ECG ODE-GAN: Learning Ordinary Differential Equations of ECG Dynamics via Generative Adversarial Learning

Tomer Golany,[1] Daniel Freedman, [2] Kira Radinsky [1]
[1] Technion-Israel Institute of Technology
[2] Google Research
tomer.golany@cs.technion.ac.il, danielfreedman@google.com, kirar@cs.technion.ac.il

Abstract

Understanding the dynamics of complex biological and physiological systems has been explored for many years in the form of physically-based mathematical simulators. The behavior of a physical system is often described via ordinary differential equations (ODE), referred to as the dynamics. In the standard case, the dynamics are derived from purely physical considerations. By contrast, in this work we study how the dynamics can be learned by a generative adversarial network which combines both physical and data considerations. As a use case, we focus on the dynamics of the heart signal electrocardiogram (ECG). We begin by introducing a new GAN framework, dubbed ODE-GAN, in which the generator learns the dynamics of a physical system in the form of an ordinary differential equation. Specifically, the generator network receives as input a value at a specific time step, and produces the derivative of the system at that time step. Thus, the ODE-GAN learns purely data-driven dynamics. We then show how to incorporate physical considerations into ODE-GAN. We achieve this through the introduction of an additional input to the ODE-GAN generator: physical parameters, which partially characterize the signal of interest. As we focus on ECG signals, we refer to this new framework as ECG-ODE-GAN. We perform an empirical evaluation and show that generating ECG heartbeats from our learned dynamics improves ECG heartbeat classification.

1 Introduction

Electrocardiography (ECG) is a non-invasive tool, used widely by cardiologists for monitoring cardiac health. Cardiovascular diseases are the cause of death for about one-third of all deaths around the globe, therefore analyzing ECG heartbeats is of significant importance. In recent years, deep learning algorithms have been applied for the problem, yielding state-of-the-art results (Kachuee, Fazeli, and Sarrafzadeh 2018). While the performance of such algorithms is quite promising, they are purely data-driven. For a deep learning model to succeed in its task, a large amount of annotated data is needed. However, analyzing and labeling ECG signals manually is prone to errors and consumes expensive experts time. To address this problem, (Golany et al. 2020) introduced a Generative Adversarial Network (GAN) framework that learns to generate synthetic ECG heartbeats. They

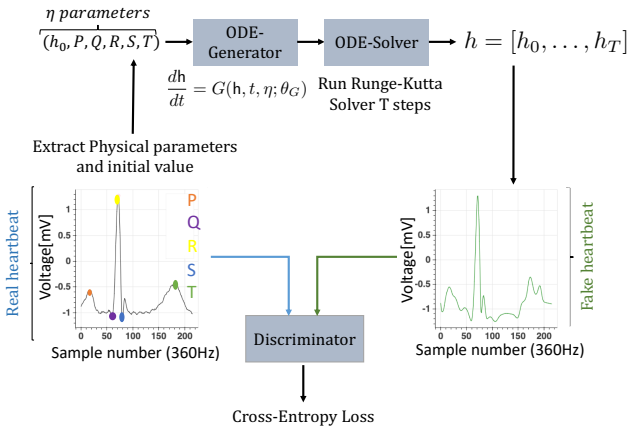


Figure 1: ECG-ODE-GAN pipeline. The ODE-Generator receives from a real heartbeat its physical parameters and its initial voltage value. The ODE resulting from the ODE-Generator is solved numerically, yielding a generated heartbeat. The discriminator tries to distinguish between real and generated heartbeats.

show that the combination of the synthetic heartbeats with a small number of labeled heartbeats significantly improves the ECG heartbeat classifier's performance.

Injecting prior knowledge into learners is a principled way to improve classifier performance. For ECG heartbeats classification, knowledge is usually integrated via feature engineering (De Chazal, O'Dwyer, and Reilly 2004). Recently, (Golany, Freedman, and Radinsky 2020) demonstrated integration of knowledge using a physically motivated ECG simulator, given by a set of ordinary differential equations (McSharry et al. 2003). They presented a Generative Adversarial Network (Goodfellow et al. 2014) that not only generates ECG signals that resemble real ECG examples, but also resemble ones that are generated from the physical simulator. In this work, we focus on the question of whether a better physical simulator can be learned. We present a methodology in which GANs are used to learn a better dynamical system, defined by ordinary differential equations, for describing electrical activity in the heart as measured by ECGs. Our model combines data-driven knowledge with physical

knowledge and learns a family of dynamical systems which describes the entire class of ECG signals.

Our contributions are twofold. First, we propose a new GAN setting, which we call ODE-GAN. In this setting, the generator model is an ordinary differential equation: $d\mathsf{x}/dt = G(\mathsf{x}, t, z; \theta_G)$. The idea is that the generator learns the dynamics of a dynamical system $\mathsf{x}(t)$. As opposed to the common GAN setting in which the generator outputs the entire signal at once, ODE-GAN gets as input a value $\mathsf{x}(t)$ and learns to generate the derivative of $\mathsf{x}(t)$ w.r.t t. Applying a numerical integration scheme to the generator then yields the synthetic ECG heartbeat. To the best of our knowledge, this is the first attempt to train a GAN to learn a dynamical system in this manner. Second, we show how to incorporate physical considerations into the ODE-GAN, by making the generator depend on physical parameters which characterize the signals or dynamical system of interest. We apply this to ECGs, yielding ECG-ODE-GAN: the physical parameters in this instance are morphological descriptors of the ECG signal, and the architecture of ECG-ODE-GAN reflects their particular form. We empirically show that utilizing the synthetically generated ECG heartbeats generated from ECG-ODE-GAN significantly improves ECG heartbeat classification using deep learning techniques.

The structure of our paper is as follows: Section 2 reviews the background and related work. Section 3 introduces the ODE-GAN framework, while Section 4 introduces the combined physics / data-drive framework of ECG-ODE-GAN. Section 5 describes how the generated synthetic heartbeats from the ECG-ODE-GAN are used to train a deep network for ECG classification. Finally, in Sections 6-7 we present empirical evaluation, comparing with state-of-the-art methods. All of our code will be shared online.

2 Related Work

The combination of deep neural networks with ODEs was first introduced by ODE-Nets (Chen et al. 2018). Those networks have been shown to be more memory efficient than networks using backpropagation during training. Specifically, it was shown that residual networks represented by a discrete sequence of hidden layers, can be parameterized by the derivative of the hidden states using a neural network. The output of the ODE network is computed using a numerical ODE-solver. They showed that computing the gradients of a loss function with respect to the network weights can be done with constant memory cost as a function of the depth of the network. In this work, we are not focused on optimization of training of neural networks, but rather propose a novel technique to train generative adversarial networks in a way in which the generator learns an ODE function which represents the dynamics of a biological system (Section 3).

ECG Classification Applying deep learning models to ECG classification has been gaining growing attention. The state-of-the-art method for ECG heartbeat-level classification (Kachuee, Fazeli, and Sarrafzadeh 2018) recently showed that superior results are reached by applying a ResNet model which classifies each heartbeat class separately. In this work, we focus on training these models with additional synthetic ECG heartbeats, generated from our proposed generative model (Section 5).

The ECG Simulator An ECG heartbeat follows a prototypical pattern of a P wave, followed by a QRS complex, and finally a T wave. To capture this pattern, (McSharry et al. 2003) proposed a model of the heart by a system of three ordinary differential equations. The resulting simulator is able to generate synthetic ECG signals with realistic PQRST morphology, as well as prescribed heart rate dynamics. The simulator is parameterized by specific heart rate statistics, such as the mean and standard deviation of the heart rate, as well as frequency-domain characteristics of the heart rate variability (Malik and Camm 1990). While the model is derived mainly from physical knowledge, it has limited expressivity. In our work, we wish to learn a more expressive dynamical system, which is both physical and data driven, using a generative deep-learning approach.

SimGans (Golany, Freedman, and Radinsky 2020) introduced a GAN-based setup enriched with additional knowledge from the ECG simulator. In a regular GAN setting, the generator learns to generate synthetic data, based on input noise. In their framework, a special loss term is added to the generator optimization. This loss term tries to minimize the distance between the generated heartbeat dynamics to the dynamics of the ECG Simulator. Combining the additional loss with the classical cross-entropy loss enables the generator to create synthetic ECG heartbeats with real morphology and characteristics which don't exist in the training set, while preserving the noise which defines the real data. They showed that utilizing the synthetically generated ECG heartbeats guided by the simulator significantly improves ECG heartbeat classification. However, their model is limited to generating heartbeats with morphology defined by a specific dynamical system, and limited by the capability of the ECG simulator. In this work, we focus on how to generate novel ODEs using GANs, as opposed to leveraging existing models. We hypothesize that this approach will learn more expressive physical simulators of the dynamic system.

3 ODE GANs

In this section, we introduce the ODE-GAN setting where the generator network is represented as an ordinary differential equation. Our goal is to learn to generate a dynamical system representable by an ODE (in our case, an ECG heartbeat). We wish to learn the following ODE-GAN model:

ODE-Generator Model: The ODE-Generator network, G, has the following ODE structure:

$$\frac{d\mathsf{x}}{dt} = G(\mathsf{x}, t, z; \theta_G) \tag{1}$$

where $t \in [0, 1]$ is a scalar denoting time; x is the scalar value of the signal at time t; z is a noise vector; and θ_G are the network weights. The generator's goal is to learn the dynamics of the signal $\mathsf{x}(t)$, expressed as the above ODE.

In order to solve the ODE, an initial condition must be specified: $\mathsf{x}(0) = x_0$. To solve the ODE in practice, numerical methods are used, in which the signal $\mathsf{x}(\cdot)$ is discretized. Specifically, we will designate the discretized sequence with subscripts as $x \equiv [x_0, \ldots, x_T]$, where $x_t = \mathsf{x}(t\Delta)$, and

$\Delta = 1/T$; note the change in font between continuous x and discrete x. To solve the ODE in (1) numerically, one can use a variety of standard techniques (Butcher and Goodwin 2008); in our experiments, we used the Runge-Kutta fourth-order method (Runge 1895). We designate a single time step of the solution generically as

$$x_{t+1} = \texttt{ODETimeStep}(x_t, \Delta, G(x_t, t\Delta, z; \theta_G)) \quad (2)$$

The entire sequence $x_0, \ldots, x_T$ can be constructed given the initial condition x_0 by running Equation (2) iteratively. We denote this solution for the entire sequence as

$$x \equiv [x_0, \ldots, x_T] = \texttt{Solve}(z, x_0; \theta_G) \quad (3)$$

where G's role is implicitly specified by its parameters θ_G.

Discriminator Model: As usual, the discriminator network D receives as input a sequence $x = [x_0, \ldots, x_T]$, and its goal is to identify whether the sequence was generated from the true dynamics or from the generated dynamics:

$$D(x; \theta_D) \quad (4)$$

ODE-Generator Training: As with ordinary GANs, the ODE-Generator's goal is to prevent the discriminator network from being able to distinguish between the generated time-series and the real time-series. Formally, the ODE-Generator aims to minimize the log-probability that the discriminator correctly assigns negative labels to the time-series produced by G. The cross-entropy loss of the generator may be written as:

$$L_G^{CE}(\theta_G) = -\mathbb{E}_{z, x_0 \sim \mathbf{z}, \mathbf{x_0}} \log D(\texttt{Solve}(z, x_0; \theta_G); \theta_D) \quad (5)$$

Discriminator Training: The objective of the discriminator is to maximize the log-probability of assigning correct labels to both real and generated samples. Formally, the discriminator cross-entropy loss is:

$$\begin{aligned} L_D(\theta_D) = &-\mathbb{E}_{x \sim \mathbf{x_{data}}} \log D(x; \theta_D) \\ &- \mathbb{E}_{z, x_0 \sim \mathbf{z}, \mathbf{x_0}} \log(1 - D(\texttt{Solve}(z, x_0; \theta_G); \theta_D)) \end{aligned} \quad (6)$$

4 Introducing Physical Considerations

Our goal is to find an expressive dynamical system for describing the electrical activity in the heart as measured by ECGs. We introduce the framework of ECG-ODE-GAN, an ODE-GAN based setup which learns ECG heartbeats dynamics. The key element which distinguishes the ECG-ODE-GAN framework from the ODE-GAN framework of Section 3 is the incoporation of *physical considerations*. More specifically, the ODE-GAN framework is purely data driven; by contrast, the ECG-ODE-GAN framework which we introduce in this section uses a combination of both data and physical considerations. This combination improves upon the original ODE-GAN framework, as we demonstrate empirically in Section 7.

The overall approach is as follows. In order to introduce physical considerations into the ODE-GAN framework of Section 3, we make one straightforward change: the noise vector z is replaced by a set of physically meaningful parameters. These parameters must in some way partially characterize dynamics of the system which produces the signal of interest. The exact way the physical parameters relate to the dynamics will vary between modelling tasks. Nevertheless, replacing the uninformative noise distribution of the ODE-GAN framework with a set of physically meaningful parameters, along with information about their corresponding distribution, is a very elegant way of combining both data and physical considerations.

We now show how to apply this framework to the problem of learning a dynamical system for ECG generation. In what follows, to emphasize the distinction between physically meaningful parameters and noise, we will denote the former by η and continue to denote the latter by z.

ECG Signals and Their Physical Parameters

An ECG signal taken from a patient in one lead is segmented into heartbeats (cardiac cycles), which are labelled h. Each such heartbeat may be written as a fixed length vector $h = [h_0, \ldots, h_T]$, which represents the time evolution of the voltage values of a single heartbeat. We set $T = 216$, where the signal domain represents a 600ms range sampled at 360 samples per second, in which the range is from 200ms before the R-peak to 400ms after the R-peak.

The physical parameters of the ECG dynamical system are taken to be the P-wave, QRS-complex, and T-wave (McSharry et al. 2003). Roughly speaking, these are points which describe various characteristic peaks on a given ECG signal. More specifically, we take the physical parameters to be both the voltage values and locations in time of each: $\eta = \{(x_\beta, t_\beta)\}_{\beta \in \mathcal{B}}$, where $\mathcal{B} = \{P, Q, R, S, T\}$.

The ECG-ODE-GAN Generator

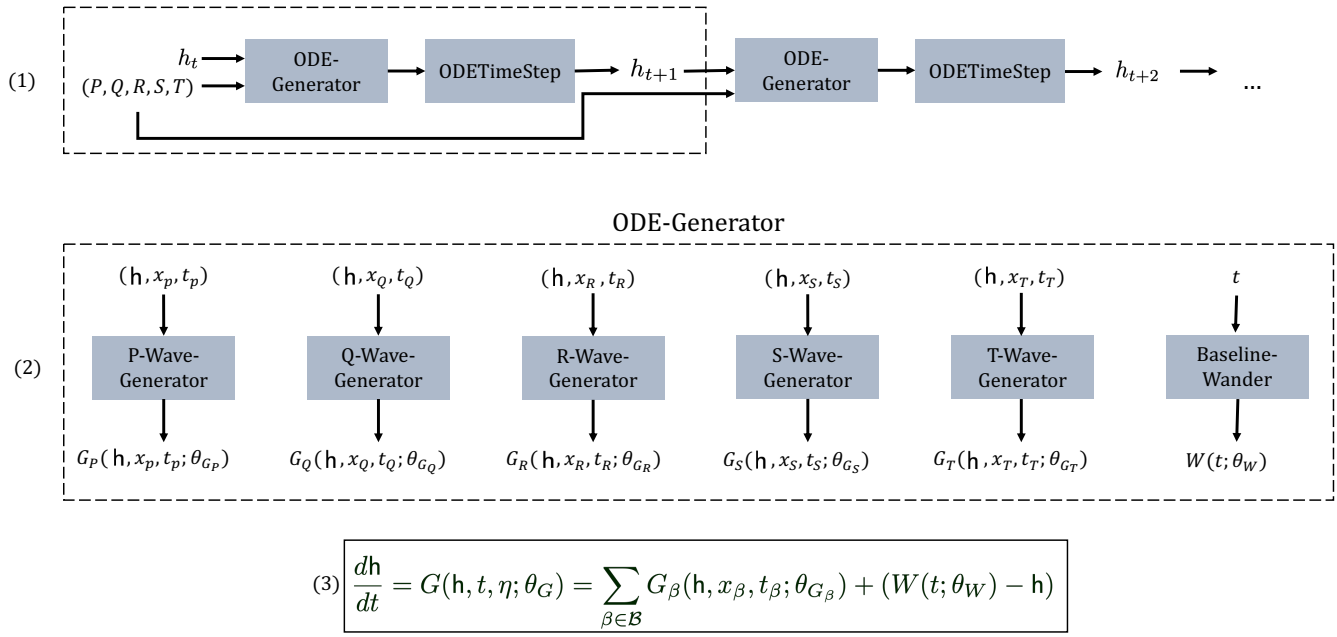
Figure 2 describes the ODE-Generator architecture. Our proposed ECG-ODE-GAN Generator architecture is composed of 6 sub-networks:

$$\begin{aligned} \frac{d\mathsf{h}}{dt} &= G(\mathsf{h}, t, \eta; \theta_G) \\ &= \sum_{\beta \in \mathcal{B}} G_\beta(\mathsf{h}, x_\beta, t_\beta; \theta_{G_\beta}) + (W(t; \theta_W) - \mathsf{h}) \end{aligned} \quad (7)$$

In addition to receiving h as an input, each of the first 5 sub-networks G_β also receive the voltage value and location of the corresponding peak, i.e. x_β, t_β. For each sub-network, the physical parameters and the current predicted voltage h are fed to 4 dense-layers. After each layer batch normalization (Ioffe and Szegedy 2015) is performed followed by a $\tanh$ activation function.

The sixth subnetwork W represents the "baseline wander" (Sörnmo and Laguna 2005), which is the noise that occurs due to patient movement, poor contact between electrode cables and ECG recording equipment, etc. For the baseline wander network, only the current time-step t is fed as input. The network consists of a single dense layer followed by a $\tanh$ activation function.

Learning the dynamical system, i.e. $G(\mathsf{h}, t, \eta; \theta_G)$, follows the procedure explained in Section 3. One key distinction is the expectation over the noise and initial conditions z, x_0 is replaced by an expectation over the physical parameters and initial conditions η, h_0. While the distribution over



(3) $$\boxed{\frac{d\mathrm{h}}{dt} = G(\mathrm{h}, t, \eta; \theta_G) = \sum_{\beta \in \mathcal{B}} G_\beta(\mathrm{h}, x_\beta, t_\beta; \theta_{G_\beta}) + (W(t; \theta_W) - \mathrm{h})}$$

Figure 2: ECG-ODE-GAN generator overview. (1) Synthetic ECG heartbeat generation process: The ODE-Generator starts by receiving as input an initial value h_0 and physical ECG parameters η. The output of the ODE-Generator is the derivative of the current value w.r.t to the current time-step. Therefore, a numeric ODE-solver can be applied for L time-steps. Each time-step produces the estimated value at the next time-step, given the value at the current time-step. (2) The ODE-Generator architecture consists of 6 sub-networks, where each of the first 5 sub-networks learns the dynamics of a specific wave within the heartbeat, and the sixth sub-network learns additional noise which occurs during measurements. (3) The final equation of the ODE-Generator which estimates the dynamics $d\mathrm{h}/dt$ of the ECG.

the noise is generally taken to be a standard distribution, like a Gaussian, the distribution over the physical parameters is more easily taken to be a discrete distribution over the physical parameters (and corresponding initial conditions) of a given fixed set of signals. It is then simple to sample from this distribution.

Finally, we note that there are multiple heartbeat classes. We learn a separate ECG-ODE-GAN model for each class.

5 Deep ECG Classification

To measure the quality of the ECG-ODE-GAN, we follow the practice of (Golany, Freedman, and Radinsky 2020). We use a trained ECG-ODE-GAN to generate synthetic heartbeats. The synthetic heartbeats are used to train a deep neural network that classifies each heartbeat according to its heartbeat class. For evaluation we used a 1D-Resnet network, which was found to have superior results on the ECG gold-standard dataset (Kachuee, Fazeli, and Sarrafzadeh 2018). The network architecture consists of a convolutional layer followed by five residual convolutional blocks. Each residual block contains two convolutional layers, two corresponding ReLU activations, a residual skip connection, and a pooling layer. The last residual block is followed by two fully-connected layers with 32 neurons each and a softmax layer to predict output class probabilities. Each convolutional layer is a one-dimensional convolution through time and has 32 kernels of size 5. The network is trained on the labeled training set of the dataset described in Section 6 and additional synthetic heartbeats coming from each of the trained generators (as described in Section 4, for each heartbeat class, we train a separate ECG-ODE-GAN model).

6 Experimental Evaluation

ECG Dataset

The data consists of ECG recordings taken from the MIT-BIH arrhythmia database (Moody, Mark, and Goldberger 2001), which is the most popular public dataset for discovering and clustering arrhythmias, and is considered the gold-standard evaluation data for ECG heartbeat classification tasks. The database contains 48 half-hour ECG records, obtained from patients studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Each record contains two 30-minute ECG lead signals digitized at 360 samples per second. The database contains annotations for both heartbeat class information and timing information verified by independent experts. The database consists of a total of 109,492 heartbeats, each of which has been labelled with one of the following four classes: Ventricular Ectopic Beat, shortened to VEB or V; Supraventricular Ectopic Beat, shortened to SVEB or S; Fusion Beat, shortened to F; Normal beat, shortened to N.

Experimental Methodology

The AAMI (Association for the Advancement of Medical Instrumentation) standard specifies a protocol for partitioning the MIT-BIH to train and test sets. We follow this partition as previous works do (De Chazal, O'Dwyer, and Reilly 2004; Golany, Freedman, and Radinsky 2020; Al Rahhal et al. 2016). This partitioning ensures that heartbeats from the same patient belong to the same partition. For evaluation we calculate a separate precision-recall curve for each type of heartbeat, also as recommended by the AAMI standard. That is, for each type of heartbeat, we measure how well the heartbeat is classified against all other type of heartbeats. We present results for the following different data synthesis regimes:

No ECG Generation Classification performance using a Resnet model which recently showed state-of-the-art results in the task of ECG heartbeat classification (Kachuee, Fazeli, and Sarrafzadeh 2018). The model is trained on the training set, without any additional synthetic ECG heartbeats.

ECG Generation using SimGAN We compare our model with the current state-of-the-art for the task of ECG heartbeat classification (Golany, Freedman, and Radinsky 2020).

ECG Generation using standard State-of-the-Art GAN frameworks We train standard GAN models, where the generator gets as input random noise from a Gaussian distribution, and outputs a full ECG heartbeat at once:

$$h = G(z; \theta_G) \tag{8}$$

where $z \in \mathbb{R}^{100}$ is white Gaussian noise. We evaluate two type of GANs:

1. *DCGAN* – the architectures of the generator and discriminator are as described by (Radford, Metz, and Chintala 2015), where each 2D-convolution layer is transformed to a 1D-convolution layer. The same model was proposed by (Golany et al. 2020). The loss functions of the generator and discriminator are the standard cross-entropy losses.
2. *WGAN* – Wasserstein-GAN (Arjovsky, Chintala, and Bottou 2017) minimizes the approximation of the Earth-Mover (EM) distance instead of the cross-entropy loss. Experiments have shown that the Wasserstein loss prevents model collapse and leads to more stable training overall. The architectures of the generator and discriminator remains the same as in the DCGAN.

ECG Generation using the ECG Simulator A purely physical model. We would like to test the contribution of learning a new dynamical model with an ODE-GAN, against the heuristic dynamical model proposed by (McSharry et al. 2003), see Section 2. We train the Resnet model on the training set with additional synthesized ECG heartbeats which come directly from this ECG Simulator.

ECG Generation using ODE-GANs We compare ECG-ODE-GAN with several variations of ODE-GANs. We would like to show that both the physics and the differential equations structure are important for a better model.

1. *Purely data-driven model*: $\frac{d\mathsf{h}}{dt} = G_\theta(\mathsf{h}, t)$. The generator attempts to learn the dynamics of the ECG heartbeat without the physical parameters of the heartbeat. We use the settings described in Section 3.
2. *Purely data-driven model with additional noise*: $\frac{d\mathsf{h}}{dt} = G_\theta(\mathsf{h}, t, z)$. We test if adding random noise to the input of an ODE-generator improves ECG heartbeats generation.
3. *Mixed noise-physical model*: $\frac{d\mathsf{h}}{dt} = G_\theta(\mathsf{h}, t, \eta, z)$. The ODE-Generator gets as input both the physical parameters and additional noise. By way of ablation study, we want to test the combination of additional noise with the physical parameters.

ECG Generation using ECG-ODE-GANs The main approach proposed in this paper. The ResNet model is trained on heartbeats from all patients from the training set with additional synthesized heartbeats from a ECG-ODE-GAN. We test the ECG-ODE-GAN with two types of architectures and optimizations:

1. The ODE-Generator is composed of 6 sub-networks as described in Section 4. Each of the first 5 sub-networks receives as input physical parameters of a specific wave within a heartbeat, in addition to the heartbeat voltage value at a specific time-step.
2. The ODE-Generator is composed of one network, which receives at once all of the physical parameters η, and the voltage value at the current time-step. In this setting, the physical parameters and the current voltage value are fed to 6 dense-layers. After each layer batch normalization is performed, followed by a $\tanh$ activation function.

The number of synthetically generated beats which are to be added to the training set is a parameter of the model, and depends on the number of samples from each class. For a heartbeat class with n samples in the base set, we experimented with the following values: $0.1n, 0.3n, 0.5n, 0.8n, n, 1.5n, 2n$.

7 Results

Comparison with the State-of-the-Art Making a fair comparison in the task of ECG heartbeat classification is somewhat challenging due to the fact that different papers use different settings. For example, previous papers presented results where heartbeats from the same subjects were shared between train and test sets (Jiang et al. 2006); classifiers which used more than one lead during training (Llamedo and Martínez 2012); classifiers which used RR intervals (Lin and Yang 2014); and semi-supervised heartbeat classifiers which had access to unlabeled data from the test (Golany and Radinsky 2019). We follow the settings described in the latest state-of-the-art methods presented by (Golany, Freedman, and Radinsky 2020) and (Kachuee, Fazeli, and Sarrafzadeh 2018), making use of only the raw ECG signals.

The results are presented in Table 1. As shown in the table, our model outperforms the state-of-the-art methods on the MIT-BIH dataset with respect to the precision-recall evaluation metrics. We observe that for all type of heartbeats, our method is better than adding synthetic heartbeats from a trained SimGAN, which learns to generate heartbeats based on the ECG simulator equations. We speculate that this is due to the fact that the data generated by SimGAN is constrained by the limited expressivity of the ECG simulator

equations. The ECG-ODE-GAN, however, is able to capture both physical aspects of an ECG heartbeat, as well as properties which come from the data.

	ECG-ODE-GAN		**SimGAN**		**Kachuee**	
Heartbeat class	Re	Pr	Re	Pr	Re	Pr
SVEB (S)	***0.40**	***0.83**	0.40	0.70	0.40	0.02
Fusion (F)	***0.20**	***0.45**	0.20	0.25	0.20	0.04
VEB (V)	***0.90**	***0.90**	0.90	0.86	0.80	0.88

Table 1: Comparison to the state-of-the-art methods on the MIT-BIH dataset. Best results are shown in bold and *.

Comparison with Standard GANs We test the contribution of ECG-ODE-GAN where the ODE-Generator learns the dynamics of the ECG signal, against standard GAN settings where the generator simply tries to learn the distribution of the ECG heartbeats. That is, in the ECG-ODE-GAN model the ODE-Generator generates synthetic ECG heartbeats by applying a numerical ODE solver on the learned dynamical system, while in the standard GAN setting, the generator learns to generate synthetic ECG heartbeats given random noise. In Table 2 we compare our method to two common standard GANs - DCGAN and WGAN. It can be seen that ECG-ODE-GAN outperforms the standard GANs for all type of heartbeats. We conclude that learning the dynamics of the ECG signal with a physically driven model is considerably more powerful then trying to simply learn the heartbeat distribution with standard GANs.

	ECG-ODE-GAN		**DCGAN**		**WGAN**	
Heartbeat class	Re	Pr	Re	Pr	Re	Pr
SVEB (S)	***0.40**	***0.83**	0.40	0.27	0.40	0.54
Fusion (F)	***0.20**	***0.45**	0.20	0.15	0.20	0.12
VEB (V)	***0.90**	***0.90**	0.88	0.80	0.88	0.75

Table 2: Comparison between our method to standard State-of-the-Art GANs. Best results are shown in bold and *.

Comparison with the ECG Simulator By way of an ablation study, we test whether the ECG-ODE-GAN model, which learns a new ECG dynamics based on physical and data parameters, brings more value then the ECG simulator. Table 3 shows significant points on the precision-recall curves for the three type of ECG heartbeat classes, SVEB, VEB and Fusion. We observe that adding synthetic heartbeats from our trained ECG-ODE-GAN, outperforms adding synthetic ECG heartbeats directly from the ECG simulator.

Contribution of Physical Considerations We would like to test the contribution of adding the physical ECG heartbeat parameters to the input of the ODE-GAN. Figure 3 presents precision recall curves for the task of classifying the MIT-BIH test-set over the heartbeat classes SVEB, Fusion and VEB respectively. We show curves for each of the four ODE-GAN types described in section 6, as well as the state-of-the-art deep learning method of (Kachuee, Fazeli, and Sarrafzadeh 2018) that does not add any synthetic ECGs.

	ECG-ODE-GAN		**ECG Simulator**	
Heartbeat class	Re	Pr	Re	Pr
SVEB (S)	***0.40**	***0.83**	0.40	0.56
Fusion (F)	***0.20**	***0.45**	0.20	0.05
VEB (V)	***0.90**	***0.90**	0.90	0.84

Table 3: Comparison to synthetic ECG heartbeats produced directly from the ECG Simulator. Best results are shown in bold and *.

First, we see that all models outperform (Kachuee, Fazeli, and Sarrafzadeh 2018), i.e., adding synthetic heartbeats from any type of ODE-GAN improves classification performance. We see significant improvements in the SVEB and Fusion heartbeats, which are quite rare and only comprise 3% and 2% of the dataset, respectively. We conjecture that when little data is available the practice of adding synthetic ECG generation to the training yields high gains.

Second, the results of our proposed ECG-ODE-GAN model (that receives only physical parameters) are demonstrably superior to those of any other ODE-GAN model. The performance of our proposed ECG-ODE-GAN model on Fusion heartbeats is significantly higher than the second best model, the purely data driven ODE-GAN. The performance on the VEB heartbeats are already quite high when evaluated by (Kachuee, Fazeli, and Sarrafzadeh 2018), yet our ECG-ODE-GAN achieves significantly higher performance.

Third, from our experiments we found that adding noise as input to the ODE-Generator yields unstable results, which are similar to the results of the purely data driven ODE-GAN. For VEB heartbeats the precision-recall curves reaches the same results of (Recall, Precision) = (0.85, 0.87). For Fusion heartbeats, the noiseless model reaches slightly better results of (Recall, Precision) = (0.2, 0.1) vs. (0.2, 0.07), and for SVEB heartbeats, the model with noise reaches (Recall, Precision) = (0.4, 0.5) vs. (0.4, 0.39) for the noiseless model. Overall, when adding noise as input to the ODE-Generator of an ECG-ODE-GAN (green curves in Figures 3), the results slightly decrease compared to the ECG-ODE-GAN which relies only on physical parameters (purple curves).

We conclude that adding physical parameters as input to the ODE-Generator is of high importance for the ODE-GAN to converge and generate realistic ECG heartbeats. Furthermore, while adding synthetic ECG heartbeats from any type of trained ODE-GAN improves ECG heartbeat classification, the ECG-ODE-GAN generation method yields the highest performance gains.

Impact of the ODE-Generator Architecture In our experiments, the ODE-Generator architecture and optimization follows the structure of the ECG Simulator. It is composed of 5 sub-networks where each one learns the dynamics of a specific wave from the physical parameters of the heartbeat, and another sub-network which learns measurement noise. Table 4 shows the impact of this structure on the ECG heartbeat classification performance. For all type

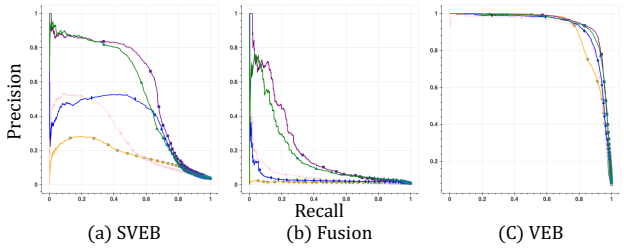


Figure 3: Precision - Recall curves of the 3 heartbeat classes evaluated on the test-set. Each subfigure shows curves for each of the four data synthesis regimes described in Section 6, as well as the state-of-the-art method of (Kachuee et al., 2018). ECG-ODE-GAN (purple), purely data-driven (pink), purely data-driven with noise (blue), mixed noise-physical (green),Kachuee et al., 2018 (orange)

of heartbeats, our design significantly improves the performance, compared to an ODE-Generator with an architecture composed of a single network which receives all physical parameters at once. An ODE-Generator composed of a single network doesn't improves results compared to SimGAN (Golany, Freedman, and Radinsky 2020). In Appendix A we show the losses of the ODE-Generators and discriminators during training. When training the ECG-ODE-GAN with an ODE-Generator composed of a single network, the discriminator easily separates between positive and negative heartbeats, while the generator loss grows. This emphasizes that constructing an architecture which learns to generate each wave of the heartbeat separately is of high importance.

	ECG-ODE-GAN 6-Sub-Networks		**ECG-ODE-GAN 1 Network**		**SimGAN**	
Heartbeat class	Re	Pr	Re	Pr	Re	Pr
SVEB (S)	***0.40**	***0.83**	0.40	0.33	0.40	0.70
Fusion (F)	***0.20**	***0.45**	0.20	0.06	0.20	0.25
VEB (V)	***0.90**	***0.90**	0.79	0.87	0.90	0.86

Table 4: Comparison of ECG-ODE-GAN with different ODE-Generator architectures. *6 Sub-Networks* refers to our suggested architecture (Section 4). *1 Network* refers to an ODE-Generator where the ODE-Generator architecture is composed of single network. Best results are in bold and *.

Qualitative Results Figure 4 presents qualitative examples for each type of heartbeat. The orange heartbeat in each graph is a real heartbeat, where the P, Q, R, S and T waves are denoted with markers. In the left column blue heartbeats are synthetic heartbeats coming from the ODE-Generator and are generated by feeding to the ODE-Generator the same physical parameters as the real heartbeats and the initial voltage value of the real heartbeats. On the right column the blue heartbeats are synthetic heartbeats coming from recent state-of-the-art SimGAN.

For all type of heartbeats, we see that the synthetic heartbeats generated from our proposed ODE-Generator (left side) contain realistic ECG morphology, and follow the same morphology as the real heartbeats; while the synthetic heartbeats coming from the SimGAN are missing the ECG morphology and have many more artifacts. For example, for a heartbeat from the Fusion class (top row), we see that the generated heartbeat coming from SimGAN only captures the R and T waves, while the locations of the P, Q and S waves are different from the real heartbeat. However, the Fusion heartbeat generated from our ODE-Generator follows a real ECG morphology with full P, Q, R, S and T waves.

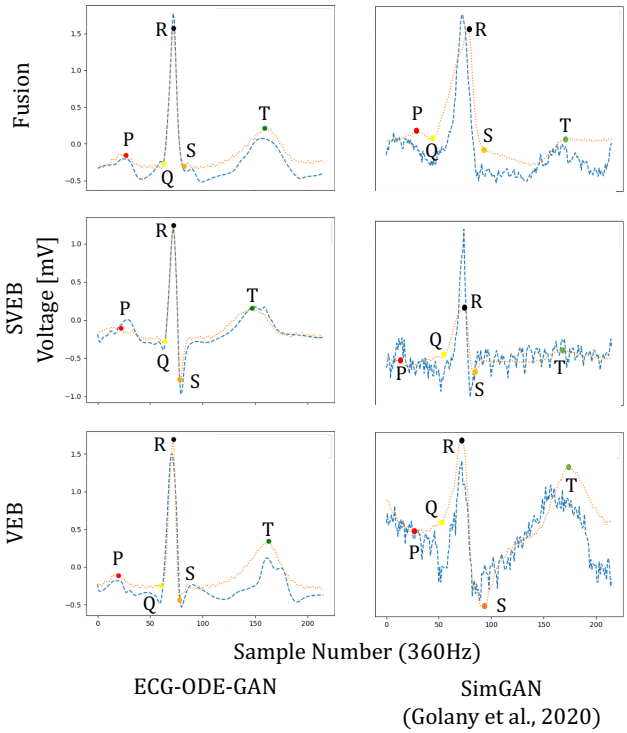


Figure 4: (Left column) Heartbeats produced by our ODE-Generator (blue) drawn together with real-heartbeats (orange) with the same physical-parameters as given to the ODE-Generator as input. (Right column) Heartbeats produced by SimGAN (Golany, Freedman, and Radinsky 2020).

8 Conclusions

We have presented a new technique for learning the dynamics of a mathematical process simulator represented by ODEs with deep generative adversarial networks. We have shown how an ODE model describing cardiac cycles can be learnt by a GAN-based generative model. Empirically, we have demonstrated that by using the synthetic cardiac cycles generated from such a model as training data, we can improve the classification accuracy of a standard ResNet ECG heartbeat classifier, attaining state-of-the-art results.

The approach presented here is not specific to cardiac cycles; it applies equally well to any system described mathematically by differential equations.

References

Al Rahhal, M. M.; Bazi, Y.; AlHichri, H.; Alajlan, N.; Melgani, F.; and Yager, R. R. 2016. Deep learning approach for active classification of electrocardiogram signals. *Information Sciences* 345: 340–354.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* .

Butcher, J. C.; and Goodwin, N. 2008. *Numerical methods for ordinary differential equations*, volume 2. Wiley Online Library.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. In *Advances in neural information processing systems*, 6571–6583.

De Chazal, P.; O'Dwyer, M.; and Reilly, R. B. 2004. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE transactions on biomedical engineering* 51(7): 1196–1206.

Golany, T.; Freedman, D.; and Radinsky, K. 2020. SimGANs: Simulator-Based Generative Adversarial Networks for ECG Synthesis to Improve Deep ECG Classification. *arXiv preprint arXiv:2006.15353* .

Golany, T.; Lavee, G.; Yarden, S. T.; and Radinsky, K. 2020. Improving ECG Classification using Generative Adversarial Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Golany, T.; and Radinsky, K. 2019. PGANs: Personalized Generative Adversarial Networks for ECG Synthesis to Improve Patient-Specific Deep ECG Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 557–564.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Jiang, X.; Zhang, L.; Zhao, Q.; and Albayrak, S. 2006. ECG arrhythmias recognition system based on independent component analysis feature extraction. In *TENCON 2006-2006 IEEE Region 10 Conference*, 1–4. IEEE.

Kachuee, M.; Fazeli, S.; and Sarrafzadeh, M. 2018. Ecg heartbeat classification: A deep transferable representation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, 443–444. IEEE.

Lin, C.-C.; and Yang, C.-M. 2014. Heartbeat classification using normalized RR intervals and morphological features. *Mathematical Problems in Engineering* 2014.

Llamedo, M.; and Martínez, J. P. 2012. An automatic patient-adapted ECG heartbeat classifier allowing expert assistance. *IEEE Transactions on Biomedical Engineering* 59(8): 2312–2320.

Malik, M.; and Camm, A. J. 1990. Heart rate variability. *Clinical cardiology* 13(8): 570–576.

McSharry, P. E.; Clifford, G. D.; Tarassenko, L.; and Smith, L. A. 2003. A dynamical model for generating synthetic electrocardiogram signals. *IEEE transactions on biomedical engineering* 50(3): 289–294.

Moody, G. B.; Mark, R. G.; and Goldberger, A. L. 2001. PhysioNet: a web-based resource for the study of physiologic signals. *IEEE Engineering in Medicine and Biology Magazine* 20(3): 70–75.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* .

Runge, C. 1895. Über die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen* 46(2): 167–178.

Sörnmo, L.; and Laguna, P. 2005. *Bioelectrical signal processing in cardiac and neurological applications*, volume 8. Academic Press.