




Article

# Conditional Synthesis of Blood Glucose Profiles for T1D Patients Using Deep Generative Models

Omer Mujahid <sup>1</sup>, Ivan Contreras <sup>1,†</sup> , Aleix Beneyto <sup>1</sup> , Ignacio Conget <sup>2,3,4</sup>, Marga Giménez <sup>2,3,4</sup> and Josep Vehi <sup>1,2,\*</sup> 

- <sup>1</sup> Modeling, Identification and Control Laboratory, Institut d'Informàtica i Aplicacions, Universitat de Girona, 17003 Girona, Spain  
<sup>2</sup> Centro de Investigación Biomédica en Red de Diabetes y Enfermedades Metabólicas Asociadas (CIBERDEM), 17003 Girona, Spain  
<sup>3</sup> Institut d'Investigacions Biomèdiques August Pi i Sunyer (IDIBAPS), 08023 Barcelona, Spain  
<sup>4</sup> Diabetes Unit, Endocrinology and Nutrition Department, Hospital Clínic, 08036 Barcelona, Spain  
\* Correspondence: josep.vehi@udg.edu; Tel.: +34-620131826  
† Professor Serra Húnter.

**Abstract:** Mathematical modeling of the glucose–insulin system forms the core of simulators in the field of glucose metabolism. The complexity of human biological systems makes it a challenging task for the physiological models to encompass the entirety of such systems. Even though modern diabetes simulators perform a respectable task of simulating the glucose–insulin action, they are unable to estimate various phenomena affecting the glycemic profile of an individual such as glycemic disturbances and patient behavior. This research work presents a potential solution to this problem by proposing a method for the generation of blood glucose values conditioned on plasma insulin approximation of type 1 diabetes patients using a pixel-to-pixel generative adversarial network. Two type-1 diabetes cohorts comprising 29 and 6 patients, respectively, are used to train the generative model. This study shows that the generated blood glucose values are statistically similar to the real blood glucose values, mimicking the time-in-range results for each of the standard blood glucose ranges in type 1 diabetes management and obtaining similar means and variability outcomes. Furthermore, the causal relationship between the plasma insulin values and the generated blood glucose conforms to the same relationship observed in real patients. These results herald the aptness of deep generative models for the generation of virtual patients with diabetes.

**Keywords:** deep generative models; conditional data synthesis; type 1 diabetes simulators; blood glucose data generation

**MSC:** 68T07



**Citation:** Mujahid, O.; Contreras, I.; Beneyto, A.; Conget, I.; Giménez, M.; Vehi, J. Conditional Synthesis of Blood Glucose Profiles for T1D Patients Using Deep Generative Models. *Mathematics* **2022**, *10*, 3741. <https://doi.org/math10203741>

Academic Editors: Costin Badica, Nick Bassiliades, Kalliopi Kravari and Theodoros Kosmanis

Received: 15 September 2022

Accepted: 8 October 2022

Published: 12 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

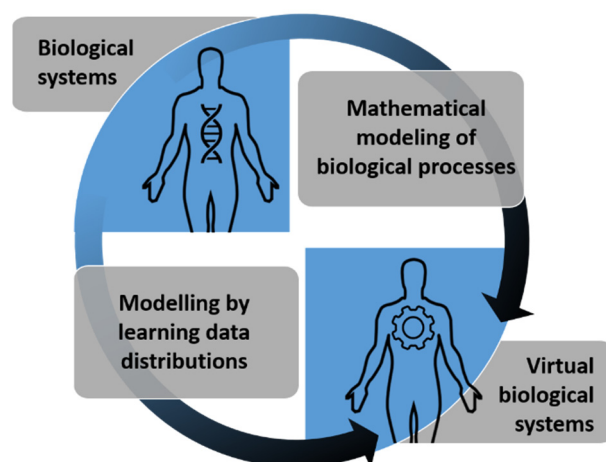
Biomedical simulation is at the heart of modern medical research. Any biomedical phenomena that can be simulated has the liberty to be subjected to the hit and trial approach of the scientific method, which is impossible to apply on living beings [1]. Similarly, type 1 diabetes (T1D) simulators have been the reason for many groundbreaking discoveries in diabetes research [2,3]. However, the conventional biomedical simulators are based on physiological mathematical models that do not cover the entirety of human physiology. Hence, the scenarios simulated by these simulators are far from perfect [4]. Keeping in view the challenges in the path of a model that should describe the complete physiology of the human body, the realization of such a digital twin seems very unlikely in the near future [5]. Even though the advancement in computing technology and power, the availability of large data storage devices, low-cost smart devices, and the ease of data acquisition provides us

with a glimmer of hope, there is still a long way to go in the path of achieving a perfect physiological description of the human body [6].

Current diabetes simulators suffer from the same fate. They try to mimic the mechanism of a human pancreas using mathematical modeling and control algorithms [7]. Most of the current diabetes simulators employ a mathematical model of the glucose–insulin system. A typical glucose–insulin model contains sub compartments such as the glucose model, the insulin delivery model, the gastro-intestinal tract, muscle and adipose tissue, and the liver model. However, it is important to note that in a real human being the BG profile is affected by many more biological systems and processes than just these. Sometimes, the real biological systems are so complex that it is impossible to approximate them mathematically. Additionally, there are certain hidden biological phenomena occurring that affect the BG profile of a human being and hence cannot be approximated at all. A generalization over all those systems that are not modeled or cannot be modelled in a T1D simulator leads to an intrinsic error that is evident in the simulated scenario [8,9].

The majority of the existing diabetes simulators also lack support for scenarios such as exercise (most types of it), illness, menstrual cycles, sleep disorders, depression, etc. Having said that, there have been efforts from scientists/engineers to incorporate disturbances such as physical activities and patient behaviors in the existing simulators [10,11]. It is understood that patient behavior affects the glycemic profile of a diabetic patient immensely. Lifestyle choices and routine habits may cause the glycemic profile of a diabetic to go from good to worse and vice versa [12]. Eating habits, alcohol consumption, missed meals, and non-compliance with the doctor’s instructions may prove decisive in achieving control of diabetes for a patient [13]. An ideal diabetes simulator should provide support for the simulation of such scenarios in order to capture the complete essence of the effects of diabetes on a person.

Regardless of the modeling issues, the conventional T1D simulators are still an effective way to simulate diabetes scenarios for *in silico* experiments. However, there is great room for improvement in these simulators and, consequently, the possibility of getting closer to a real patient with T1D. An ideal biomedical simulation task demands the simulator produce phenomena that fulfills certain performance criteria just like the real human body would. In addition, realistic simulators must have mechanisms that include a certain level of randomness to mimic the biological stochasticity of real-life phenomena. Though it is understood that the simulated scenarios of a T1D simulator must be reusable, understandable, and reproducible, a T1D simulator in no way should be completely deterministic. All this points to the need of such a T1D simulator that encompasses the true and whole physiology of the human glucose–insulin model, is capable of simulating customized scenarios, and produces unique outcomes. Figure 1 shows two ways of approximating the biological systems/processes of the human body in order to be incorporated into a simulator. It can be seen from the figure that the framing of a biological system could be done either by physiological modeling or by learning the distribution of data these biological systems produce. It could be understood intuitively that the compound effect of all the disturbances on the glucose–insulin system of a person is reflected in the data acquired from the system. When investigated for its probability distribution, this data contains the collective features of all the processes and disturbances that influence the glycemic profile of a person. The idea is to learn this underlying probability distribution of the data and then generate novel samples of data using the learned distribution.



**Figure 1.** Two approaches to synthetic human biological systems.

Data-driven generic function approximators such as deep neural networks (DNNs) have the ability to learn from data the underlying behaviors without deriving from first principle [14]. DNNs have been extensively used for the prediction, detection, and classification of adverse glycemic events in T1D management [15,16]. With sufficient good quality data, DNNs have the ability to perform better than conventional physiological models [17,18]. Models that can synthesize the observed data are known as generative models. Generative models learn the joint probability distribution of data, unlike discriminative models that learn the conditional probability distribution of data instead. When DNNs are used for this purpose, they are termed as deep generative models (DGMs). DGMs have the ability to learn from a complex distribution and then generate new samples from the learned distribution. Such deep learning models could be well suited for biomedical data simulation tasks because, firstly, they can learn almost all the characteristics of a cohort of patients given that there is sufficient data, and secondly, they can produce unique samples from the learned distribution that are statistically similar to original data [19]. One of the most successful techniques capable of generating synthetic data based on a training in real scenarios is the generative adversarial network (GAN), which in recent years have demonstrated its value in biomedical data generation [20,21]. A particular type of GAN is known as conditional GAN (CGAN). CGAN differs in its functionality from a common GAN in such a way that an input label can control the generated data. CGANs have been used extensively in image translation, i.e., converting one type of image in to another type of image. It is equivalent to saying that one type of image is used as an input label to produce the targeted output. The image-to-image translation task is often referred to as pixel-to-pixel translation and the CGAN used for it is known as a pixel-to-pixel (Pix2Pix) GAN [22]. This paper proposes a strategy for the generation of realistic BG profiles of T1D patients conditioned on input insulin values using a Pix2Pix GAN. Furthermore, this work also discusses the possibility of a DGM that may be trained for the generation of T1D scenarios and can emulate a real T1D patient. The reason GAN is preferred over other types of DGMs in the proposed work is the fact that it is explicitly set up to optimize the generative task. Other DGMs, such as variational autoencoders, flow-based models, and diffusion models, are all set up to model the latent variable and are avoided here because of the problems related to latent variable approximation.

The major contributions of this work are as follows:

- A GAN has been used for the conditional generation of realistic BG values for the first time.
- The rationale for the development of a GAN-based diabetes patient simulator is given.

The rest of the paper is structured as follows: Section 2 explains deep generative models in detail along with its various types. Section 3 elaborates the methodology of the proposed work, with Section 4 dwelling in on the implementation and results of the

proposed research work. Discussion about the proposed methodology, implementation, and results is presented in Section 5. Section 6 concludes the paper with various suggestions for future work.

## 2. Deep Generative Models

Deep generative models have been used extensively to generate synthetic data for numerous sorts of applications over the past few years [23,24]. Though the most famous of these applications are the ones that involve facial image generation [25] and are under scrutiny for various reasons [26,27], the generative ability of these models could be harnessed positively in a variety of other ways. The DeepMind's rain predictor is one such example out of the many practical applications of deep generative models [28]. The aim of a DGM is to approximate a complex and high dimensional probability distribution using neural networks [29]. It does so by training a generative neural network in such a way that the obtained generator model learns to transform a tractable probability distribution  $Z$  in to a complex probability distribution  $X$ . Since DGM is an ill-posed problem, it is almost impossible to uniquely identify the probability distribution of a finite set of samples and hence the generation accuracy of a DGM is highly dependent on the hyper-parameters involved in the design of the model. There are several types of DGMs being used by researchers. None could be said to have any advantage or disadvantage over the other but that each one performs better than others under certain defined circumstances. The most common types of DGMs are the generative adversarial networks (GANs), the variational auto-encoders, the Bayesian networks, and the normalizing flows. Since our system uses a variant of GAN, our main focus of explanation is this methodology.

### 2.1. Generative Adversarial Network

GANs are a type of DGM that use two competing models in a zero-sum game, trying to optimize the loss function [30]. These models are a discriminator model ( $D$ ) and a generator model ( $G$ ). The function of  $D$  is to tell real data from the data generated by  $G$ , while  $G$  keeps on improving its generated output in order to fool  $D$ . Both  $D$  and  $G$  are DNNs trained to outperform each other.  $D$  is trained on real data and hence can classify real data from fake data.  $G$ , on the other hand, is trained using the feedback from  $D$ . The training of  $D$  is an act of minimizing a negative log likelihood loss whereas  $G$  uses two types of losses, the adversarial loss from  $D$  and the L1 loss from the similarity between the generated and real data. Once  $G$  learns the probability distribution of the real data set well enough, it can create infinite samples that are statistically similar to the real data and fool the discriminator.

### 2.2. Conditional Generative Adversarial Networks

CGAN is a variant of GAN that takes labeled data to train the discriminator instead of solo data. Where in a simple GAN the generation of data is not in our control and  $G$  generates new samples from the latent space  $Z$  that are unknown to us, in a CGAN the generation of new samples can be controlled with the help of data labels. Data samples along with their labels are fed to  $D$  for training purpose. Data labels are also used to condition  $Z$  in order to generate outputs that are similar to the data associated with the label. CGAN has been used effectively in text-to-image GAN architectures [31]. These models generate images according to the textual description provided by the user. The labels are provided as an embedded layer to the neural network during training. While generating new samples, the label information is added to  $Z$  so that the output is dependent on the label.

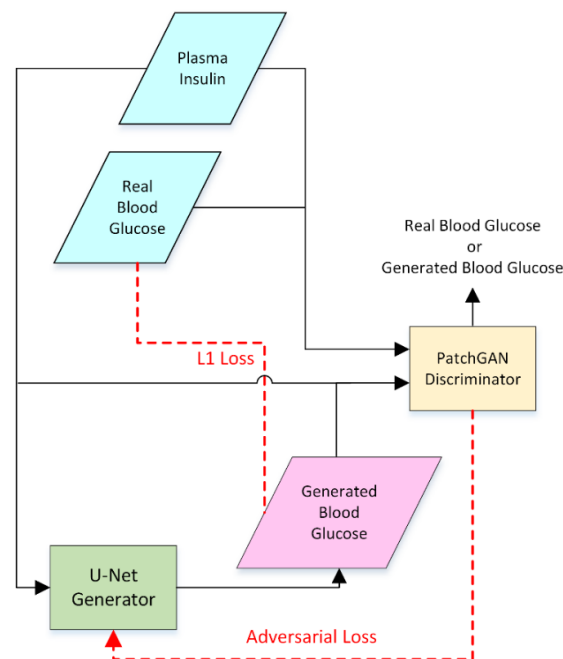
The operation of a CGAN may serve the purpose of a diabetes virtual patient simulator aptly. First, it can draw random samples from a random latent space and convert them into realistic glucose values. Secondly, it can also condition the generation of these glucose values on any of the other data such as insulin, carbohydrates, physical activity, etc. The generation of BG values that are conditioned on input insulin values mean that the

generated values are insulin dependent and will follow the underlying distribution of the insulin profile provided to the model as input.

### 2.3. Image Translation Using Conditional Generative Adversarial Networks

A variant of CGAN, known as the Pix2Pix architecture, is implemented in this research work. Part of the reason why we opted for a Pix2Pix architecture was to capture not only the data distribution but also its internal representations in a structured way from the portrayal of the BG and insulin time-series in image form. The Pix2Pix GAN performs the task of translating one type of image into another type of image. The generator in a Pix2Pix GAN learns to map an input image to an output image. To carry out this task, the input image is taken as a label to the target image. Pairs of input image and target image are used to train the discriminator. Additionally, the generator latent space is conditioned on the input label image as well in order to generate the images of our choice. Once trained,  $G$  is used to generate new images that are similar to real images. Each time the trained  $G$  is fed with a new input/label image, it will produce an output/target from the latent space. Hence, it is synonymous to say that unique target images will be generated using unique inputs. Fundamentally, the mechanism of generation in the proposed methodology is a translation task from one type of data to another type of data. In our model, having the generation of BG values respond to the input insulin labels gives us the liberty to generate BG scenarios of our choice. Moreover, along with the quantity of input insulin, the generated BG values are also affected by how the insulin samples interact with each other, hence preserving the temporal characteristics of the data.

For synthetic BG value generation, the insulin data are used as input, whereas the BG values are used as output data. The numerical data is first converted into images to prepare it for training. The conversion of data from numerical form to a graphical form is carried out using the HSV color map. After the formation of the image data set, training of the model is performed. This can be observed in Figure 2. Once, the generator learns the underlying probability distribution of the real data set, it can produce unique BG images when provided with unique insulin images.



**Figure 2.** Pix2Pix generative adversarial network architecture in training configuration for translation of plasma insulin images into blood glucose images.

### 3. Methodology

The proposed system uses a Pix2Pix GAN together with a series of various other techniques to achieve the task of virtual patient generation. The proposed system consists of four main parts that are also its main operations. In the order of operation, these parts are:

- Numerical data to graphical data conversion;
- Pix2Pix GAN Training;
- Data generation using a trained Pix2Pix generator;
- Graphical data to numerical data conversion.

#### 3.1. Experimental Data Sets

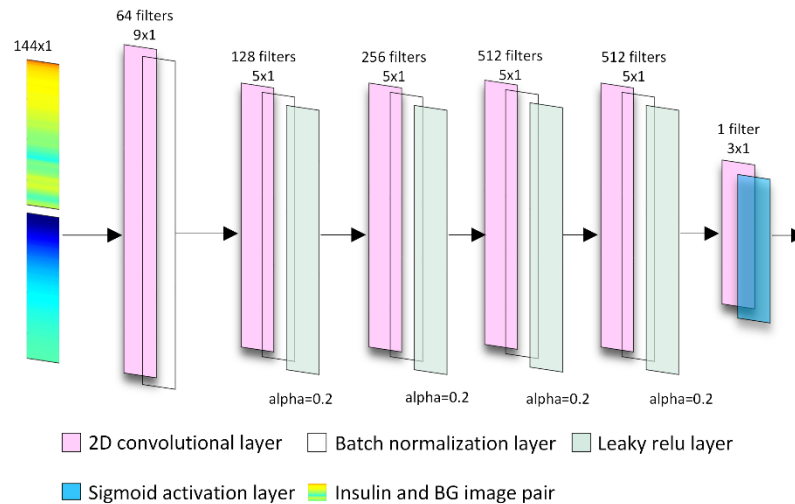
The proposed model is put to test in order to generate data that are statistically similar to two distinct data sets from separate studies. The first data set contains 29 patients on insulin pump therapy provided by the Hospital Clínic de Barcelona [32]. All the patients included in this data set wore Medtronic's MiniMed 640G insulin pumps. This data set contains a cumulative sum of 1169 days of insulin and BG data. By insulin data, here, we mean the plasma insulin approximation obtained using the Hovorka plasma insulin model of the total insulin injected to the patient [33]. The second data set used to train the Pix2Pix GAN is the Ohio T1DM data set [34]. We used data from the first 6 patients of this data set that consists of 264 days of BG and plasma insulin data. These Ohio T1DM data set patients were on insulin pump therapy as well and wore Medtronic 530G or 630G insulin pumps.

#### 3.2. Model Architecture

A Pix2Pix GAN is implemented using the Keras deep-learning framework in Python. As discussed already, a Pix2Pix GAN is a type of CGAN that conditions the generation of target images on some sort of label images. In our case, the generated/target images were that of BG and the input labels were insulin. It helps understanding the phenomena better if the insulin images are imagined as inputs while the glucose images as output. The implementation of Pix2Pix GAN involves expert designing of the generator and discriminator model and the strict optimization of the system. Deciding the specifications of  $G$  and  $D$  is the most important step in moving towards the implementation of a GAN. We have used the same architecture as presented by [22] with some modifications to the model layers and hyper-parameters. The  $D$  model is a DNN that performs the binary classification task of classifying images as real or fake. The images are provided to  $D$  as input and output pair as shown in Figure 2. The  $D$  model receives an insulin-BG image pair and classify them as real or fake. However, the  $D$  model used in the proposed model is a special type of model known as the patchGAN discriminator. Figure 3 shows the formation of the  $D$  model. The patchGAN  $D$  works on the concept of the effective receptive field of the model. It essentially means that the discriminator neural network only classifies a portion of the input image and do not take the entire image for the classification purpose. In the end, the outcomes of all such decisions are convoluted to obtain the one combined output. The benefit of such an approach is that different sizes of images could be classified with the help of such a model. Moreover, in our application, the patchGAN discriminator learns about the relationship between a particular segment of insulin image and a particular segment of BG image. This in effect means that our model does not only learn the general relationship between the 6 h chunks but also approximates the relationships between the samples falling inside these 6 h chunks.

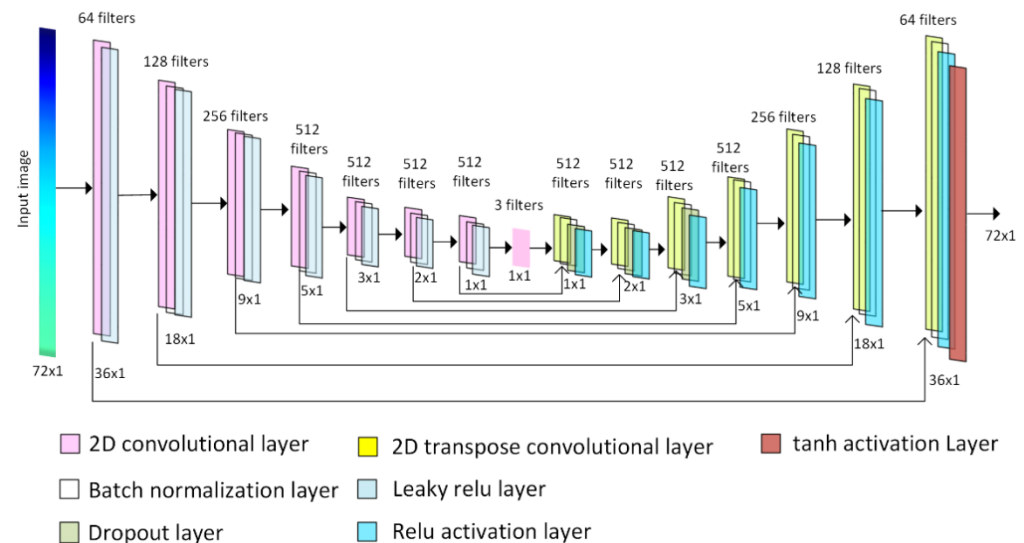
Since for our model the input is a  $72 \times 1$  dimension image, the kernel (filter) size is also kept as one-dimensional (1D). Here, it is important to consider that the convolutional layer used is still a two-dimensional (2D) layer and not 1D. The concept of padding has been employed in order to keep the spatial dimensions of the layer's outputs same as the inputs. Leaky rectified linear unit (Leaky ReLU) is used as the activation function at each layer. A total of six 2D convolutional layers are used in  $D$ . The training pair of plasma insulin image and BG image are concatenated and labeled as 1 for the real image pair before feeding

them to the Pix2Pix model for training. For the generated images, the plasma insulin and generated BG images are concatenated and labeled as 0. Concatenation is performed in order to have both types of images in the batch when the model weights are updated and the mapping between the images is learned.



**Figure 3.** The patchGAN discriminator model that performs classification between real and generated BG images.

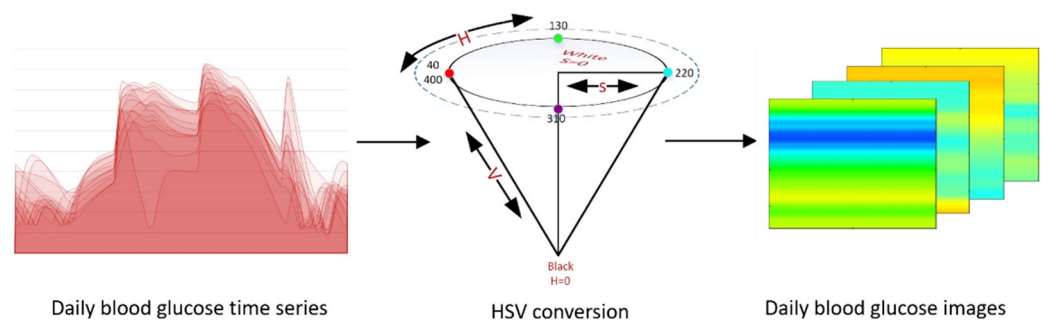
On the other hand, the G model, as shown in Figure 4, is comparatively more complex. This model is based on an encoder–decoder architecture which consists of two blocks of CNNs that are standardized for the purpose of replication: the encoder block and the decoder block. Both of these blocks are connected with the help of a bottleneck convolutional layer. Moreover, a slight modification is made to the conventional design of the encoder–decoder model by forming connections between layers of same size in both of these blocks [35]. This modification is usually referred to as the U-net configuration and is the reason why the generator in our designed Pix2Pix model is referred to as a U-Net generator. This design has been proven to facilitate the learning of minute details in an image [35] and can support the algorithm in learning transitions between colors for time-dependent characteristics of the insulin and BG time-series.



**Figure 4.** The U-Net generator model used to generate novel plausible BG images.

### 3.3. Translation between Numerical and Image Information

Since the GAN used in this work is a convolutional neural network-based image to image translator, the data used to train this model are image data. For this particular purpose, numerical data from the actual diabetes patient cohort are first converted into images. Conversion to image data is carried out through the HSV color map as shown in Figure 5. Six hours of insulin/BG data are converted into one HSV image using MATLAB. A duration of six hours is selected in order to completely approximate the effect of insulin and meals in the output BG profile. Each pixel in the HSV image represents the magnitude of the insulin/BG value in the actual 1D array. Six hours of data mean that there are 72 pixels in the obtained image since the sample time in the actual time-series is 5 min. The proposed system uses plasma insulin and glucose data for training and hence, only these two sets of data are converted into images.



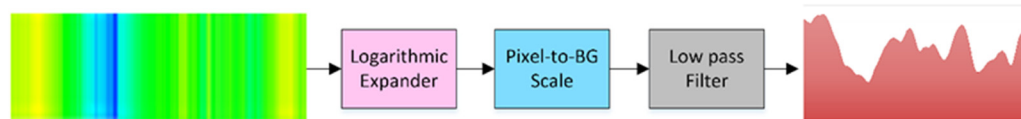
**Figure 5.** Numerical values to image translation using HSV color map: *H* stands for hue, *S* stands for saturation, *V* stands for value.

The HSV color map is chosen for image generation because of its simpler parametric composition. The HSV color scale is inspired by how the human eye perceives colors [36]. We see the same colors as those depicted by an RGB scale but do not comprehend them as a mixture of red, blue, and green. Instead, the human eye comprehends a color in terms of its saturation, tone, and brightness/darkness levels. The HSV color scale represents these parameters as the hue, the saturation, and the value (of brightness). The hue of the color is the color itself. The saturation of the color defines how saturated the quantity of color is, whereas the value of a color describes the brightness/darkness of a color.

It is important to note that even though its three channels, i.e., *H*, *S*, and *V*, represent the definition of a color on an HSV scale, it is still possible to depict any color correctly by using just the *H* parameter while keeping the *S* and *V* parameters constant (greater than 0). We have leveraged this property of the HSV scale by using just the *H* parameter for color portrayal while keeping the *S* and *V* parameters at a constant value of 1. This has made the translation from time-series data to color images a 1D-to-1D data translation in essence. Figure 5 shows an HSV cone. The *H* parameter represents the circumference of the circular plane in the cone. Each of its 360 degrees represents one color. The *S* parameter is represented by the radius of this plane, whereas the *V* parameter lies on the plane that connects the flat circular plane to the apex. To increase the saturation of a particular hue/color, we move away from the axis of the cone on the *S* parameter. On the other hand, increasing the *V* parameter takes us towards the lighter colors. Since, each color in the HSV color map can be represented by a single numerical value, i.e., the *H* parameter, it is easier to map an insulin/BG value to a pixel in the obtained image. The *H* parameter encodes each color as one of the 360 degrees of a circular plan. Since each color in the pallet is represented by one degree, there are a total of 360 colors on the *H* scale. The numerical time-series is transformed into its graphical representation by choosing one of these 360 colors for one magnitude value. Since for both insulin and BG time-series the possible values do not exceed the range of 360, transformation into images is a straightforward affair.



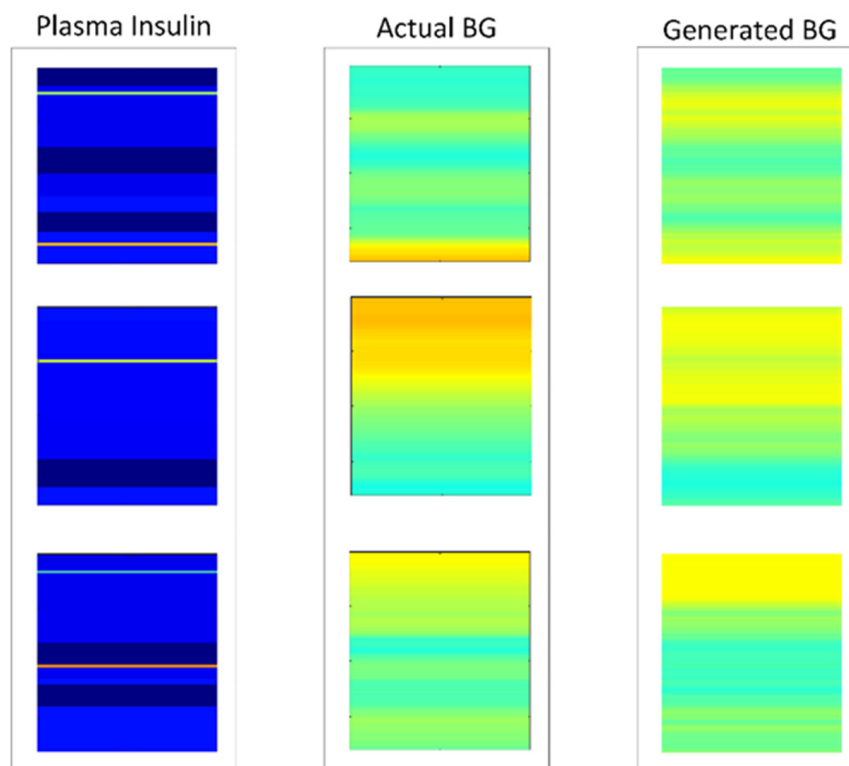
Furthermore, a translation mechanism is employed in order to translate the images back into numerical values. Since the output of the trained model is in image form, this translation mechanism is used to obtain the final BG time-series. As shown in Figure 6, while translating the generated images into BG values, the images are first given as an input to a  $\mu$ -law expander in order to increase the dynamic range of colors. There are several reasons for the low dynamic range of colors in the generated images. One reason is the bottleneck layer in the U-Net generator model where significant image details maybe lost. Decrease in the dynamic range might be considered somewhat similar to a logarithmic compression process where the amplitudes are brought more or less to the same level and the dynamic range is compressed. The expander reverses this process and increases the difference between the smallest and largest values of the signal. After passing through a  $\mu$ -law expander, the images are translated into BG values by using a pixel-to-BG scale. According to the pixel-to-BG scale, one pixel value is translated to one BG value. Since the HSV color map has 360 degrees representing 360 colors and the BG scale of 40–400 mg/dL too has 360 BG values, each color value can be mapped to each BG value. After the pixels are translated in to BG values, the resultant time-series is passed through a low-pass filter for further smoothing. A BG time-series is obtained at the output of the low-pass filter. Figure 6 demonstrates the conversion of generated BG images into numerical values using the translation mechanism discussed above.



**Figure 6.** Image to BG translation mechanism depicted with example: 18 h of generated image data is translated into blood glucose time-series.

### 3.4. Model Training and Data Generation

The proposed Pix2Pix generator is trained using an entire batch of paired images, i.e., the insulin and BG pairs. Each input insulin image corresponds to one target BG image. The total number of steps required to complete the training depends on the number of images in the data set multiplied by the number of epochs. Since there are 29 patients in the first data set, the Pix2Pix GAN was trained 29 times by excluding data from one patient during each training instance. This way, a total of 29 models were obtained. Similarly, for the Ohio data set, the model was trained 6 times for 6 patients in order to obtain 6 separate models. The training task of the proposed GAN architecture involves optimizing for two types of loss functions, the discriminator loss and the generator loss. The model weights for both the discriminator and generator model are updated after training on each individual image. The model is trained for 50 epochs because an empirical testing phase has pointed out this value as a common framework where the models achieved the convergence. The trained models are then used to generate data for each patient individually. The training process could be understood by looking at Figure 2. Once the generator is able to generate fake images that are real enough to fool the discriminator completely, the generator weights stop updating because the discriminator adversarial loss becomes zero. For an optimally trained GAN model the discriminator must demonstrate theoretically equal loss values for both real data and generated data. This means that the discriminator is unable to distinguish between real and fake images. In other words, the generator is producing images that are plausible enough to be considered real. This is when a generator model is considered to be fully trained and its weights should not be updated further. A fully trained generator model is then used to generate new fake images that are considerably similar to the real images. Figure 7 shows three sample images that have been generated by our generator along with the corresponding real BG data. Each generated image represents 6 h of BG data.



**Figure 7.** Generated images using a Pix2Pix GAN trained on blood glucose and plasma insulin data.

At least 4 weeks of data is generated for each patient in order to achieve the optimal amount of data required for reporting the glycemic performance parameters such as times in ranges and coefficient of variation (CV) etc. [37]. The generated BG data is then analyzed for various clinical parameters and times in standardized ranges of BG levels. These ranges are classified into three groups, namely the time below range (TBR), the time in range (TIR), and the time above range (TAR). Other metrics such as mean, CV, and maximum and minimum values of BG are also calculated for each patient. Means of all these parameters for the generated cohort are then compared against the means of the actual cohort. Moreover, the  $p$ -values from the Wilcoxon rank sum test are computed for each of these metrics of the generated data set against the metrics of the real data set. The 25% and 75% quartiles are computed for each metric as well and given as such.

#### 4. Results

Tables 1 and 2 contain all the results for both data sets. Table 1 shows the times in standardized ranges of BG levels, mean, CV, minimum and maximum values, and their corresponding  $p$ -values for the 29 patients real and generated cohorts. It could be clearly understood from the table that  $p$ -values depict strong statistical similarity between the two cohorts. In other words, the generated cohort and the real cohort display statistical significance since all the  $p$ -values qualify the 0.05 threshold of hypothesis confirmation. Similarly, Table 2 shows the results for the Ohio T1D data set. It could be observed for this data set, too, that the  $p$ -values for all the parameters qualify the null hypothesis of the two-sample test except for the  $p$ -value of the maximum value.

The inter quartile ranges (IQ) for the Ohio data set hints at space for improvement. The reason for this is understood to be the lesser quantity of data. Just like it has been a trend with DNNs, an increase in the amount of training data improves the performance of the model.

Figures 8 and 9 present a graphical comparison of the generated data against the real data for both experimental data sets in terms of the standardized ranges of BG levels. The bar graphs show the mean values with the  $p$ -values that show statistical similarities are

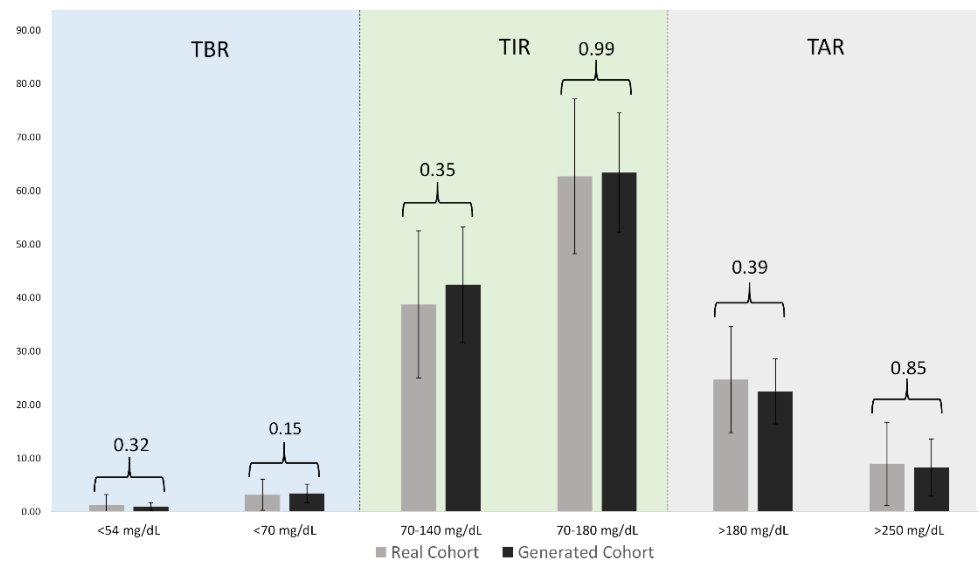
written on top of the quantities in comparison. It can be seen that the generated data sets are significantly similar in terms of both the means of data and the statistical *p*-values. This graph also provides us an insight on the standard deviations of the two data sets.

**Table 1.** A comparison of generated cohort and real cohort: Medtronic’s MiniMed 640G data set (29 patients).

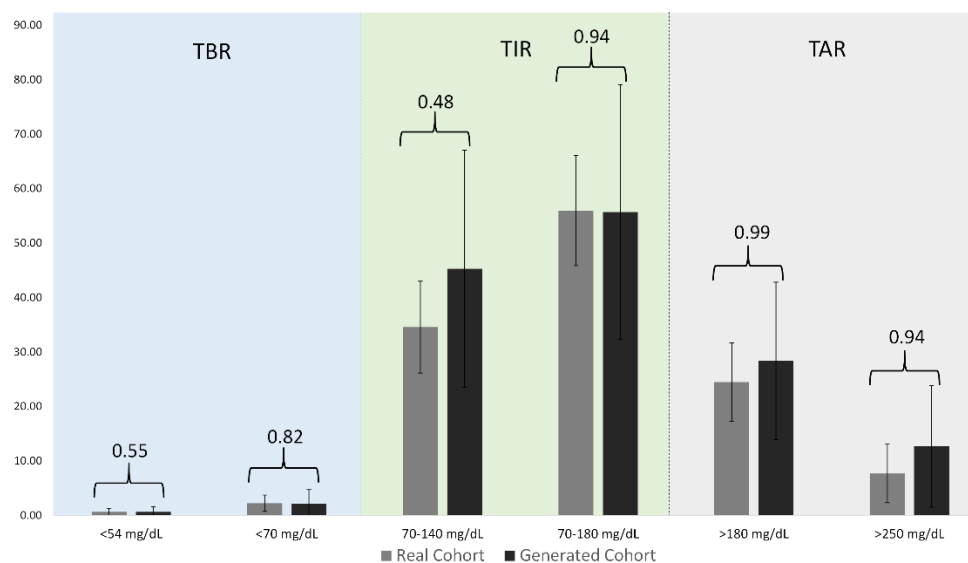
Ranges	Real Cohort	Generated Cohort	<i>p</i> -Values
Below 54 (mg/dL)	1.26(0.16–1.03)	0.90 (0.34–1.15)	0.32
54 to 69 (mg/dL)	3.14 (1.41–4.03)	3.41 (2.43–4.56)	0.15
70 to 140 (mg/dL)	38.72 (29.32–46.81)	42.40 (37.98–47.99)	0.35
70 to 180 (mg/dL)	62.68(55.64–69.45)	63.39 (56.91–70.71)	0.99
180 to 250 (mg/dL)	24.70 (17.52–30.46)	22.46 (18.34–27.51)	0.39
Above 250 (mg/dL)	8.94 (3.43–11.75)	8.25 (4.31–11.21)	0.85
Mean CGM (mg/dL)	159.46 (148.06–173.86)	157.64 (147.76–169.16)	0.86
CV (Percentage)	35.96 (32.89–39.05)	38.56 (35.85–41.88)	0.12
Maximum CGM (mg/dL)	373.03 (358–400)	389.27 (387.93–400)	0.06
Minimum CGM (mg/dL)	43.86 (40–43)	46.21 (40.00–50.20)	0.25

**Table 2.** A comparison of generated cohort and real cohort: Ohio T1D data set (6 patients).

Ranges	Real Cohort	Generated Cohort	<i>p</i> -Values
Below 54 (mg/dL)	0.63 (0.19–0.81)	0.63 (0.06–0.84)	0.55
54 to 69 (mg/dL)	2.20(1.45–2.43)	2.18(0.08–3.64)	0.82
70 to 140 (mg/dL)	34.52(32.07–40.77)	45.24(26.33–63.36)	0.48
70 to 180 (mg/dL)	55.91(50.98–61.97)	55.65(34.92–74.06)	0.94
180 to 250 (mg/dL)	24.45(20.23–28.49)	28.34(15.94–38.83)	0.99
Above 250 (mg/dL)	7.66(4.75–10.25)	12.65(4.65–18.52)	0.94
Mean CGM (mg/dL)	146.53(133.25–156.02)	158.63(127.73–190.56)	0.99
CV (Percentage)	49.28(43.60–56.20)	40.44(34.47–45.89)	0.31
Maximum CGM (mg/dL)	395.67(397.75–400.00)	340.43(331.15–347.81)	0.004
Minimum CGM (mg/dL)	41.00 (40–40)	51.55 (40.08–62.53)	0.12



**Figure 8.** Times in standardized ranges of blood glucose for real and generated cohorts with corresponding *p*-values of the Midtronic’s MiniMed 640G 29 patient data set.



**Figure 9.** Times in standardized ranges of blood glucose for real and generated cohorts with corresponding  $p$ -values of the Ohio 6 patient data set.

## 5. Discussion

The proposed system demonstrated that, first, our Pix2Pix GAN learned from the distributions of BG profiles of diabetic patients labeled with plasma insulin estimation and, second, that it is able to generate new BG values conditioned on plasma insulin values. The generated BG values showed statistical similarity with the original data set. Since the model was trained and tested with two different data sets, the results for both data sets confirmed the effectiveness of our method. However, it is important to note that the approximation was better for the 29 patients' data set as compared to the 6 patients' data set, confirming that the amount of data used for training is a significant performance constraint. The need for more data, though paradoxical when it comes to data generation, is also evident from the theory of GANs [38].

Since, we used a variant of GAN that performs the task of image translation, the numerical data was transformed into graphical data in order to be used with the GAN. This strategy proved efficient for our cause because fundamentally, graphical data are a depiction of numerical data and the transformation was a number-to-number transformation in essence, i.e., the translation task of numerical values to images and then back to numerical values; however, this needs to be carried out with utmost care as slight neglect may lead to information loss and induction of errors. For instance, the expander parameters in the proposed model needs to be selected after rigorous testing or it could induce errors in the measurements. Keeping this in mind, the translation task could well be coupled with a correction factor of some sort to correct for the induced errors. Having said that, there can never be complete compensation for all sorts of losses and errors. Furthermore, the Pix2Pix GAN used in the proposed study learns from an original distribution as well as trains a loss function. This approach makes it possible for it to be used in any sort of image-to-image translation task without the need of designing a customized loss function. Additionally, this GAN is found to be efficient at pushing the output color distribution closer to the input color distribution [22]. Furthermore, the patchGAN discriminator used in our model encourages greater color diversity, which makes it a suitable choice of discriminator for the problem of translating a particular type of heat map into another type.

Since the aim of this work is to move towards a working environment that could simulate scenarios mimicking T1D patients, the correction factors used in the translation task should be evaluated thoroughly for all possible errors and documented accordingly. Furthermore, the translation algorithm should be tested for multiple cohorts and any errors should be quantified. In case of the proposed work, the same translation algorithm worked

equally good for both the data sets. In future, the generation of BG can be conditioned on various other types of data such as carbohydrates or physical activity data. In order to enable such DGMs to be able to approximate the whole of the human glucose–insulin mechanism, they must be trained on a large amount of versatile data. The proposed methodology may prove advantageous when a variety of data is used for glucose conditioning. In the current form, another column of pixel data may be sufficient for a new type of data to be added to the input image. Pix2Pix GAN’s adaptability to various types and sizes of images makes it suitable for such kind of a problem where the formation of images may change because of the addition or removal of data.

From the clinical perspective, the proposed methodology shows promise in the creation of simulation environments that could be used for the development and testing of new treatment methodologies in T1D management. One of the strengths of conventional T1D simulators is the control of various phenomena affecting the glycemic profile, such as the amount and type of meal and insulin, among other factors. These simulators are also capable of reproducing results for a particular combination of the affecting factors. All of this points towards the approximation of causal relationships between these factors and the generation of BG time-series. One way to approximate causal relationships is through recurrence. Introducing snippets of recurring relationships to a CGAN enables it to generate data that depicts a causal relationship to the data it is conditioned on. Once an understanding of the causality is established, generative models could be utilized to generate data in future and hence provide predictions. The predictions, when tested experimentally, may give us a deeper understanding of the biological system at hand and help us devise treatments that are effective in real life.

## 6. Conclusions

Existing T1D simulators are based on physiological models and fail to provide an all-inclusive synthetic T1D patient platform for researchers to work with in the field. Data-driven models have the capability to learn those underlying characteristics of the human body from data that are hard to represent through physiological models. This research work proposed a Pix2Pix GAN model for the generation of BG values conditioned on plasma insulin approximation of T1D patients. The glucose data generated by the proposed system are realistic enough to be used as an alternative to real data. The BG data generated by our model could also be added to real data in order to improve the effectiveness of certain other data driven models, i.e., predictive models or classification models. In future, the generation of BG values could be conditioned on other types of data, such as carbohydrates or physical activity data, for more realistic results.

**Author Contributions:** Conceptualization, O.M., I.C. (Ivan Contreras) and J.V.; data curation, O.M. and A.B.; formal analysis, O.M., I.C. (Ivan Contreras) and A.B.; funding acquisition, I.C. (Ivan Contreras) and J.V.; investigation, O.M., I.C. (Ivan Contreras) and A.B.; methodology, O.M., I.C. (Ivan Contreras), A.B. and J.V.; project administration, I.C. (Ivan Contreras) and J.V.; resources, I.C. (Ignacio Conget), M.G. and J.V.; software, O.M.; supervision, I.C. (Ivan Contreras), I.C. (Ignacio Conget), M.G. and J.V.; validation, O.M., I.C. (Ivan Contreras), A.B., I.C. (Ignacio Conget) and M.G.; visualization, O.M. and I.C. (Ivan Contreras); writing—original draft, O.M., I.C. and A.B.; writing—review and editing, O.M., I.C. (Ivan Contreras), A.B., I.C. (Ignacio Conget), M.G. and J.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the Spanish Ministry of Universities, the European Union through Next GenerationEU (Margarita Salas), the Spanish Ministry of Science and Innovation through grant PID2019-107722RB-C22/AEI/10.13039/501100011033, PID2020-117171RA-I00 funded by MCIN/AEI/10.13039/501100011033 and the Government of Catalonia under 2017SGR1551 and 2020 FI\_B 00965.

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the Ethics Committee of Hospital Clínic de Barcelona (protocol code HCB/2015/0683 and date of approval 22 September 2015).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Restrictions apply to the availability of these data. Data was obtained from the Spanish Consortium on Artificial Pancreas and Diabetes Technology and are available on request from the corresponding author with the permission of the Spanish Consortium on Artificial Pancreas and Diabetes Technology.

**Conflicts of 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.

## References

1. Sá-Couto, C.; Patrão, L.; Maio-Matos, F.; Pêgo, J.M. Biomedical simulation: Evolution, concepts, challenges and future trends. *Acta. Med. Port.* **2016**, *29*, 860–868. [[CrossRef](#)] [[PubMed](#)]
2. Man, C.D.; Micheletto, F.; Lv, D.; Breton, M.; Kovatchev, B.; Cobelli, C. The UVA/PADOVA type 1 diabetes simulator: New features. *J. Diabetes Sci. Technol.* **2014**, *8*, 26–34. [[CrossRef](#)]
3. Wilinska, M.E.; Chassin, L.J.; Acerini, C.; Allen, J.M.; Dunger, D.; Hovorka, R. Simulation Environment to Evaluate Closed-Loop Insulin Delivery Systems in Type 1 Diabetes. *J. Diabetes Sci. Technol.* **2010**, *4*, 132–144. [[CrossRef](#)] [[PubMed](#)]
4. Alkhalifah, T.; Wang, H.; Ovcharenko, O. MLReal: Bridging the gap between training on synthetic data and real data applications in machine learning. *Eur. Assoc. Geosci. Eng.* **2021**, *2021*, 1–5. [[CrossRef](#)]
5. Jan, S.V.S.; Geris, L. Modelling towards a more holistic medicine: The Virtual Physiological Human (VPH). *Morphologie* **2019**, *103*, 127–130. [[CrossRef](#)]
6. Shengli, W. Is Human Digital Twin possible? *Comput. Methods Programs Biomed. Updat.* **2021**, *1*, 100014. [[CrossRef](#)]
7. Kovatchev, B.P.; Breton, M.; Man, C.D.; Cobelli, C. In Silico Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes. *J. Diabetes Sci. Technol.* **2009**, *3*, 44–55. [[CrossRef](#)]
8. Cobelli, C.; Man, C.D.; Sparacino, G.; Magni, L.; De Nicolao, G.; Kovatchev, B.P. Diabetes: Models, Signals, and Control. *IEEE Rev. Biomed. Eng.* **2009**, *2*, 54–96. [[CrossRef](#)]
9. Hester, R.L.; Iliescu, R.; Summers, R.; Coleman, T.G. Systems biology and integrative physiological modelling. *J. Physiol.* **2011**, *589*, 1053–1060. [[CrossRef](#)]
10. Vettoretti, M.; Facchinetti, A.; Sparacino, G.; Cobelli, C. Type-1 Diabetes Patient Decision Simulator for In Silico Testing Safety and Effectiveness of Insulin Treatments. *IEEE Trans. Biomed. Eng.* **2017**, *65*, 1281–1290. [[CrossRef](#)]
11. Roversi, C.; Vettoretti, M.; Del Favero, S.; Facchinetti, A.; Sparacino, G. Modeling Carbohydrate Counting Error in Type 1 Diabetes Management. *Diabetes Technol. Ther.* **2020**, *22*, 749–759. [[CrossRef](#)] [[PubMed](#)]
12. American Diabetes Association. 15. Diabetes Care in the Hospital: Standards of Medical Care in Diabetes—2021. *Diabetes Care* **2021**, *44* (Suppl. S1), S211–S220. [[CrossRef](#)] [[PubMed](#)]
13. Karimy, M.; Koohestani, H.R.; Araban, M. The association between attitude, self-efficacy, and social support and adherence to diabetes self-care behavior. *Diabetol. Metab. Syndr.* **2018**, *10*, 86. [[CrossRef](#)] [[PubMed](#)]
14. Bertachi, A.; Biagi, L.; Contreras, I.; Luo, N.; Vehí, J. Prediction of Blood Glucose Levels And Nocturnal Hypoglycemia Using Physiological Models and Artificial Neural Networks. *KHD@IJCAI* **2018**, 85–90.
15. Mujahid, O.; Contreras, I.; Vehí, J. Machine Learning Techniques for Hypoglycemia Prediction: Trends and Challenges. *Sensors* **2021**, *21*, 546. [[CrossRef](#)]
16. Contreras, I.; Vehí, J. Artificial Intelligence for Diabetes Management and Decision Support: Literature Review. *J. Med. Internet Res.* **2018**, *20*, e10775. [[CrossRef](#)]
17. Nalisnick, E.; Matsukawa, A.; Teh, Y.W.; Gorur, D.; Lakshminarayanan, B. Do deep generative models know what they don't know? *arXiv* **2018**, arXiv:1810.09136.
18. Nikzad, M.; Movagharnejad, K.; Talebna, F. Comparative Study between Neural Network Model and Mathematical Models for Prediction of Glucose Concentration during Enzymatic Hydrolysis. *Int. J. Comput. Appl.* **2012**, *56*, 43–48. [[CrossRef](#)]
19. Noguier, J.; Contreras, I.; Mujahid, O.; Beneyto, A.; Vehí, J. Generation of Individualized Synthetic Data for Augmentation of the Type 1 Diabetes Data Sets Using Deep Learning Models. *Sensors* **2022**, *22*, 4944. [[CrossRef](#)]
20. Buczak, A.L.; Babin, S.; Moniz, L. Data-driven approach for creating synthetic electronic medical records. *BMC Med. Inform. Decis. Mak.* **2010**, *10*, 59. [[CrossRef](#)]
21. Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic data augmentation using GAN for improved liver lesion classification. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 289–293. [[CrossRef](#)]
22. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
23. Jordon, J.; Yoon, J.; van der Schaar, M. PATE-GAN: Generating synthetic data with differential privacy guarantees. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
24. Frid-Adar, M.; Diamant, I.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing* **2018**, *321*, 321–331. [[CrossRef](#)]

25. Lee, C.-H.; Liu, Z.; Wu, L.; Luo, P. Maskgan: Towards diverse and interactive facial image manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5549–5558.
26. Meskys, E.; Liaudanskas, A.; Kalpokiene, J.; Jurcys, P. Regulating deep fakes: Legal and ethical considerations. *J. Intellect. Prop. Law Pract.* **2020**, *15*, 24–31. [[CrossRef](#)]
27. Dornis, T.W. Artificial Creativity: Emergent Works and the Void in Current Copyright Doctrine. *Yale J. Law Technol.* **2020**, *22*, 1.
28. Ravuri, S.; Lenc, K.; Willson, M.; Kangin, D.; Lam, R.; Mirowski, P.; Fitzsimons, M.; Athanassiadou, M.; Kashem, S.; Madge, S.; et al. Skillful Precipitation Nowcasting using Deep Generative Models of Radar. *arXiv* **2021**, arXiv:2104.00954.
29. Ruthotto, L.; Haber, E. An introduction to deep generative modeling. *GAMM-Mitteilungen* **2021**, *44*, e202100008. Available online: <https://arxiv.org/abs/2103.05180> (accessed on 20 August 2022). [[CrossRef](#)]
30. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [[CrossRef](#)]
31. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
32. Ahmad, S.; Ramkissoon, C.; Beneyto, A.; Conget, I.; Giménez, M.; Vehi, J. Generation of Virtual Patient Populations that Represent Real Type 1 Diabetes Cohorts. *Mathematics* **2021**, *9*, 1200. [[CrossRef](#)]
33. Hovorka, R.; Canonico, V.; Chassin, L.J.; Haueter, U.; Massi-Benedetti, M.; Federici, M.O.; Pieber, T.R.; Schaller, H.C.; Schaupp, L.; Vering, T.; et al. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiol. Meas.* **2004**, *25*, 905–920. [[CrossRef](#)]
34. Marling, C.; Bunescu, R. The OhioT1DM Dataset for Blood Glucose Level Prediction: Update 2020. *CEUR Workshop Proc.* **2020**, *2675*, 71–74.
35. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
36. Vadivel, A.; Sural, S.; Majumdar, A.K. Human color perception in the HSV space and its application in histogram generation for image retrieval. *Proc. SPIE* **2005**, *5667*, 598–609. [[CrossRef](#)]
37. PHerrero, P.; Alalitei, M.A.; Reddy, M.; Georgiou, P.; Oliver, N. Robust Determination of the Optimal Continuous Glucose Monitoring Length of Intervention to Evaluate Long-Term Glycemic Control. *Diabetes Technol. Ther.* **2021**, *23*, 314–319. [[CrossRef](#)]
38. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [[CrossRef](#)]