and the use of an autoregressive DL model to generate the synthetic signals. This technique is very different from what we proposed in here since we restrained from degrading the temporal resolution of the ECGs.

Determining the overlap between the probability densities estimated via UMAP and the projected synthetic ECGs is

TABLE IV: Classification results of the GAN-train (upper) and GAN-test (lower) classifiers. Support refers to the number of heartbeat used.

Class	Tested on fake data		Tested on real data	
	F1 Score	Support	F1 Score	Support
IMI	0.83	394	0.43	6017
NORM	0.98	366	0.78	5906
LVH	0.94	386	0.56	6054
LAFB	0.96	389	0.68	5822
ASMI	0.88	385	0.54	5716
Accuracy	0.92		0.60	

Class	Tested on real data		Tested on fake data	
	F1 Score	Support	F1 Score	Support
IMI	0.81	1183	0.56	2000
NORM	0.96	1182	0.86	2000
LVH	0.88	1253	0.78	2000
LAFB	0.82	1148	0.84	2000
ASMI	0.71	1137	0.55	2000
Accuracy	0.84		0.72	

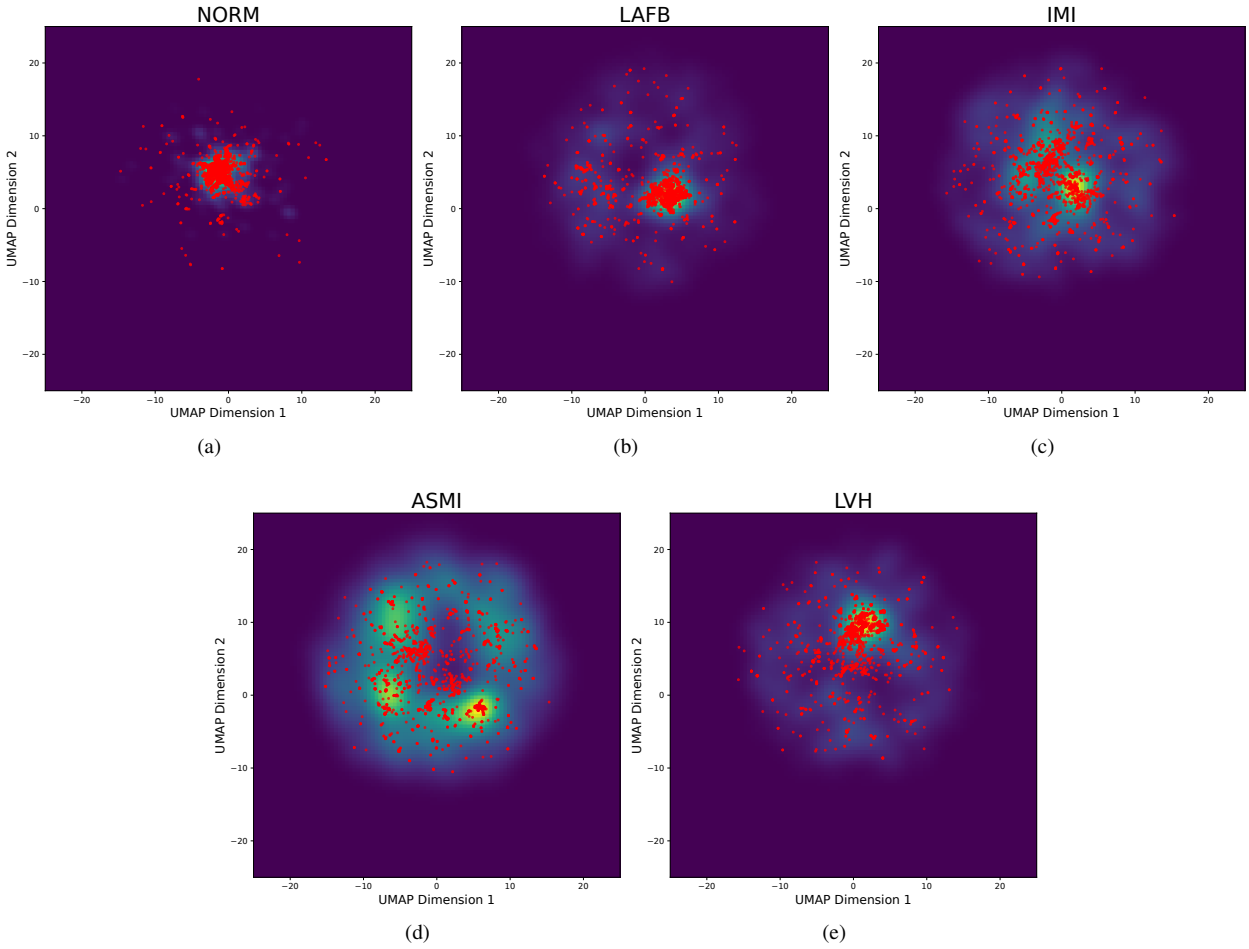


Fig. 3: UMAP projections of generated ECG signals (red dots) overlapped to heatmaps of the distribution of the real signals' projections.

TABLE V: Mean and standard deviation of the IoU across leads, for each diagnostic class.

Diagnostic class	IoU (mean $\pm$ std)
NORM	0.78 ± 0.02
IMI	0.80 ± 0.02
ASMI	0.76 ± 0.05
LVH	0.73 ± 0.05
LAFB	0.76 ± 0.03

technically feasible. For example, one may consider the log-likelihood as a measure of goodness. However, this approach would suffer from two major problems. The first one is mode collapse. When the model outputs only the most frequent ECGs for all input text, the likelihood is maximum but the model is not able to generate sufficiently diverse signals. The second, more crucial point, is that such quantification entails the selection of a metric. However, selecting a proper metric is challenging, if not impossible, since knowing the metric, one could train a generator to optimize it. To conclude, the evaluation of the quality of the generated ECGs is still an open problem.

IV. CONCLUSIONS

In this study, we implemented a Text-to-ECG pipeline to generate ECG heartbeats directly from short text reports describing the cardiac condition. Despite the pipeline still requires further improvements, the model showed satisfactory results with respect to different metrics. Future studies will investigate the possibility of augmenting the number of cardiac conditions as well as the generation of a longer 10 s diagnostic ECG signals. The short text reports on which the framework was trained can be enriched to include detailed patient characteristics such as sex, age, electrode placement and other relevant information, to increase the personalization and clinical accuracy of generated signals. Finally, the involvement of cardiologists to validate the realism and fidelity of generated signals, while complex to set up could strengthen the reliability of the framework.

APPENDIX

A. *GAN-train*

The ECG classifier was trained with a batch size of 256 for 10 epochs, with learning rate of 0.001 and using Adam optimizer with β values of 0.9 and 0.999. A set of $10,000$

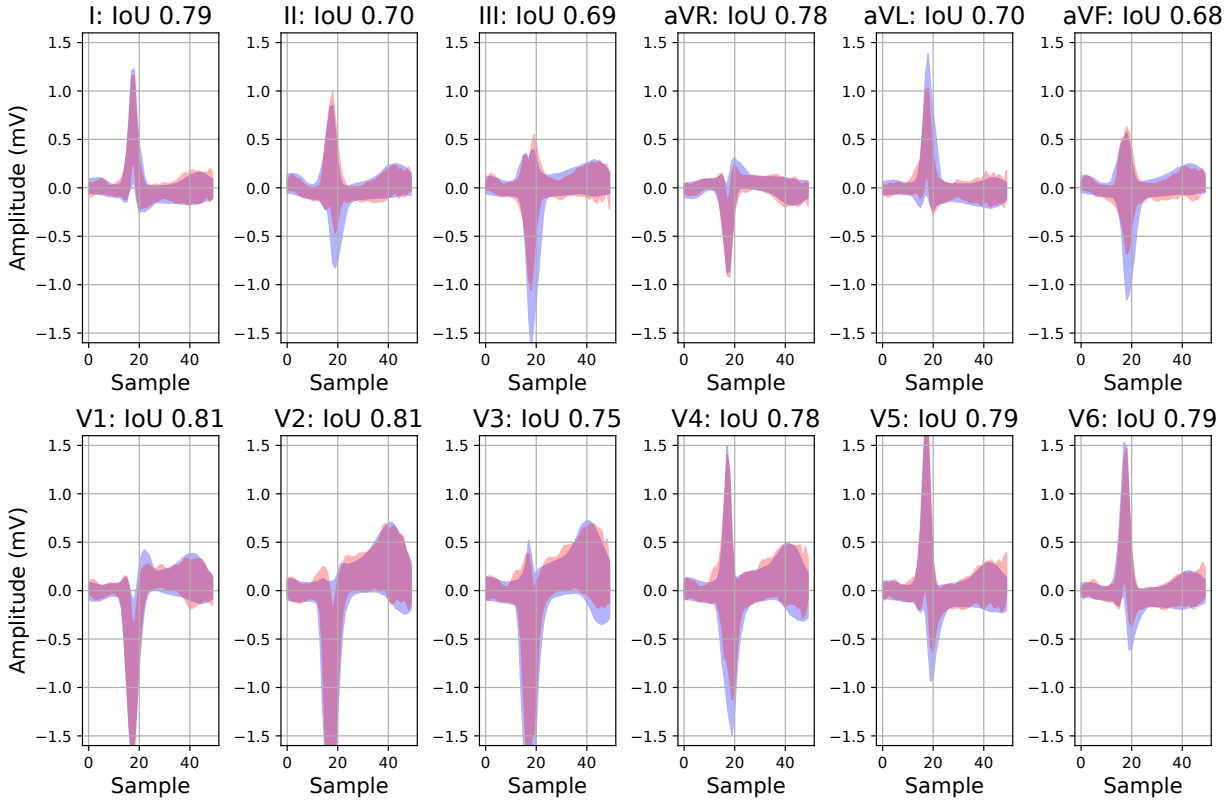


Fig. 4: 90% confidence band of 12-lead ECG signals of the real dataset (in blue) and the generated dataset (in red) for class ASMI.

synthetic samples, *i.e.*, $2,000$ for each diagnostic class, was generated using our T2ECG pipeline. Four percent of these data was used as a validation dataset to determine the optimal number of epochs to train the classifier. From the $9,600$ remaining samples in the training dataset, $1,920$ were used as a test dataset and the remaining for traning. The classifier weights were initialized using Xavier initialization [19].

B. GAN-test

The ECG classifier was trained on a new random balanced sampling of the original dataset (see steps described in sec. II-A) to avoid the possibility of using heartbeats of the same patient in the construction of the WGAN generator and GAN-test classifier. Out of the new $29,515$ total signals, $5,903$ were set as the test dataset, $4,723$ were placed in the validation dataset and the remaining $18,889$ signals formed the training dataset. The classifier was trained with a batch size of 256 for 40 epochs, with learning rate of 0.001 and using Adam optimizer with β values of 0.9 and 0.999, respectively. The classifier weights were initialized using Xavier initialization.

C. UMAP

Each 12×50 heartbeat signal was reshaped to a 1×600 vector by concatenating each row. Subsequently, the UMAP algorithm was trained on all the real ECG dataset's signals. UMAP parameters were: neighbors $= 15$, Euclidean distance, min_distance$= 0.1$, n_components $= 1$.

References

[1] F. Cao, A. Budhota, H. Chen, and K. S. Rajput, "Feature matching based ECG generative network for arrhythmia event augmentation," in *Annu Int Conf IEEE Eng Med Biol Soc*, 2020, pp. 296–299.

[2] E. Brophy, M. De Vos, G. Boylan, and T. Ward, "Multivariate generative adversarial networks and their loss functions for synthesis of multichannel ECGs," *IEEE Access*, vol. 9, pp. 158 936–158 945, 2021.

[3] M. M. Rahman, M. W. Rivolta, F. Badilini, and R. Sassi, "A systematic survey of data augmentation of ECG signals for AI applications," *Sensors*, vol. 23, no. 11, p. 5237, 2023.

[4] H. Chung, J. Kim, J.-M. Kwon, K.-H. Jeon, M. S. Lee, and E. Choi, "Text-to-ECG: 12-lead electrocardiogram synthesis conditioned on clinical text reports," in *Proc IEEE Int Conf Acoust Speech Signal Process*, 2023, pp. 1–5.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun ACM*, vol. 63, no. 11, p. 139–144, Oct. 2020.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc 34th Int Conf Mach Learn*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 214–223.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Adv Neural Inf Process Syst*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 5769–5779.

[8] P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter, "PTB-XL, a large publicly available electrocardiography dataset," *Sci Data*, vol. 7, no. 1, p. 154, 2020.

[9] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[10] C. Xie, L. McCullum, A. Johnson, T. Pollard, B. Gow, and B. Moody, "Waveform Database Software Package (WFDB) for Python (version 4.1.0)," 2023, PhysioNet. DOI: 10.13026/9njx-6322.

[11] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical BERT embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

[12] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Sci Data*, vol. 3, no. 1, pp. 1–9, 2016.

[13] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, pp. 1–15, 2014.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Adv Neural Inf Process Syst*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016, pp. 2234–2242.

[15] S. Battiston, R. Sassi, and M. W. Rivolta, "Evaluating the quality of CycleGAN generated ECG data for myocardial infarction classification," in *Comput Cardiol*, vol. 51, 2024.

[16] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2020.

[17] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my GAN?" in *Comput Vis – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Springer International Publishing, 2018, pp. 218–234.

[18] A. H. Ribeiro, M. H. Ribeiro, G. M. M. Paixão, D. M. de Oliveira, P. R. Gomes, J. A. Canazart, M. P. S. Ferreira, C. R. Andersson, P. W. Macfarlane, W. M. Jr., T. B. Schön, and A. L. P. Ribeiro, "Automatic diagnosis of the 12-lead ECG using a deep neural network," *Nat Commun*, vol. 11, no. 1, p. 1760, 2019.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J Mach Learn Res - Proc Track*, vol. 9, pp. 249–256, 2010.