A Personalized and Adaptive Insulin Bolus Calculator Based on Double Deep Q-Learning to Improve Type 1 Diabetes Management

Giulia Noaro, Taiyu Zhu, *Graduate Student Member, IEEE*, Giacomo Cappon, Andrea Facchinetti, and Pantelis Georgiou, *Senior Member, IEEE*

***Abstract*—Mealtime insulin dosing is a major challenge for people living with type 1 diabetes (T1D). This task is typically performed using a standard formula that, despite containing some patient-specific parameters, often leads to sub-optimal glucose control due to lack of personalization and adaptation. To overcome the previous limitations here we propose an individualized and adaptive mealtime insulin bolus calculator based on double deep Q-learning (DDQ), which is tailored to the patient thanks to a personalization procedure relying on a two-step learning framework. The DDQ-learning bolus calculator was developed and tested using the UVA/Padova T1D simulator modified to reliably mimic real-world scenarios by introducing multiple variability sources impacting glucose metabolism and technology. The learning phase included a long-term training of eight sub-population models, one for each representative subject, selected thanks to a clustering procedure applied to the training set. Then, for each subject of the testing set, a personalization procedure was performed, by initializing the models based on the cluster to which the patient belongs. We evaluated the effectiveness of the proposed bolus calculator on a 60-day simulation, using several metrics representing the goodness of glycemic control, and comparing the results with the standard guidelines for mealtime insulin dosing. The proposed method improved the time in target range from 68.35% to 70.08% and significantly reduced the time in hypoglycemia (from 8.78% to 4.17%). The overall glycemic risk index decreased from 8.2 to 7.3, indicating the benefit of our method when applied for insulin dosing compared to standard guidelines.**

***Index Terms*—Bolus calculator, deep learning, insulin therapy, reinforcement learning, type 1 diabetes.**

Manuscript received 6 October 2022; revised 26 January 2023; accepted 17 February 2023. Date of publication 27 February 2023; date of current version 5 May 2023. This work was supported by the Italian Minister for Education (MIUR) under the initiative "Departments of Excellence" under Grant Law 232/2016. *(Corresponding author: Andrea Facchinetti.)*

This work did not involve human subjects or animals in its research.

Giulia Noaro, Giacomo Cappon, and Andrea Facchinetti are with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (e-mail: noarogiuli@dei.unipd.it; cappongi@dei.unipd.it; andrea.facchinetti@unipd.it).

Taiyu Zhu and Pantelis Georgiou are with the Centre for Bio-Inspired Technology, Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: taiyu.zhu17@imperial.ac.uk; pantelis@imperial.ac.uk).

Digital Object Identifier 10.1109/JBHI.2023.3249571

I. Introduction

THE number of people diagnosed with type 1 diabetes (T1D) has been rising worldwide in recent years [1]. The management of T1D requires lots of effort from those who are affected, as their body is no longer able to produce insulin, one of the key hormones in blood glucose (BG) regulation. To keep BG within the safe normoglycemic range, i.e., [70-180] mg/dL [2], T1D standard therapy suggests the administration of slow-acting insulin injections, usually performed once or twice a day, to control BG in fasting conditions, together with fast-acting insulin injections in correspondence of meals, to counteract the subsequent rising of BG. According to the standard guidelines, the latter dosage should be calculated following the formula in (1), hereafter defined as the standard bolus calculator (BC_S) at mealtime [3]

$$BC_S = \frac{CHO}{CR} + \frac{G_c - G_t}{CF} - IOB \tag{1}$$

where CHO (g) is the meal carbohydrate intake, CR (g/U) and CF (mg/dL/U) are the insulin-to-carbohydrates ratio and the correction factor, i.e., two therapy parameters tuned by the clinician [4], G_c (mg/dL) is the mealtime BG level, G_t (mg/dL) is the target BG level, and IOB (U) is the insulin on board, indicating the amount of previously injected insulin that is still acting in the organism [5].

Leveraging the UVA/Padova T1D simulator, a FDA-approved environment that can be used to test new insulin therapies via in-silico clinical trials [6], we showed in [7], [8], [9] that (1) can be sub-optimal and potentially harmful in some situations. Indeed, excessive insulin dosage could lead to low BG levels ($<$ 70 mg/dL), i.e. hypoglycemia [10], a major obstacle to glycemic control for many patients, that lead to dangerous complications including neurological damage, coma, and even death [11]. On the other hand, underestimation of mealtime insulin can result in prolonged hyperglycemic events (BG $>$ 180 mg/dL), which are responsible for the development of microvascular and macrovascular complications, and retinopathy [12].

These results highlighted the need to improve mealtime insulin dosing by adopting newer, more refined, and, most of all, personalized and adaptive approaches.

The increased amount of available data collected thanks to new technologies such as minimally-invasive continuous glucose monitoring (CGM) sensors [13], [14], [15], which allow tracking glucose levels in almost real-time, and insulin pumps, that enabled automated insulin delivery [16], fostered the development of deep learning-based decision support algorithms [17] in T1D management and smart bolus calculators integrating powerful data-driven strategies aimed at improving the standard bolus calculator based on different optimization techniques.

For instance, in Herrero et al. [18], run-to-run control and case-based reasoning were used to provide insulin dose recommendations based on the retrospective optimization of the therapy parameters of (1) (i.e., CR and CF) performed on a daily basis, while in Fabris et al. [19] they proposed the use of two methods to inform insulin dosing with biosignals from wearable sensors, i.e. insulin sensitivity estimated through CGM signal and the step count. In addition, in our recent work [20], we developed a new bolus calculator based on LASSO regression by leveraging a simulated dataset, while in Cappon et al. [21], a simple neural network was preliminarily investigated for such a scope. All these studies produced positive results, encouraging further efforts on this direction. However, the design of a supervised learning framework is far from being trivial in such a context, due to the difficulty in retrieving the optimal target of the learning task. Indeed, when considering data collected from people living with T1D, the administered mealtime insulin bolus is in most cases sub-optimal, leading to poor postprandial glycemic control. Moreover, the insulin amount estimation is highly patient-dependent, making it difficult to train a general model which is valid for different subjects. Hence, the difficulty of having reliable data, which includes an optimal insulin bolus, together with the need for a model which is tailored to the patient's needs, justifies the application of reinforcement learning (RL) for such a task. Indeed, the application of RL is particularly suitable, since it aims at automating the decision-making process which characterized T1D management, by learning through the interaction with a specific environment to achieve a goal, without the need for labeled data. The work of Zhu et al. [22] proposed a preliminary study in this direction, by applying a deep RL algorithm (deep deterministic policy gradient) for the design of an insulin bolus advisor. The use of this methodology to develop a bolus calculator showed encouraging results, suggesting that RL is suitable for such a task. However, the simulated environment used in [22] did not consider relevant variability sources which could impact glycemic control, e.g., CGM measurement error, and the size of virtual cohort is relatively small ($n = 20$).

In this article, we addressed the issue of mealtime insulin dosing, being the most demanding and patient-dependent action among all those required by the standard T1D therapy. For such a scope, we leveraged an RL algorithm, in particular double deep Q-learning (DDQ), which relies on a high level of individualization, by integrating a two-step learning process together with a clustering procedure, to ensure a proper personalization of the algorithm. The proposed DDQ-learning bolus calculator was developed and tested on an updated version of the FDA-accepted UVA/Padova T1D simulator [6], which allows recreating a realistic simulation environment, by including multiple variability sources in a large virtual cohort of 100 subjects.

The use of a simulation tool that reliably mimics a real-world scenario is crucial to preliminary assess the benefit that an RL technique could bring when applied for the purpose of T1D management, and in particular insulin dosing, before moving to an ad-hoc clinical setting.

The paper is organized as follows. Section II reports the basic principles of DDQ-learning together with the design of the algorithm for the specific task of mealtime insulin dosing. Section III describes the simulation environment used for this purpose, and specifies the experimental set-up. In Section IV and V we discuss the two-step learning framework and the assessment of the resulting models respectively, while Section VI reports some final considerations and future developments of the proposed work.

II. Methods

Hereafter, we will formulate the insulin dosing task as an RL problem. First, in Section II-A we will describe in detail the core algorithm adopted in our strategy, i.e., DDQ-learning, which is a variant of Q-learning, a popular RL technique that learns the value of an action in a particular state and aims at finding an optimal policy in the sense of maximizing the expected value of the total reward over any successive step, starting from the current state [23]. Then, in Section II-B, we will present how we integrated DDQ-learning for the specific purpose of developing a new mealtime insulin bolus calculator.

A. Background on Double Deep Q-Learning

In general, the goal of RL is training an agent to perform a task thanks to the interaction with a defined environment, by assigning a specific reward to each agent's action [24]. In particular, for each discrete time step t the environment can be represented by the state vector (s_t), which better describes the current status of the environment among all possible vectors of the state space S. Hence, the state at time t, is used by the agent to choose an action (a_t) from all the possible sets of actions (A), according to its policy ($\pi : S \rightarrow A$). The action, which is selected based on the policy π, is applied to the environment, that evolves into the subsequent state (s_{t+1}). Finally, the action related to the specific state is evaluated through a reward (r_t), that is assigned to the corresponding state-action pair (s_t, a_t).

In this context, the goal of the algorithm is learning an optimal policy π which maximizes the cumulative discounted future reward (i.e., the return G_t):

$$G_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_{\tau+1} \tag{2}$$

where γ is the discount factor, which takes values between [0,1], and determines how much the rewards in the distant future should be considered within the calculation.

In Q-learning, in order to learn the optimal policy and maximize the reward defined in (2), the so-called action-value function is used [23]:

$$Q_\pi(s_t, a_t) = \mathbb{E}[G_t | s_t, a_t, \pi] \tag{3}$$

which is the expected return given the state s_t and action a_t under a specific policy π, representing the goodness of the action for the given state. The optimal action-value function $Q^*(s, a)$ can be defined as the one maximizing (3), and satisfying the Bellman optimality equation [23]:

$$Q^*(s_t, a_t) = \mathbb{E}\left[r_{t+1} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) | s_t, a_t\right] \tag{4}$$

When the number of possible state-action pairs is high, the amount of time required to explore each state and apply a specific action is impracticable. Thus, generating the so-called Q-table, which stores $Q^*(s_t, a_t)$ for each state-action pair, becomes an excessively time-consuming process. To overcome this issue, in deep Q learning, the exact action-value function is replaced by a function approximator based on deep learning, i.e., a deep Q network (DQN). A DQN is a deep multi-layered neural network parametrized by its weight vector θ, which for a given state s provides as output the approximation of $Q^*(s, a)$, i.e., $Q_\theta(s, a)$ for all the possible actions in A.

Improving learning stability of DQN: In this article, we took advantage of two important factors related to the DQN algorithm, which increase learning stability: the experience replay and the target network [25]. By leveraging experience replay, we stored the agent's experiences in the form $(s_t, a_t, r_{t+1}, s_{t+1})$ in a cyclic buffer called replay memory, from which a minibatch was randomly drawn to update the DQN, instead of performing such a task at each time step and thus leading to poor stability. The use of experience replay also helps in breaking the correlation between two consecutive samples while training the network. The second stabilizing method was obtained by means of a target network, which is a separate network having weights θ^-, that is initially equal to the ones of the network enacting the policy. During training, the weights of the target network θ^- were updated to match the policy network θ after a fixed number of steps [25].

In this study, we applied Double Deep Q-learning (DDQ), a variant of deep Q-learning, which leverages the target network θ^- to tackle the maximization bias issue, i.e., the systematic overestimation of the action-value function due to the maximization step in (4), which characterized such an algorithm. The max operator in DQN, in (4), uses the same values both to select and evaluate action. This makes it more likely to select overestimated values, resulting in overoptimistic value estimates. To prevent this, DDQ-learning decouples the action selected from the action evaluation by leveraging the target network parametrized by θ^- [26]. Hence, in DDQ-learning the target Q-value is computed as follows:

$$Q(s_t, a_t) = r_{t+1} + \gamma Q_\theta \left(s_{t+1}, \underset{a_{t+1}}{\text{argmax}}\, Q_{\theta^-}(s_{t+1}, a_{t+1})\right) \tag{5}$$

Note that, the selection of the action is due to the policy network θ, while the second network θ^- is used to fairly evaluate the value of this policy [26].

Fig. 1 visually summarizes the different steps and elements which compose the DDQ-learning algorithm.

B. Development of the Insulin Bolus Calculator Based on Double Deep Q-Learning

In this section, we present how we used the DDQ-learning algorithm to develop a mealtime insulin bolus calculator. In particular, at mealtime t, the patient needs to estimate and inject the insulin dose to counteract the glycemic excursion due to the meal. Within this framework, the mealtime condition is represented by a specific state s_t, composed of easily accessible physiological parameters of the T1D individual as described in II-B1. Based on the current mealtime status, the best action a_t selected by the policy was applied to the patient, i.e., the best insulin amount to be injected, as defined in II-B2. Lastly, after the insulin amount delivery, the postprandial glycemic outcome was evaluated through the reward function specified in II-B3.

1) State Vector Choice: In this work, we used easily accessible variables together with patient-specific therapy parameters to describe the mealtime condition, which provides a comprehensive view of the prandial status.

We defined the state vector at time t (s_t) by extracting, for each meal, the following variables: the current CGM measurement G_c [mg/dL]; the carbohydrate content of the meal (CHO) [g]; the CGM rate of change (ROC) [mg/dL/min], which gives insight into the glucose dynamics, by indicating whether glucose level is falling or rising and to what extent; the prandial insulin-to-carbohydrates ratio (CR) [g/U], a therapy parameters indicating how many grams of CHO will be covered by one unit of insulin and which could vary from meal to meal; together with the prandial insulin dose computed through the standard therapy (BC_S) [U] as in (1). In this process of feature selection, we followed our previous work's rationale [7], which involved the exclusion of variables that, in this specific application, assumed a constant value for each meal, e.g., the body weight (BW), the correction factor (CF), the target glucose (G_t). As a result, we defined s_t as:

$$s_t = \{G_c, CHO_t, ROC_t, CR_t, BC_{S_t}\}. \tag{6}$$

2) Set of Possible Actions: In our scenario, there are multiple types of actions which could potentially be adopted, such as the estimation of the insulin dose itself, or the correction of the dose suggested by BC_s. Since the number of possible insulin amounts associated with a state is considerably high, we decided not to directly estimate the mealtime bolus, but to correct the dose suggested by BC_s, which thus will be considered as a starting value. This choice was made to avoid an excessive number of possible actions, which could lower the performance of our algorithm. Hence, we defined an action a_t as a percentage modulation of the mealtime insulin dose suggested by the standard therapy. In particular, the BC_s dose can be decreased or increased by a percentage α, which ranges from these possible values $\alpha = \{\pm 25, \pm 20, \pm 10, 0\}\%$. The motivation behind this choice

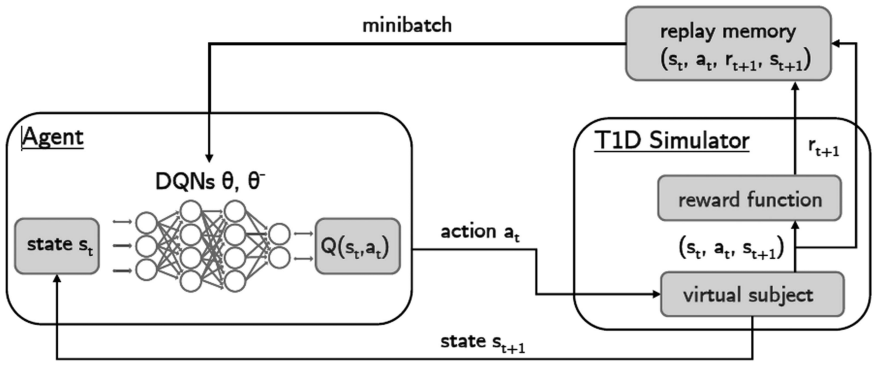

Fig. 1. Representative scheme of the DDQ-learning framework applied to the T1D simulation environment. Within the **Agent** block, the state s_t feeds both the policy and the target network, which will be used to approximate the Q(s_t, a_t). Then, the action a_t associated with the maximum estimated Q-value will be an input of the **T1D Simulator** block, and in particular to the virtual subject, which will receive the insulin dose corresponding to the chosen action. At time step $t+1$ the environment evolves into the subsequent state s_{t+1}, hence the reward r_{t+1} is computed and the transitions (s_t, a_t, r_{t+1}, s_{t+1}) are stored within the replay memory and will be used every N steps to update the policy network.

lies in the results obtained in our previous work [7] where multiple state-of-art insulin adjustment methods were preliminarily assessed in silico. The literature approach which was correcting the dose based on a percentage modulation resulted the safest method under specific mealtime conditions [27], potentially indicating that the insulin correction should also depend on the entity of the meal and not be fixed to constant values.

Following this rationale, at each time step t, the chosen action $a_t = \alpha_t$ was applied to the bolus calculator as follows:

$$BC_{ddqn}(t) = BC_s(t) + \alpha_t BC_s(t) \tag{7}$$

where $BC_{ddqn}(t)$ represents the insulin amount suggested by the DDQN algorithm at time step t.

3) Reward Function: We employed as reward function the piecewise constant function presented in the preliminary work of [22]. The constant values, i.e. the weights penalizing the different glycemic ranges, which describes the reward function, were tuned retrospectively in order to adapt the weights to the different simulation environment used for this study, which includes multiple variability sources. Indeed, the piecewise linear reward function was selected since it provided high flexibility in penalizing narrow glycemic intervals with different weights. As mentioned above, we tuned the weights retrospectively by testing different values and selecting the ones which allowed us to achieve the best trade-off between reducing time in hypoglycemia, without impacting on hyperglycemia, while improving the time in range. Hence, the aim being evaluating the glycemic control following the current meal at time instant t, we assigned the weight selected during the previous step to each postprandial CGM sample between the current time instant t and t^*, which corresponds either to the following meal or a postprandial 6 hours interval if the next meal occurred at a longer time distance. In particular, the reward equation is reported in (8):

$$r_t = \frac{1}{t^* - t} \sum_{k=t}^{t^*} f_R(G_k) \tag{8}$$

where $t^* = \min(t + 6h, t + 1)$. Note that weights were assigned differently based on the glycemic range. Particularly,

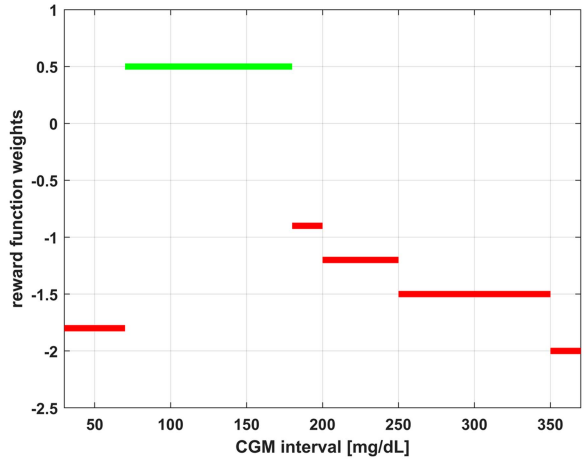

Fig. 2. Reward function employed for the proposed DDQN algorithm. Each CGM interval is associated with a constant value used within the reward function. Green line represents the euglycemic range, while red lines indicate intervals associated to adverse event.

hypoglycemia was penalized more compared to hyperglycemia, since a hypoglycemic excursion is riskier than a hyperglycemic excursion having the same amplitude. The selected rewards were tuned by trial-and-error and associated with each glycemic interval as follows:

$$f_R(G_k) = \begin{cases} +0.5 & \text{if } 70 \leq \mathrm{G_k} \leq 180 \\ -0.9 & \text{if } 180 < \mathrm{G_k} \leq 200 \\ -1.2 & \text{if } 200 < \mathrm{G_k} \leq 250 \\ -1.5 & \text{if } 250 < \mathrm{G_k} \leq 350 \\ -1.8 & \text{if } 30 < \mathrm{G_k} < 70 \\ -2 & \text{else} \end{cases} \tag{9}$$

As reported in (9), a glucose level within the target range was associated with a positive reward. The highest penalty was assigned to values in hypoglycemia, while hyperglycemic values were not penalised all equally, having different weights based on the severity of the interval. Fig. 2 depicts the different weights assigned to each glycemic interval.

TABLE I
MINIMUM AND MAXIMUM VALUES OF THE POSSIBLE CHO AMOUNT AND TIME OF CONSUMPTION FOR THE DIFFERENT MEALS

Meal type	CHO amount [g]	Meal timing
breakfast	[19–97]	[6.30 am – 8.00 am]
lunch	[31–124]	[11.30 am – 1.00 pm]
dinner	[28–140]	[6.00 pm – 8.30 pm]

TABLE II
HYPERPARAMETERS USED FOR BOTH THE SUB-POPULATION MODEL TRAINING AND PERSONALIZED MODEL TUNING

Hyperparameter	Value
Number of random episodes before learning starts	30
Sub-population ϵ-greedy exploration	$0.9 \rightarrow 0.1$
Number of episodes after which the target DQN is updated	50
Replay memory maximum size	800
Discount factor γ	0.95
Minibatch size	32
Learning rate	0.001
Hidden nodes of the DQNs	[32, 16]

The hyperparameters do not differ among the virtual subjects.

C. Implementation Details on the DDQN Algorithm

In this Section, implementation details are reported to further clarify the training process. During the learning phase, all the hyperparameters highlighted in Section II-A have been set following a trial-and-error procedure, by analyzing the simulations. The values selected for each hyperparameters are reported in Table II, while the weights of the DQNs were initially initialized as random.

Before starting DDQN training, the replay memory introduced in Section II-A was first filled by applying random actions chosen from the set of possible actions of Section II-B2. Particularly, the hyperparameter related to the number of episodes before the start of learning was set as the one that allowed to explore a sufficient number of meals for a first fill of the replay memory. After this phase, characterized by actions chosen completely random, we started the learning procedure by selecting either a random action with probability ϵ or an action based on the agent with probability $1 - \epsilon$. In particular, the agent makes use of a DQN, having two hidden layers composed by 32 and 16 nodes respectively, which maps state-space ($\mathbb{R}^5$) to action-space ($\mathbb{R}^7$). The weights of the policy network were updated at the end of one episode, according to minibatch gradient descent, by randomly sampling from the replay memory, while the target network was updated after a fixed number of episodes.

III. SIMULATION ENVIRONMENT

A. The UVA/Padova T1D Simulator

The proposed algorithm is trained and tested by means of a well-established computer simulation software: the UVA/Padova T1D Simulator (T1DS). Indeed, due to the nature of RL techniques, the use of a simulation environment is particularly suitable, being the learning of the model achieved through the interaction with the environment and thus, following a trial-and-error procedure. Performing such a process on a virtual subject is key to avoiding dangerous situations within a real clinical setting.

Briefly, T1DS [6], consists of a mathematical model describing the glucose-insulin dynamics in people with T1D using 13 differential equations and more than 30 parameters to represent the large inter-subject variability. T1DS is equipped with 100 virtual subjects, each associated with a different realization of the parameter set, which can be used to assess the performance of new strategies for T1D management by designing ad-hoc in silico clinical trials. The software has been accepted in 2008 by the Food and Drug Administration as an alternative to preclinical trials and it is widely used by the diabetes technology research community [8], [19], [20], [22], [28].

In this work, we used an updated version of the FDA-accepted UVA/Padova T1D Simulator [6], which includes several elements that enhance the realism of the simulated scenarios. These improvements include a dedicated model to describe the dawn phenomenon and time-varying therapy parameters modelled as in [29], a model of CGM sensor measurement error [30], [31], and a behavioural model of people with T1D [32]. Note that the simulated CGM measurements have a sampling time of 5 minutes, i.e. 288 CGM readings are available for each day.

B. Simulated Scenario

In this work, the virtual cohort was composed of 100 adult subjects, and the total population was divided into two different subgroups, hereafter labeled as Set A and Set B. We obtained the two subsets by randomly selecting 50 subjects for each group while performing a stratified split, in which distribution of the average CR pattern, the CF, and the body weight (BW) variables between the two groups remained the same. The stratification was performed to ensure homogeneity between the two sets, being the aforementioned parameters used to identify the subjects used to train the population models within the first step of the learning framework. Hence, Set A was employed for the development of the first learning step, while Set B was retained to fine-tune the generalized models and, lastly, test the performances. The first stage, applied on Set A, consisted of a long-term training of K population models on 1200 simulated days, while the personalization of the resulting DDQN models was carried out on Set B, on a 180-day simulation as performed in [22]. Such a long simulation time was needed since meals are infrequent events within a day (three meals per day).

One day of simulation included three meals per day: breakfast, lunch and dinner. Both meal timing and CHO content of the meals were extracted from uniform distributions to match the data reported in [33]. Table I reports the minimum and maximum values of the distributions from which the meal timing and CHO amount are drawn.

Moreover, to further improve the realism of the simulated scenarios, meal carbohydrates counting error has been modeled and added to the original meal amount as described in [34]. Of note, no corrective actions (rescue carbohydrates intakes or corrective insulin boluses) were included during the simulations to fairly compute the reward corresponding to the action chosen by the algorithm. Indeed, a corrective action could add confounding factors within the rewarding process, since it would bring glycemic levels back into the normoglycemic range, not letting us assess the efficacy of the chosen action alone. As a result, the only control action a_t allowed between mealtime at time t and the following meal at time $t + 1$ was represented by the meal insulin bolus suggested by the policy.

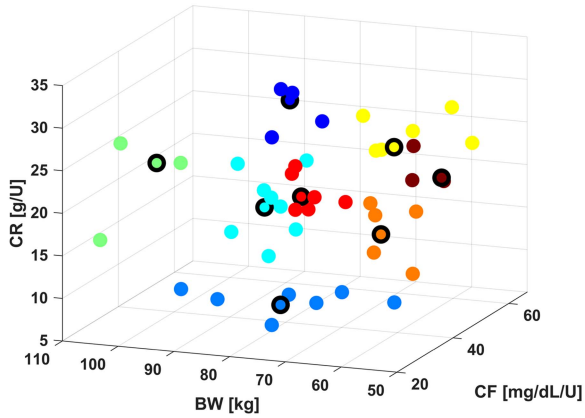

Fig. 3. Scatter plot of the 8 different clusters resulting from the application of the K-medoids algorithm. Each of the 50 subjects is represented by a specific BW, CF and average CR value. The sub-population subjects, i.e., the medoids of each cluster, are outlined in black.

IV. Learning of the DQN

Learning of our model was performed on the first set of 50 subjects extracted in Section III, following the same two-step learning rationale used in Zhu et al. [22] In particular, in the first step we used a long-term simulation to train a set of K population models able to represent the inter-subject physiological variability observed in people with T1D, as described below in Section IV-A (Step 1). Secondly, for a given subject, a specific model is selected between the K population models and then personalized in order to better fit his/her peculiar physiology, as reported in Section IV-B (Step 2).

A. Step 1: Sub-Population Model Training

In order to obtain K sub-population models which are able to describe the variability in T1D subject's physiology, we performed a sub-population model training by dividing Set A of subjects into K different groups that share similar characteristics. To achieve this goal, each subject was described with three patient-specific parameters, i.e., the average CR value of the daily pattern, the CF variable, and the BW. Then, we applied the K-medoids clustering algorithm [35], in order to divide the 50 subjects composing the used population into K clusters. The k-medoids method was selected in contrast to k-means, as it choses datapoints as centers (medoids) of the clusters. This procedure allowed us to extract K subject to be used as population subjects. In particular, $K = 8$ was identified as the optimal number of clusters by employing the elbow method. Finally, we selected the $K = 8$ representative subjects as the medoids of each cluster. Fig. 3 depicts, with different colours, the 8 identified clusters and the respective representative subjects, highlighted with a black circle.

B. Step 2: Personalization of DQN-Learning Models on Patient-Specific Data

In the second step, the sub-population DDQNs were fine-tuned and thus, further personalized on each subject of Set B. Indeed, after having defined the cluster to which each virtual patient of Set B belongs, the DDQN weights were initialized with the one resulting from the previous phase, i.e., θ_k and θ_k^-, where k indicates the k-th sub-population subject.

By setting specific safety constraints, which limits the meal insulin correction, this phase of individualization could also be safely carried out on the real subject within a clinical trial setting. In addition to the possibility of introducing security constraints, this phase may be performed using simulation tools which leverage real data, such as [36], thus enabling the fine-tuning of the generalized model on simulations derived from patients' real data.

V. Evaluation of the DQN Algorithm

The assessment of the RL algorithm was performed on 60 simulated days, where the experimental set-up was the one described in Section III, and the population cohort was composed of 50 subjects (Set B). Moreover, to evaluate the efficacy of the proposed methodology, results were compared to 60 days of simulation in which the standard bolus calculator reported in (1) was used for the insulin dosing task. It should be pointed out that, the daily scenarios in terms of mealtimes and CHO amounts composing the meal within the 60-day testing period differ from those within the 180-day personalization period of Section IV-B. Lastly, to ensure a fair comparison between the two insulin dosing methods and to allow replicable analysis, the seed of the random number generator was fixed for each virtual subject.

A. Metrics

From the 60-day glucose profile, we extracted several metrics evaluating glycemic control, which are widely adopted by the diabetes research community [37], [38]. In particular, we derived three metrics related to the the percentage of time spent within the different glycemic ranges, that is the normoglycemic range (TIR), i.e., $70 \leq CGM \leq 180$ mg/dL, below this range (TBR), i.e., $CGM < 70$ mg/dL, and above this range (TAR), i.e., $CGM > 180$ mg/dL. Moreover, we computed two popular indices used to quantify the risk of hypo- and hyperglycemia, namely the low blood glucose index (LBGI) and high blood glucose index (HBGI) respectively, together with the overall blood glucose risk index (BGRI), which sums the two contributions in one risk index, indicating the goodness of the overall glycemic control [39]. In addition, the median number of hypo- and hyperglycemic events per day was extracted to assess the benefit introduced by the proposed algorithm. Lastly, to evaluate the statistical significance of the resulting metric distributions we applied a paired t-test with significance level equal to 5% to those metrics having a Gaussian distribution based on the Lilliefors test (i.e., TIR, TAR) [40], while the Wilcoxon test with significance level equal to 5% was used for the TBR metric, which showed a non-Gaussian distribution.

B. Results

In Fig. 4, the mean and the corresponding standard deviation extracted from all the glucose profiles of the virtual subjects belonging to the testing set are shown for both the DDQN algorithm and the standard therapy. It is noted that, considering the

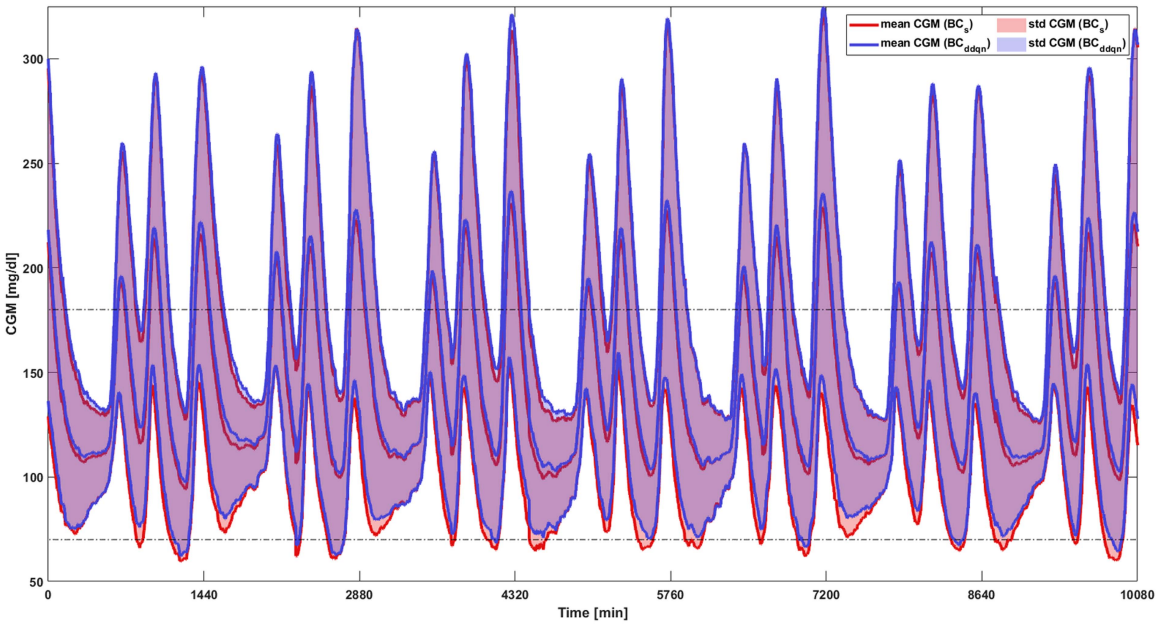

Fig. 4. CGM mean and standard deviation intervals resulting from the virtual subjects belonging to the test set are reported for a representative one-week-long simulation. CGM values related to the standard insulin dosing and DDQN bolus calculator are shown in red and blue respectively. Dashed lines indicate the normoglycemic range.

TABLE III
VALUES RELATED TO MEDIAN AND INTERQUARTILE RANGES OF TBR [%], TIR [%], TAR [%], N_{hypo} AND N_{hyper} ARE REPORTED FOR BOTH THE STANDARD AND DDQN BOLUS CALCULATORS

	TBR [%]	TIR [%]	TAR[%]	N_{hypo}	N_{hyper}
BC_s	8.78 (2.90, 12.44)	68.35 (64.87, 72.65)	22.24 (15.89, 30.89)	1 (0, 1)	2 (2, 3)
BC_{ddqn}	4.17* (2.70, 8.17)	70.08 (66.49, 76.55)	23.47 (18.25, 28.64)	0 (0, 1)	2 (2, 3)

*Statistically significant compared to BC_s with $p < 5\%$

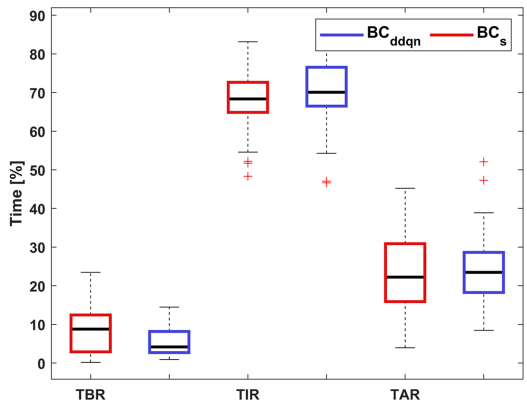

Fig. 5. Distributions of TBR, TIR and TAR resulting from the testing phase are reported in blue for the DDQN bolus calculator, while in red for the standard bolus calculator.

negative shift of the standard deviation, in most hypoglycemic events, the nadir is considerably reduced when BC_{ddqn} is used, while the increase in terms of maximum hyperglycemic value introduced by the DDQN algorithm is almost negligible.

The aforementioned remarks are consistent with the results obtained by analyzing the metrics related to the time in different glycemic ranges, as reported in Table III. Indeed, the advantage brought in terms of hypoglycemia reduction is reflected in a significantly lower distribution of TBR for BC_{ddqn} compared to BC_s, with a median value of 4.17% and 8.78% respectively. This was achieved with a small negative impact on hyperglycemia, as can be seen from the TAR distributions in Fig. 5, which reports a median value of 23.47% and 22.24% for the DDQN and standard therapy respectively, not showing a statistical significance between the two distributions. In general, the benefit introduced by the proposed algorithm is positive, since both the interquartile and the median value related to the TIR are improved. Moreover, 5 shows the TBR, TIR and TAR distributions related to both BC_{ddqn} and BC_s, together with the presence of only two outlier values for TIR and TAR data.

The positive impact given by the DDQN method was also confirmed when observing the median number of adverse events per day, shown in Table III. Indeed, median N_{hypo} is reduced from 1 event per day to zero, while N_{hyper} remains unchanged for both methods.

To ensure that the moderate increase in TAR is not influencing negatively the overall glycemic control, the risk metrics described in Section V-A were analysed and reported in Fig. 6. As expected, the LBGI is significantly reduced, while HBGI showed a slight increase in the median value. However, the BGRI, which summarizes the two aforementioned metrics, showed an improvement, by decreasing the median value from 8.2 of the benchmark to 7.3 of the BC_{ddqn}. Also, the average amount of daily insulin injected for each subject remained stable for both the methods (11 U for the standard bolus calculator and 10.3 U for the proposed method), suggesting that the significant

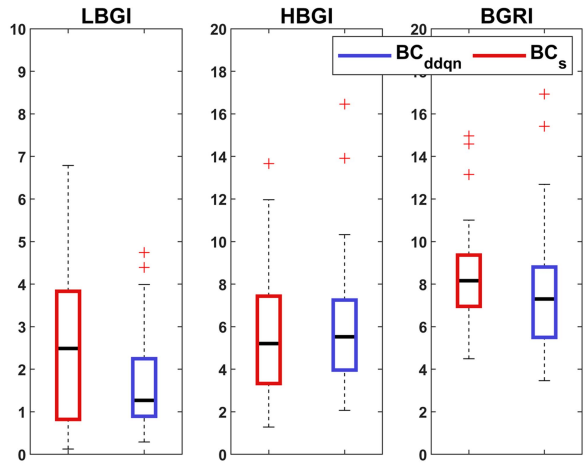

Fig. 6. Distributions of LBGI, HBGI, and BGRI resulting from the testing phase are reported in blue for the DDQN bolus calculator, while in red for the standard bolus calculator.

improvement in terms of metrics related to hypoglycemia is not simply due to a decrease in insulin dosage, but to a more effective redistribution of such hormone.

The reported results pointed out that, in general, the use of BC_{ddqn} could provide a beneficial impact on glycemic control, by considerably reducing not only the occurrence but also the duration of hypoglycemia, without significantly affecting hyperglycemia.

VI. Discussion and Conclusion

In this article, we proposed a mealtime insulin bolus calculator based on a DDQN algorithm, which leverages a high level of personalization and adaptation through a two-step learning framework combined with a clustering procedure. This method allowed us to fine-tune the generalized model related to the subpopulation subject which shows more similarity to the patient. Such individualization of the insulin dosing was implemented to deal with the multiple variability sources introduced by the updated and highly realistic version of the UVA/Padova T1D Simulator used in this work, as described in Section III.

The presented bolus calculator was tested within the simulated environment for a 60 days simulation, and compared to the state-of-art method for insulin dose calculation, i.e., the standard bolus calculator in (1), by extracting different metrics widely used by the diabetes research community to assess the quality of glycemic control. Despite the challenging scenario provided by the employed simulated environment, results in terms of time spent within the different glycemic ranges, the number of adverse events and glycemic risk metrics were encouraging, showing the ability of the proposed method to completely avoid, in some cases, or mitigate the magnitude of postprandial hypoglycemic events, while improving the time in range from 68.35% to 70.08% and without significantly impacting on the time in hyperglycemia compared to the benchmark.

The obtained performances indicated the potential efficacy of the algorithm in adjusting the standard dosage based on the mealtime state of the subject. Moreover, the average amount of daily insulin delivered to the subject remained stable both for the standard and the proposed method, pointing out that the reduction of hypoglycemia is not due to a simple decrease of the mealtime insulin amount, but rather the algorithm is redistributing the daily amount of insulin more effectively.

Limitations of the work include the need for a relatively long period to personalize the population model, being the meal an infrequent event within a day. However, two main points need to be considered. Being T1D an autoimmune disease characterized by a life-long therapy, a significant availability of data related to its monitoring and treatment (performed through CGM sensors and insulin delivery devices) is present. Consequently, many subjects already have access to a great amount of data related to their past therapy, which could be leveraged for the personalization phase on this algorithm. Moreover, for those individual not having access to past data, many datasets are also available in the literature, which could be used to apply transfer learning or meta-learning [41], [42]. Indeed, the availability of such datasets, could allow us to employ simulation tools, such as the one developed by Cappon et al. [36] or, which identify a model on a patient's glucose trace and to replay the scenario by changing the inputs of the model (e.g, insulin bolus), thus evaluating the effectiveness of a specific therapy. On the other hand, the standard therapy followed by people living with T1D is tuned by the clinician on a visit to visit bases, and is characterized by a constant learning process to understand how the insulin and the other physiological factors affects the body. For this reason, in such application, the requirement of a prolonged time is normal. Moreover, being this study a proof-of-concept, we focused only on the comparison with the standard formula, as our final aim is proposing a method which provides a more personalized therapy, tailored to the specific subject. For this reason, future developments of this study will involve a comprehensive comparison with the main literature methods aimed at personalizing this dosage. Hence, future works will involve an assessment in a clinical setting together with the exploration of the aforementioned methodology to speed up the personalization phase. Moreover, further developments of the method may include the exploration of a simulated scenario which also takes into account the insulin sensitivity variability, a challenging factor impacting T1D management, together with the assessment of the algorithm within a simulated environment which considers also confounding factors such as rescue carbohydrates or corrective insulin boluses. In conclusion, the application of a DDQ-learning algorithm combined with an effective model personalization procedure allowed us to achieve promising results within the updated version of the FDA-accepted UVA/Padova simulated environment when applied for prandial insulin dose adjustment.

References

[1] M. Mobasseri, M. Shirmohammadi, T. Amiri, N. Vahed, H. H. Fard, and M. Ghojazadeh, "Prevalence and incidence of type 1 diabetes in the world: A systematic review and meta-analysis," *Health Promot. Perspectives*, vol. 10, no. 2, pp. 98–115, 2020.

[2] R. I. Holt et al., "The management of type 1 diabetes in adults. A consensus report by the American Diabetes Association (ADA) and the European Association for the Study of Diabetes (EASD)," *Diabetes Care*, vol. 44, no. 11, pp. 2589–2625, 2021.

[3] S. Schmidt and K. Nørgaard, "Bolus calculators," *J. Diabetes Sci. Technol.*, vol. 8, no. 5, pp. 1035–1041, 2014.

[4] P. C. Davidson, H. R. Hebblewhite, R. D. Steed, and B. W. Bode, "Analysis of guidelines for basal-bolus insulin dosing: Basal insulin, correction factor, and carbohydrate-to-insulin ratio," *Endocr. Pract.*, vol. 14, no. 9, pp. 1095–1101, 2008.

[5] T. M. Gross, D. Kayne, A. King, C. Rother, and S. Juth, "A bolus calculator is an effective means of controlling postprandial Glycemia in patients on insulin pump therapy," *Diabetes Technol. Therapeutics*, vol. 5, no. 3, pp. 365–369, 2003.

[6] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli, "The UVA/PADOVA type 1 diabetes simulator: New features," *J. Diabetes Sci. Technol.*, vol. 8, no. 1, pp. 26–34, 2014.

[7] G. Noaro, G. Cappon, G. Sparacino, F. Boscari, D. Bruttomesso, and A. Facchinetti, "Methods for insulin bolus adjustment based on the continuous glucose monitoring trend arrows in type 1 diabetes: Performance and safety assessment in an in silico clinical trial," *J. Diabetes Sci. Technol.*, vol. 17, no. 1, pp. 107–116, 2023.

[8] G. Cappon, F. Marturano, M. Vettoretti, A. Facchinetti, and G. Sparacino, "In silico assessment of literature insulin bolus calculation methods accounting for glucose rate of change," *J. Diabetes Sci. Technol.*, vol. 13, no. 1, pp. 103–110, 2019.

[9] D. Bruttomesso et al., "A 'slide rule' to adjust insulin dose using trend arrows in adults with type 1 diabetes: Test in silico and in real life," *Diabetes Ther.*, vol. 12, no. 5, pp. 1313–1324, 2021.

[10] P. E. Cryer, S. N. Davis, and H. Shamoon, "Hypoglycemia in diabetes," *Diabetes Care*, vol. 26, no. 6, pp. 1902–1912, 2003.

[11] G. Shafiee, M. Mohajeri-Tehrani, M. Pajouhi, and B. Larijani, "The importance of hypoglycemia in diabetic patients," *J. Diabetes Metabolic Disord.*, vol. 11, no. 1, pp. 1–7, 2012.

[12] D. Daneman, "Type 1 diabetes," *Lancet*, vol. 367, no. 9513, pp. 847–858, 2006.

[13] G. Cappon, M. Vettoretti, G. Sparacino, and A. Facchinetti, "Continuous glucose monitoring sensors for diabetes management: A review of technologies and applications," *Diabetes Metab. J.*, vol. 43, no. 4, pp. 383–397, 2019.

[14] D. DeSalvo and B. Buckingham, "Continuous glucose monitoring: Current use and future directions," *Curr. Diabetes Rep.*, vol. 13, no. 5, pp. 657–662, 2013.

[15] M. Vettoretti, G. Cappon, G. Acciaroli, A. Facchinetti, and G. Sparacino, "Continuous glucose monitoring: Current use in diabetes management and possible future applications," *J. Diabetes Sci. Technol.*, vol. 12, no. 5, pp. 1064–1071, 2018.

[16] P. Pozzilli, T. Battelino, T. Danne, R. Hovorka, P. Jarosz-Chobot, and E. Renard, "Continuous subcutaneous insulin infusion in diabetes: Patient populations, safety, efficacy, and pharmacoeconomics," *Diabetes/Metab. Res. Rev.*, vol. 32, no. 1, pp. 21–39, 2016.

[17] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Deep learning for diabetes: A systematic review," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 7, pp. 2744–2757, Jul. 2021.

[18] P. Herrero, P. Pesl, M. Reddy, N. Oliver, P. Georgiou, and C. Toumazou, "Advanced insulin bolus advisor based on run-to-run control and case-based reasoning," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 3, pp. 1087–1096, May 2015.

[19] C. Fabris, B. Ozaslan, and M. D. Breton, "Continuous glucose monitors and activity trackers to inform insulin dosing in type 1 diabetes: The university of virginia contribution," *Sensors*, vol. 19, no. 24, 2019, Art. no. 5386.

[20] G. Noaro, G. Cappon, M. Vettoretti, G. Sparacino, S. Del Favero, and A. Facchinetti, "Machine-learning based model to improve insulin bolus calculation in type 1 diabetes therapy," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 247–255, Jan. 2021.

[21] G. Cappon, M. Vettoretti, F. Marturano, A. Facchinetti, and G. Sparacino, "A neural-network-based approach to personalize insulin bolus calculation using continuous glucose monitoring," *J. Diabetes Sci. Technol.*, vol. 12, no. 2, pp. 265–272, 2018.

[22] T. Zhu, K. Li, L. Kuang, P. Herrero, and P. Georgiou, "An insulin bolus advisor for type 1 diabetes using deep reinforcement learning," *Sensors*, vol. 20, no. 18, 2020, Art. no. 5058.

[23] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[25] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[26] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, vol. 30, pp. 2094–2100.

[27] B. Buckingham, D. Xing, and S. Weinzimer, "Diabetes research in children network (direcnet) study group. use of the direcnet applied treatment algorithm (data) for diabetes management with a real-time continuous glucose monitor (the freestyle navigator)," *Pediatr. Diabetes*, vol. 9, no. 2, pp. 142–147, 2008.

[28] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 4, pp. 1223–1232, Apr. 2021.

[29] R. Visentin et al., "The UVA/PADOVA type 1 diabetes simulator goes from single meal to single day," *J. Diabetes Sci. Technol.*, vol. 12, no. 2, pp. 273–281, 2018.

[30] A. Facchinetti, S. Del Favero, G. Sparacino, and C. Cobelli, "Model of glucose sensor error components: Identification and assessment for new dexcom G4 generation devices," *Med. Biol. Eng. Comput.*, vol. 53, no. 12, pp. 1259–1269, 2015.

[31] M. Vettoretti, C. Battocchio, G. Sparacino, and A. Facchinetti, "Development of an error model for a factory-calibrated continuous glucose monitoring sensor with 10-day lifetime," *Sensors*, vol. 19, no. 23, 2019, Art. no. 5320.

[32] M. Vettoretti, A. Facchinetti, G. Sparacino, and C. Cobelli, "Type-1 diabetes patient decision simulator for in silico testing safety and effectiveness of insulin treatments," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 6, pp. 1281–1290, Jun. 2018.

[33] A. Brazeau et al., "Carbohydrate counting accuracy and blood glucose variability in adults with type 1 diabetes," *Diabetes Res. Clin. Pract.*, vol. 99, no. 1, pp. 19–23, 2013.

[34] C. Roversi, M. Vettoretti, S. Del Favero, A. Facchinetti, G. Sparacino, and H.-R. Consortium, "Modeling carbohydrate counting error in type 1 diabetes management," *Diabetes Technol. Therapeutics*, vol. 22, no. 10, pp. 749–759, 2020.

[35] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 2009.

[36] G. Cappon, A. Facchinetti, G. Sparacino, and S. Del Favero, "A Bayesian framework to identify type 1 diabetes physiological models using easily accessible patient data," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Jul. 2019, pp. 6914–6917, doi: 10.1109/EMBC.2019.8856846.

[37] D. M. Maahs et al., "Outcome measures for artificial pancreas clinical trials: A consensus report," *Diabetes Care*, vol. 39, no. 7, pp. 1175–1179, 2016.

[38] T. Danne et al., "International consensus on use of continuous glucose monitoring," *Diabetes Care*, vol. 40, no. 12, pp. 1631–1640, 2017.

[39] B. P. Kovatchev, D. J. Cox, L. A. Gonder-Frederick, and W. Clarke, "Symmetrization of the blood glucose measurement scale and its applications," *Diabetes Care*, vol. 20, no. 11, pp. 1655–1658, 1997.

[40] W. J. Conover, *Practical Nonparametric Statistics*. Hoboken, NJ, USA: Wiley, 1999, vol. 350.

[41] C. Marling and R. Bunescu, "The OhioT1DM dataset for blood glucose level prediction: Update 2020," in *CEUR Workshop Proc.*, 2020, vol. 2675, pp. 71–74.

[42] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Personalized blood glucose prediction for type 1 diabetes using evidential deep learning and meta-learning," *IEEE Trans. Biomed. Eng.*, vol. 70, no. 1, pp. 193–204, Jan. 2023.

Open Access funding provided by 'Università degli Studi di Padova' within the CRUI CARE Agreement