Model-Based Reinforcement Learning for Type 1 Diabetes Blood Glucose Control

Taku Yamagata, Aisling O'Kane, Amid Ayobi, Dmitri Katz, Katarzyna Stawarz, Paul Marshall, Peter Flach and Raúl Santos-Rodríguez

Abstract In this paper we investigate the use of model-based reinforcement learning to assist people with Type 1 Diabetes with insulin dose decisions. The proposed architecture consists of multiple Echo State Networks to predict blood glucose levels combined with Model Predictive Controller for planning. Echo State Network is a version of recurrent neural networks which allows us to learn long term dependencies in the input of time series data in an online manner. Additionally, we address the quantification of uncertainty for a more robust control. Here, we used ensembles of Echo State Networks to capture model (epistemic) uncertainty. We evaluated the approach with the FDA-approved UVa/Padova Type 1 Diabetes simulator and compared the results against baseline algorithms such as Basal-Bolus controller and Deep Q-learning. The results suggest that the model-based reinforcement learning algorithm can perform equally or better than the baseline algorithms for the majority of virtual Type 1 Diabetes person profiles tested.

Taku Yamagata
University of Bristol, Bristol BS8 1UB UK e-mail: taku.yamagata@bristol.ac.uk

Amid Ayobi
University of Bristol, Bristol BS1 5DD UK e-mail: amid.ayobi@bristol.ac.uk

Aisling O'Kane
University of Bristol, Bristol BS8 1UB UK e-mail: a.okane@bristol.ac.uk

Dmitri Katz
The Open University, Milton Keynes MK7 6AA UK e-mail: dmitri.katz@open.ac.uk

Katarzyna Stawarz
Cardiff University, Cardiff CF24 3AA UK e-mail: StawarzK@cardiff.ac.uk

Paul Marshall
University of Bristol, Bristol BS8 1UB UK e-mail: p.marshall@bristol.ac.uk

Peter Flach
University of Bristol, Bristol BS8 1UB UK e-mail: Peter.Flach@bristol.ac.uk

Raúl Santos-Rodríguez
University of Bristol, Bristol BS8 1UB UK e-mail: enrsr@bristol.ac.uk

1 Introduction

Type 1 Diabetes is a chronic condition that is characterized by the lack of insulin secretion and resulting in uncontrolled blood glucose level increase [1, 9]. High blood glucose levels for extended periods of time can result in permanent damage to the eyes, nerves, kidneys and blood vessels, while low blood glucose levels can lead to death [19, 20, 23]. To manage blood glucose level, people on multi-dose injection (MDI) therapy usually take two types of insulin injections: basal and bolus. The basal is long-acting insulin, which provides a constant supply of insulin over 24-48 hours, helping maintain resting blood glucose levels. The bolus is fast-acting insulin which helps to suppress the peak of the blood glucose levels caused by meals or to counteract hyperglycemia [23]. People with diabetes must make constant decisions of the timing and amount of these insulin injections, which is often challenging as insulin requirements for meals can change depending upon many factors such as exercise, sleep, or stress. The idiosyncratic nature of the condition means that triggers, symptoms and even treatments are often quite individual [18, 24, 25, 26, 29], which creates challenges to developing diabetes self-management technologies.

In this paper we consider the benefits of using model-based reinforcement learning (MBRL) to assist decisions about bolus insulin injections. The goal of reinforcement learning (RL) is to learn sequences of actions in an unknown environment [30]. The learner (Agent) interacts with the environment, observes its consequences, and receives a reward (or a cost) signal, which is a numerical number assessing current the situation. The agent decides a sequence of actions to maximize the reward (or minimize the cost) as shown in Fig.1. RL is well-suited to this task because it can learn the model in an online manner with minimal assumptions about the underlying process of the blood glucose behaviour and hence can adapt to different individuals or changes over time. MBRL is particularly well suited to this objective because it is more sample-efficient than alternative RL approaches (model-free reinforcement learning (MFRL)) and also allows us to generate predictions for consequences of counterfactual actions that can be used as *explanations* of the suggestion. In our MBRL setting, we also can estimate the confidence level of the predictions by using the prediction uncertainty. It is very important to show the explanation for the suggestion together with its confidence level so that the person that receives the suggestion can make a decision whether they would follow the recommended course of action.

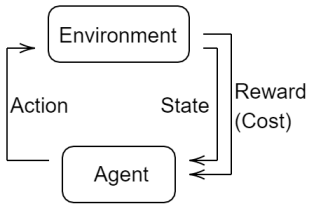

Fig. 1 Reinforcement learning framework overview.

As a first step towards realising such a recommender system, we investigated how well MBRL can learn the insulin injection decision and compared it with both a typical MFRL algorithm (deep Q-Learning (DQN)) and an algorithm that mimics human decision-making (Basal-Bolus controller (BBController)). We used an FDA-approved Type 1 Diabetes computer simulator and let the algorithms decide the insulin injections and evaluated its blood glucose level behaviours.

Our MBRL approach builds upon previous work on Echo State Networks (ESNs) [14, 13], the ensembles of models for MBRL [5] and model predictive controller (MPC) for artificial pancreas [4, 3]. However we believe this is the first attempt to combine these algorithms for the Type 1 Diabetes blood glucose level control task, and evaluate its performance against non-MBRL algorithms.

This paper is organized as follows. Section 2 introduces related work regarding the blood glucose control task. Section 3 describes our MBRL method. Section 4 presents our evaluation method, benchmark algorithms and the evaluation results. Finally, Section 5 concludes with a summary and possible future work.

2 Related Work

Several attempts have been made for a closed-loop artificial pancreas, especially in the control system society using MPC [3], proportional-integral-derivative control [28] and fuzzy logic [2].

However, there are relatively few studies on the blood glucose levels control task using RL approaches. Most of the early works employ compartmental blood glucose and insulin models to infer some of insulin/glucose related internal states of human body, and then learn its insulin injection policy with relatively simple MFRL algorithms such as Q-Learning [21, 22] or Actor-Critic [8, 7]. Fox *et al.* employed more recent RL techniques [12], such as deep neural networks for the Q-Learning algorithm – arguably the most common MFRL algorithm. They showed that although the agent was not given any prior knowledge of the blood glucose/insulin relations, it learns its insulin injection policy and achieves performance comparable with existing algorithms.

In the field of model-based system control several approaches exist – we refer the reader to [3] and the references therein. The closest to our work is [4], where the authors use a linear compartmental model for predicting the mean and variance of the future blood glucose levels. It exploits MPC for planning by taking into account the variance of the blood glucose level prediction. The main differences from our work are: (1) they employ a linear compartmental model which has a small number of parameters and hence easier to learn, whereas we use more generic recurrent neural networks, which have greater flexibility to adapt to any personal blood glucose level behaviour; (2) their model parameters are learnt off-line, whereas ours are adjusted online; and (3) the handling of uncertainty – we measure the model's uncertainty while they measure the uncertainty involved in meal events.

3 Methods

In order to apply RL algorithms to this problem, we formulate the task as Markov Decision Process (MDP), which has four tuples (S, A, p, c) where S is a set of states, A is a set of actions, p is the state transition probabilities and c is a cost function. Essentially the blood glucose control task is a Partially Observable MDP, however we see it as an MDP by defining state S as all history of insulin doses and carbohydrate intakes.

More precisely, the overall pipeline makes use of ESNs to store the history in its hidden states, shown in Section3.2. The corresponding actions A are the dosages of bolus insulin. We exploit the risk function introduced in [16] as our cost function c, described in Section 3.1. While we use the model-based reinforcement learning (MBRL) algorithm with ESNs for the prediction of blood glucose levels, MPC generates the insulin dose suggestions from the blood glucose level predictions (Section 3.4) and their uncertainty estimations (Section 3.3). .

3.1 Cost function

For our task, it is natural to use as cost function a measure of risk associated with the given blood glucose level. However it is not straightforward to define such a measure, as it presents different scales of risks between higher than normal blood glucose levels (hyperglycemia) and lower than normal blood glucose levels (hypoglycemia). Kovatchev *et al.* proposed the following expression to symmetrize the risks of hyper and hypoglycemia [16]. This blood glucose risk function f_r is defined as in Eq. 1. The blood glucose level transition from 180 to 250mg/dl would appear threefold larger than a transition from 70 to 50mg/dl, whereas these are similar in terms of the risk function variations.

$$f_r(BGL) = 15.09 \cdot \left(log(BGL)^{1.084} - 5.381 \right)^2 \tag{1}$$

where BGL is the blood glucose level in mg/dl. Fig. 2 shows the mapping between blood glucose level (x-axis) to the risk function (y-axis). We used the risk function value as the cost function, hence our RL agent searches a policy minimising the total risk values over an episode.

3.2 Echo State Networks

ESNs were proposed as an alternative structure of standard recurrent neural networks in machine learning [14]. They are also called liquid state machine in computational neuroscience [13]. ESNs take an input sequence $\mathbf{u} = (\mathbf{u}(1), \mathbf{u}(2), ..., \mathbf{u}(T))$ by recursively processing each symbol while maintaining its internal hidden state $\mathbf{x}$. At each

time step t, the ESN takes input $\mathbf{u(t)} \in \mathbb{R}^K$ and updates its hidden state $\mathbf{x}(t) \in \mathbb{R}^N$ by:

$$\tilde{\mathbf{x}}(t) = f(\mathbf{W}^{in} \cdot \mathbf{u}(t) + \mathbf{W} \cdot \mathbf{x}(t-1)) \tag{2}$$

$$\mathbf{x}(t) = (1-\alpha) \cdot \mathbf{x}(t-1) + \alpha \cdot \tilde{\mathbf{x}}(t), \tag{3}$$

where f is the internal unit activation function, which is $tanh$ in our model, $\mathbf{W}^{in} \in \mathbb{R}^{N \times K}$ is the input weight matrix, $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the internal connections weight matrix and $\alpha \in (0, 1]$ is the leakage rate, which controls the speed of the hidden states change hence controls the output smoothness.

The output at time step t, $\mathbf{y}(t) \in \mathbb{R}^L$ is obtained from the hidden states and the inputs by:

$$\mathbf{y}(t) = f^{out}\left(\mathbf{W}^{out} \cdot \left[\mathbf{x}(t)^T, \mathbf{u}(t)^T\right]^T\right), \tag{4}$$

where f^{out} is the output unit activation function (which is the identity function in our model as we are dealing with a regression task) and $\mathbf{W}^{out} \in \mathbb{R}^{L \times (N+K)}$ is the output weights matrix.

The matrices for updating the hidden states, $\mathbf{W}^{in}$ and $\mathbf{W}$, are randomly initialized and fixed (not updated during learning process), only the output weights matrix $\mathbf{W}^{out}$ is leaned to obtain the target output sequences. As it only learns the output weights, it doesn't require back propagation through the network nor time, hence it learns much faster than the normal recurrent neural networks. The downside of using ESN is that it requires much higher number of hidden states to achieve good performance, hence it required more computational power for inference.

To make ESNs work properly, the fixed weights must satisfy the so-called *echo state property*: the internal states $\mathbf{x}(t)$ should be uniquely defined only by the past inputs $\mathbf{u}(k)|_{k=\ldots t}$ [14]. The actual method to initialise the weights can be found in [17], which also gives useful guidance for using ESNs.

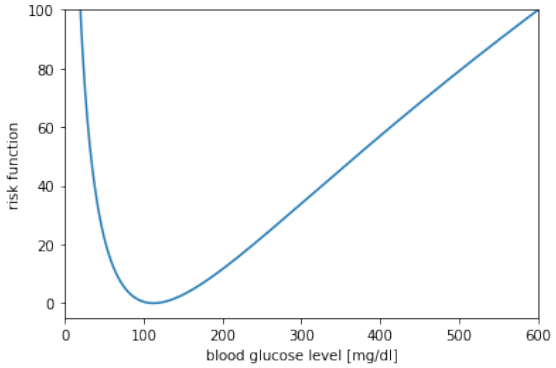

Fig. 2 Risk function proposed by Kovatchev *et al.* [16] The figure shows the relationship between blood glucose level [mg/dl] and its risk function value. The risk function value shows asymmetric shape – increase rapidly for low blood glucose levels compare to the high blood glucose levels, which is aligned with the clinical risk of low and high blood glucose levels.

ESNs for the blood glucose level prediction task

In our work, the ESN takes a sequence of bolus insulin injection and carbohydrate intakes as inputs, and predicts the blood glucose level.

To learn the ESN output weights we use the Mean Squared Error between predicted and observed blood glucose levels as loss function.

$$\mathcal{L}_d(\theta) = \frac{1}{T}\sum_{t=1}^{T}(\mu_\theta(t) - BGL(t))^2 \tag{5}$$

Here, $\mu_\theta(t)$ is the predicted blood glucose level by ESN at time step t, where θ is the optimization parameter (here it is $\mathbf{W}^{out}$) and $BGL(t)$ is observed blood glucose level. As it can be seen as a linear regression problem, the output weights are derived by solving the Normal equation [17].

To capture model (epistemic) uncertainty, it applies multiple instances of ESNs, and each of them has different input and internal connection weights. ESNs are well suited for the ensemble approach as it has fixed random internal weights which project the inputs sequence into different hidden states. So naturally they output different values where there is no training data, capturing higher epistemic uncertainty. In our evaluation, we employ five instances of ESNs, which is suggested by [5].

3.3 Uncertainty quantification

We employ multiple ESNs to capture the uncertainty in predicted blood glucose level. They produce multiple predictions of the blood glucose levels from the ESN models for each action sequence. To quantify the cost (risk) of uncertainty, we take the mean of the cost of the predicted blood glucose levels for each of action sequence $\frac{1}{MT}\sum_{t=n}^{n+T-1}\sum_{m=1}^{M} c(BGL_t^m)$, where $c(.)$ is a cost function, BGL_t^m is blood glucose levels prediction from ESN model m at time step t, and M and T are number of ESN models and number of time steps in the action sequence. We then select the action sequence which minimises this mean cost.

We encourage (optimistic or exploratory approach) or discourage (pessimistic or safe approach) taking risks by designing the cost function accordingly. Here we define a risk margin RM as the difference between the averaged cost function and cost of the averaged blood glucose level predictions.

$$RM = E[c(BGL)] - c(E[BGL]). \tag{6}$$

A positive (negative) risk margin means our metric $E[c(BGL)]$ discourages (encourages) taking risks. If we use a convex cost function as described in Section 3.1, RM is positive according to Jensen's inequality, hence it discourages risks.

3.4 Model Predictive Controller

Model predictive controller (MPC) is a planning method to facilitate control of systems with a long time delay and non-linear characteristics. The MPC uses a prediction model to estimate the consequences of a sequence of actions and repeats the process for many action sequences. Then it picks the sequence of actions that gives the best consequence and applies the first action of the sequence. In the next time step this process is repeated. This effectively means it re-plans the sequence of actions based on the latest state information from the environment, which makes the algorithm robust against any noise or prediction errors.

There are several algorithms to generate the sequence of actions to test – such as random shooting [27] and cross entropy method [10]. In our work, we use a fixed table for the sequence of actions to test. The table has six action sequences, each of which takes a different amount of bolus injection as its first action. The amount of bolus injection at the first action is {0, 5, 10, 20, 40, 80} times of the person's basal infusion rate. Following the approach of [12], the basal infusion rate is given for each virtual person's model, and we use it to scale the bolus injection. While our model generates suggestions for bolus injections, for the basal injections, it assumes the person is taking the given basal infusion rate. The action sequence length (time horizon) is set to 48 time steps, which is 4 hours long as each time step represents a five-minute period. Each action sequence has a bolus injection as the first action of the sequence. We believe this is sensible because the bolus injections is normally taken just after or before a meal and there is no meal announcement in our system at moment (the algorithm does not know the meal event until it happens). Therefore, the best time to take bolus injection would be immediately after detecting the meal event, which is the first action in the sequence. A proper meal announcement mechanism is left for future work.

4 Evaluation

We empirically evaluated how well the model-based reinforcement learning (MBRL) can learn insulin injection decisions and compared it with a typical model-free reinforcement learning (MFRL) algorithm and also with a non-RL algorithm designed to mimic human decision-making. In this paper, we did not compare the blood glucose level prediction accuracy with other prediction models. Instead, we focused on evaluating the performance of the agents. The overview of the evaluation system is shown in Fig. 3. We used an FDA-approved Type 1 Diabetes simulator, which takes meal and insulin injection information, then outputs a blood glucose level (BGL) as a continuous glucose monitor (CGM) reading at each time step. The algorithms (agents) receive the meal, insulin and blood glucose level information and decides the amount of insulin taking in the next time step. We simulated the algorithms together with the Type 1 Diabetes simulator, and evaluated how well the blood glucose levels are managed.

4.1 UVa/Padova Type 1 Diabetes simulator

The UVa/Padova Type 1 Diabetes Simulator [6] was the first computer model accepted by the FDA as a substitute for preclinical trials of certain insulin treatments, including closed-loop algorithms. The model takes carbohydrate intakes and insulin injection as inputs, simulates human body insulin/blood glucose behaviours and outputs the blood glucose level measurements. It has gastro-interstinal tract, glucose kinetics and insulin kinetics sub models. Each of these sub models is defined with differential equations with parameters to simulate different individuals. Our simulator is based on an open source implementation of the UVa/Padova Type 1 Diabetes simulator [15], which comes with different profiles for 30 virtual people with type 1 diabetes – ten each for children, adolescents and adults. Our experiments use nine virtual people, three of each age group.

4.2 Benchmark algorithms

We used two benchmark algorithms to compare the proposed approach against, one from RL algorithms (GRU-DQN) and the other one from non-RL approaches (BBController). These are described below.

GRU-DQN

Deep Q-Learning (DQN) is a common MFRL algorithm, which learns the action-value function $Q(s,a)$ – expected cumulative future rewards starting with state s and action a. It then uses the learned action value function to decide which action to take at time step t by $a_t = argmax_{a\in\mathcal{A}}Q(s_t,a)$. In our work, the agent observes the blood glucose levels from a CGM, carbohydrate intakes and insulin injections, and infers the action value function. It is a partially observable model so we used gated recurrent units (GRU) to infer the hidden states and approximate the action value function. GRU-DQN was successfully applied to this problem before [12] so we followed their same set up which involves two GRU recurrent layers of 128 hidden states and followed by a fully connected output layer size of 128. However, our our

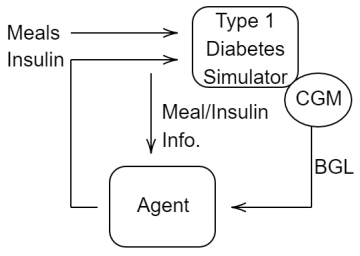

Fig. 3 Evaluation system top level diagram.

Table 1 Parameters for meal event generator.

Meal type	Prob.	Time [hours]				Carbs. [g]	
		lower bound	upper bound	mean	std.	mean	std.
Breakfast	0.95	5	9	7	1	45	10
Snack#1	0.3	9	10	9.5	0.5	10	5
Lunch	0.95	10	14	12	1	70	10
Snack#2	0.3	14	16	15	0.5	10	5
Dinner	0.95	16	20	18	1	80	10
Snack#3	0.3	20	23	21.5	0.5	10	5

states (the input of GRU-DQN) include carbohydrate information, whereas [12] does not. We include it here to make our comparison fair against the MBRL algorithm, which has acess to the carbohydrate information.

BBController

Basal-Bolus Controller mimics how an individual with Type 1 Diabetes controls their blood glucose levels. The UVa/Padova simulator comes with the necessary parameters for this algorithm for each of the virtual people with Type 1 Diabetes models, such as basal insulin rates bas, a correction factor CF and a carbohydrate ratio CR. The simulator decides the amount of insulin injection by $bas + (c_t > 0) \cdot (c_t/CR + (b_t > 150) \cdot (b_t - b_{tgt})/CF)$, where c_t is carbohydrate intake at time step t, b_t is the blood glucose measurements, b_{tgt} is a target blood glucose level. The last term is only applied when the blood glucose measurement exceeds 150 mg/dl. We use the implemented model that comes with the Type 1 Diabetes simulator [15].

4.3 Simulation Conditions

Each episode lasts 24 hours, starting at 6am and finishing at 6am the next day. Three meals and three snack events are simulated with some randomness in terms of amount, timing and also whether they take the meal/snack. The timing follows a truncated normal distribution and the amount is normally distributed. The meal parameters are shown in Table 1. The agent receives information from the environment such as the meal (carbohydrate), insulin and blood glucose levels, and decides the insulin dose for the next time step. Each time step is set to five minutes in length. In this evaluation, the person does not take food to compensate for low blood glucose levels (the meal event always follows a pre-defined order as described above). While this is not realistic, it is a good way to measure how well the algorithm works because ultimately we would like to develop an algorithm that does not require any corrections from the user. The episode is terminated if the blood glucose level goes

Table 2 % of number of completed episodes without termination due to extreme blood glucose level value

Virtual Person Profile	BBContoller	GRU-DQN	MBRL
child#001	30.0	3.3	**100**
child#002	**90**	23.3	53.3
child#003	**66.7**	43.3	30.0
adolescent#001	**100**	**100**	**100**
adolescent#002	**66.7**	56.7	0.0
adolescent#003	90	20	**100**
adult#001	**100**	70.0	96.7
adult#002	**100**	**100**	**100**
adult#003	96.7	16.7	**100**

below 20 mg/dl or beyond 600 mg/dl, as these limit are extreme and they are outside of the possible blood glucose level range considered by [16].

4.4 Results

We train MBRL for 200 episodes and GRU-DQN for 1000 episodes, then use the last 30 episodes to measure the percentage of episodes completed without termination due to extreme blood glucose levels. For BBController, we just run 30 episodes to measure, as it has pre-optimized model parameters and no training is required.

The results are given in Table 2. MBRL gives better results than GRU-DQN and comparable with BBController. MBRL struggles with child#002, #003 and adolescent#002. By looking into these cases, we found that MBRL fails due to the MPC time horizon not being long enough. The MPC time horizon is set to 4 hours, hence the agent could not foresee a possible hypoglycemia event in the early morning after the person takes an evening meal. The agent suggests too much insulin, and it causes hypoglycemia in the early morning. This can be fixed by increasing the MPC time horizon, but requires some additional consideration as it might lead to inappropriate suggestions during the day.

Table 3 shows the percentage of time spent in a target blood glucose level range (70-180 mg/dl.) These are measured in the last 10 of the completed episodes(i.e., not terminated). Here MBRL gives the best overall results compared to the other agents. Note that no data is available for adolescent#002, as it fails to get any non-terminated episode (due to the reason described above).

We also evaluated the effect of the uncertainty estimation by comparing the results from MBRL with/without it. For MBRL without uncertainty, we take an average over multiple ESNs predictions to come up with a single blood glucose prediction, and then we calculate its cost. Whereas MBRL with uncertainty computes the cost of the all predictions, then takes average of the costs as described in Section 3.3.

Table 3 % of time spent in the target blood glucose level range (70 - 180 mg/dl)

Virtual Person Profile	BBContoller	GRU-DQN	MBRL
child#001	44.0	28.3	**59.6**
child#002	42.6	38.2	**55.3**
child#003	40.7	36.0	**45.1**
adolescent#001	85.8	81.4	**100.0**
adolescent#002	**49.0**	39.8	n/a
adolescent#003	46.7	42.4	**66.1**
adult#001	**60.1**	50.3	56.8
adult#002	**73.3**	66.9	**73.3**
adult#003	58.7	46.9	**68.8**

Figure 4 shows the learning curves for these two MBRL algorithms with adult#001. The upper plot shows the episode period, which goes up to 24 hours if there is no termination, and the bottom plot shows % of time spent in the target blood glucose range. From the upper plot, the algorithm with uncertainty achieves "no episode termination" (24 hours episode) much earlier than the one without estimating uncertainty. At an early stage of the learning process, the prediction model is not very accurate, so it is much better by taking into account its uncertainty. For the later stages, the predictions become more accurate, hence it shows similar performance in both cases. Table 4 shows asymptotic results of the percentage of time spent in the target blood glucose range, indicating that both have similar asymptotic performances.

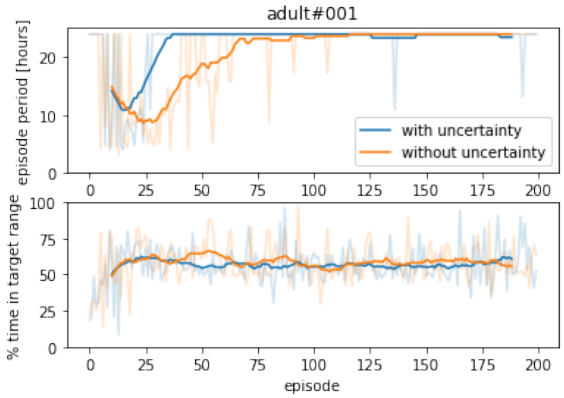

Fig. 4 Comparison between MBRL with uncertainty and without uncertainty models. The upper plot shows the learning curve for simulated period for each episode, which goes up to 24 hours if the blood glucose level is controlled well. The lower plot shows % of time spent in the target blood glucose range (70-180mg/dl)

Table 4 % of time spent in the target blood glucose range (70 - 180 mg/dl)

Virtual Person Profile	MBRL (with uncertainty)	MBRL (without uncertainty)
child#001	59.6	57.5
adolescent#001	100.0	95.9
adult#001	56.8	56.7

5 Conclusions and Future Work

We investigated the use of MBRL to assist Type 1 Diabetes decision-making by evaluating MBRL with the FDA-approved UVa/Padova simulator. We compared the results with two baseline algorithms, GRU-DQN and BBController. The results suggest that the MBRL approach works better than the GRU-DQN algorithm and similar or slightly better than the BBController. Also, our results show that taking into account the model uncertainty improves its performance in the early stages of learning.

There are several avenues for future work. At the present stage we only tested our algorithms with the UVa/Padova Type 1 Diabetes simulator, which is good for single meal scenarios but not for multiple meals [6]. This is primarily because the model has fixed parameters for each person and does not simulate meal-by-meal nor day-by-day parameter drifting. In addition, our current learning method must be extended to adapt to parameter drifts. A possible approach for such an extension would be to introduce meta-learning [11].

Another area for further work relates to meal information. We assumed all meal events are correctly given by the person when the event is happening; however, this may not be very realistic as it is a considerable burden for a person to put every single meal event into the algorithm. It is also hard to know the exact carbohydrate count of each meal. Some researchers therefore structure the blood glucose predictor without having a meal input. Another alternative would be to have a model to back-predict a meal event from the observed blood glucose levels. We think it is possible to learn the meal event in conjunction with the blood glucose level prediction model with occasional human inputs.

Acknowledgements This project is funded by the Innovate UK Digital Catalyst Award – Digital Health and is in partnership with Quin Technology.

References

1. Alberti, K.G.M.M., Zimmet, P.Z.: Definition, diagnosis and classification of diabetes mellitus and its complications. part 1: diagnosis and classification of diabetes mellitus. provisional report of a who consultation. Diabetic medicine **15**(7), 539–553 (1998)

2. Atlas, E., Nimri, R., Miller, S., Grunberg, E.A., Phillip, M.: MD-logic artificial pancreas system: A pilot study in adults with type 1 diabetes. Diabetes Care (2010). DOI 10.2337/dc09-1830
3. Bequette, B.W.: Algorithms for a closed-loop artificial pancreas: The case for model predictive control. Journal of Diabetes Science and Technology **7**(6), 1632–1643 (2013). DOI 10.1177/193229681300700624
4. Cameron, F., Bequette, B.W., Wilson, D.M., Buckingham, B.A., Lee, H., Niemeyer, G.: A closed-loop artificial pancreas based on risk management. Journal of Diabetes Science and Technology **5**(2), 368–379 (2011). DOI 10.1177/193229681100500226
5. Chua, K., Calandra, R., McAllister, R., Levine, S.: Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In: Advances in Neural Information Processing Systems, vol. 2018-Decem (2018)
6. Dalla Man, C., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., Cobelli, C.: The UVA/PADOVA type 1 diabetes simulator: New features. Journal of Diabetes Science and Technology **8**(1), 26–34 (2014). DOI 10.1177/1932296813514502
7. Daskalaki, E., Diem, P., Mougiakakou, S.G.: An Actor-Critic based controller for glucose regulation in type 1 diabetes. Computer Methods and Programs in Biomedicine **109**(2), 116–125 (2013). DOI 10.1016/j.cmpb.2012.03.002. URL `http://dx.doi.org/10.1016/j.cmpb.2012.03.002`
8. Daskalaki, E., Diem, P., Mougiakakou, S.G.: Personalized tuning of a reinforcement learning control algorithm for glucose regulation. In: 2013 35th Annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp. 3487–3490. IEEE (2013)
9. Davis, A.K., DuBose, S.N., Haller, M.J., Miller, K.M., DiMeglio, L.A., Bethin, K.E., Goland, R.S., Greenberg, E.M., Liljenquist, D.R., Ahmann, A.J., et al.: Prevalence of detectable c-peptide according to age at diagnosis and duration of type 1 diabetes. Diabetes care **38**(3), 476–481 (2015)
10. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Annals of Operations Research (2005). DOI 10.1007/s10479-005-5724-z
11. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. 34th International Conference on Machine Learning, ICML 2017 **3**, 1856–1868 (2017)
12. Fox, I., Wiens, J.: Reinforcement Learning for Blood Glucose Control: Challenges and Opportunities (2019). URL `https://openreview.net/forum?id=ryeN5aEYDH`
13. Gürbilek, N.: Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. Journal of Chemical Information and Modeling **53**(9), 1689–1699 (2013). DOI 10.1017/CBO9781107415324.004
14. Jaeger, H.: The " echo state " approach to analysing and training recurrent neural networks – with an Erratum note 1. GMD Report (148), 1–47 (2010). DOI citeulike-article-id:9635932
15. Jinyu Xie.: Simglucose v0.2.1 (2018). URL `https://github.com/jxx123/simglucose`
16. Kovatchev, B.P., Cox, D.J., Gonder-Frederick, L.A., Clarke, W.: Symmetrization of the blood glucose measurement scale and its applications. Diabetes Care **20**(11), 1655–1658 (1997). DOI 10.2337/diacare.20.11.1655
17. Lukoševičius, M.: A practical guide to applying echo state networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **7700 LECTU**, 659–686 (2012). DOI 10.1007/978-3-642-35289-8-36
18. Mianowska, B., Fendler, W., Szadkowska, A., Baranowska, A., Grzelak-Agaciak, E., Sadon, J., Keenan, H., Mlynarski, W.: Hba 1c levels in schoolchildren with type 1 diabetes are seasonally variable and dependent on weather conditions. Diabetologia **54**(4), 749–756 (2011)
19. Mol, A., Law, J.: Embodied action, enacted bodies: The example of hypoglycaemia. Body & society **10**(2-3), 43–62 (2004)
20. Mynatt, E.D., Abowd, G.D., Mamykina, L., Kientz, J.A.: Understanding the potential of ubiquitous computing for chronic disease management. Health Informatics: A Patient-Centered Approach to Diabetes. Health Informatics pp. 85–106 (2010)
21. Ngo, P.D., Wei, S., Holubová, A., Muzik, J., Godtliebsen, F.: Control of blood glucose for type-1 diabetes by using reinforcement learning with feedforward algorithm. Computational and mathematical methods in medicine **2018** (2018)

22. Ngo, P.D., Wei, S., Holubová, A., Muzik, J., Godtliebsen, F.: Reinforcement-learning optimal control for type-1 diabetes. In: 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), pp. 333–336. IEEE (2018)
23. NHS Choices: Type 1 diabetes (2018). URL `https://www.nhs.uk/conditions/type-1-diabetes/`
24. O'Kane, A.A., Han, Y., Arriaga, R.I.: Varied & bespoke caregiver needs: organizing and communicating diabetes care for children in the diy era. In: Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare, pp. 9–12 (2016)
25. O'Kane, A.A., Park, S.Y., Mentis, H., Blandford, A., Chen, Y.: Turning to peers: integrating understanding of the self, the condition, and others' experiences in making sense of complex chronic conditions. Computer Supported Cooperative Work (CSCW) **25**(6), 477–501 (2016)
26. Pesl, P., Herrero, P., Reddy, M., Oliver, N., Johnston, D.G., Toumazou, C., Georgiou, P.: Case-based reasoning for insulin bolus advice: evaluation of case parameters in a six-week pilot study. Journal of diabetes science and technology **11**(1), 37–42 (2017)
27. Rao, A.V.: A survey of numerical methods for optimal control. In: Advances in the Astronautical Sciences (2010)
28. Steil, G.M.: Algorithms for a closed-loop artificial pancreas: The case for proportional-integral-derivative control. Journal of Diabetes Science and Technology **7**(6), 1621–1631 (2013). DOI 10.1177/193229681300700623
29. Storni, C.: Complexity in an uncertain and cosmopolitan world. rethinking personal health technology in diabetes with the tag-it-yourself. PsychNology Journal **9**(2) (2011)
30. Sutton, R.S., Barto Andrew G.: Reinforcement Learning. The MIT Press (1998)