HAL open science

End-to-end Offline Reinforcement Learning for Glycemia Control

Tristan Beolet, Alice Adenis, Erik Huneker, Maxime Louis

To cite this version:

Tristan Beolet, Alice Adenis, Erik Huneker, Maxime Louis. End-to-end Offline Reinforcement Learning for Glycemia Control. 2023. ⟨hal-04241900⟩

HAL Id: hal-04241900

https://hal.science/hal-04241900v1

Preprint submitted on 14 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autorisation HAL

HAL Authorization

End-to-end Offline Reinforcement Learning for Glycemia Control

Tristan Beolet[1,*], Alice Adenis[1], Erik Huneker[1], Maxime Louis[1]

[a]*Diabeloop, 17 rue Félix Esclangon, Grenoble, 38000, France*

Abstract

The development of closed-loop systems for glycemia control in type I diabetes relies heavily on simulated patients. Improving the performances and adaptability of these close-loops raises the risk of over-fitting the simulator. This may have dire consequences, especially in unusual cases which were not faithfully -if at all- captured by the simulator. To address this, we propose to use offline RL agents, trained on real patient data, to perform the glycemia control. To further improve the performances, we propose an end-to-end personalization pipeline, which leverages offline-policy evaluation methods to remove altogether the need of a simulator, while still enabling an estimation of clinically relevant metrics for diabetes.

Keywords: Offline reinforcement learning, Glycemia control, Offline policy evaluation, Type 1 diabetes

1. Introduction

Type I diabetes is an autoimmune disease that leads to the destruction of beta cells in the pancreas. These cells play a crucial role in producing insulin, a hormone responsible for converting blood sugar into energy and regulating blood glucose levels. Without proper treatment, diabetic patients will often experience elevated blood sugar levels, which can lead to both short-term and long-term complications.

Currently, the only course of treatment is insulin therapy: the timely delivery of insulin throughout the day to counteract glucose intake and endogenous production. In *open-loop* treatment, the insulin delivery is done via boluses–large fast-acting insulin doses–accompanying meals and basal insulin–a stream of insulin acting steadily through the day–to counteract endogenous glucose production. Closed-loop systems [1, 2, 3, 4, 5]–or *artificial pancreas*–automatize this procedure: an algorithm connected to both a continuous glucose monitoring device (CGM) [6, 7] and an insulin pump takes real-time and periodic decisions

*Corresponding author: beolet.tristan@orange.fr

regarding insulin delivery. This solution not only optimizes glycemia control but also reduces the cognitive load associated with managing the condition.

Given the sensitive nature of these algorithms, whose actions have a direct and potentially harmful impact on the patient's health, the safety and accuracy of the design is of a paramount importance. In vivo testing of such algorithms is also very critical and expensive, hence the design of the algorithm must be made in silico beforehand and with very high confidence. Therefore the primary way to design and validate such algorithms is to use a virtual patient simulator. Such simulators [8] aim at simulating accurately and with enough variety the glycemia patterns of diabetic patients as well as their responses to meals, physical activities, stress, etc. They must also faithfully represent links between patients physiological parameters (e.g. weight, total daily dose etc.) and their glycemia patterns and response to insulin.

Such simulator-based design and validation may be good enough to build and validate high bias/low variance control algorithms, for which the risk is limited. This is how most current commercial closed loop systems have been developed. For instance, using a proportional-integral-derivative (PID) controller, a meal bolus calculator and cutting all insulin delivery in case of upcoming hypoglycemia is a design choice with high bias and low variance which leads to satisfactory performances (see e.g. Table 2 in [9]) in a closed loop, and is fairly robust to simulator biases.

Further improving the quality of a closed loop requires more complex control algorithms. Indeed, perfecting the closed loop control may require to take into account the individual patient physiology, the past patterns observed in the glycemia, the use of additional observables such as sensor data, a learning procedure regarding meals and physical activities, etc. As a consequence, the insulin function of the patient state must take a more flexible form, which is less robust to potential simulator biases. Online reinforcement learning (RL) algorithms [10, 11, 12, 13, 14, 15] in particular, which explicitly aim at overfitting the simulator, can prove particularly risky when used in real life. For instance, on Figure (2a) in [15], one can notice the low basal rate selected by the RL agent even when the glycemia remains above 225 mg/dL during the whole postprandial period. This behavior may be attributed to the simulator on which the model was trained [16], which is known to never plateau after meal intake, but returns to a steady glycemia state far away from meals. Hence, real life application of this RL agent may lead to lasting hyperglycemia after meals. In [17], the authors also show that the description of the hypoglycemia dynamics in the UVA/PADOVA simulator may not be faithful. In the end, relying on a simulator to train online RL algorithms for glycemia control may prove dangerous. Therefore, such algorithms must be used with a lot of additional safety -e.g. far from meals, far from hypoglycemia events or when the patient state/action is sufficiently close to real data [15]- which may decrease performances.

To address these shortcomings, offline reinforcement learning [18] -RL agents which only use data acquired using some existing policy- is a promising lead. The inability to make counterfactual queries and to explore using online data is replaced with either a policy constraint -e.g. the learned agent must stay close

to some existing agent- or an uncertainty-based method -prohibiting optimism in uncertain situations. Such an approach has been used in the context of glycemia control *on simulated data* in [19]. In this paper, the authors made a proof-of-concept: validating the performances and some aspects of safety and robustness of offline RL agents in this context. The fundamental limitation of their work is that it only uses simulated data and validates the RL agents through simulations: they do not demonstrate the applicability of the method on real life data, and do not free themselves from potential simulator limitations.

In this paper, we propose to use offline reinforcement learning to train agents to control the glycemia using *real data* acquired during the commercial exploitation of an existing closed-loop system. Then, we propose an end-to-end pipeline for offline patient-wise personalization of RL agents which *does not require a simulator*. To achieve this, we show how an adaptation of offline policy evaluation (OPE) techniques allows to directly estimate patient-wise clinically relevant metrics such as the time in range (see TIR in Table 2). Results show the validity and efficiency of the approach. To the best of our knowledge, it is the first use of offline RL techniques on real diabetic patient data.

In Section 2, we provide an overview of the existing literature on the topic. In Section 3, we explain how we choose to formalize the problem into an offline RL problem to build population models for control. In particular, we detail the selected offline RL algorithms, the choice of reward function and its importance, the construction of the states and important safety around the agents. In Section 4, we show the population model results: how they improve over the demonstration data and how they can deal with unannounced meals, in a fully closed loop fashion. Finally, we propose in Section 5 and end-to-end pipeline for patient-wise offline RL personalization without any use of simulator.

To summarize, our original contributions are:

- The first comparison of offline RL agents for glycemia control using a large set of real data,
- An adaptation of FQE to directly estimate key diabetes metrics,
- An end-to-end personalization procedure for glycemia control which does not require a simulator -even for validation.

2. Related work

Indirect RL use. A first group of research studies focuses on optimizing parameters within closed-loop systems through the application of reinforcement learning, resulting in an indirect influence on glycemia control. In [10, 11], an actor-critic algorithm is used to refine the insulin to carbohydrate ratio, the meal ratio and the reference basal used by a basal-bolus controller–a simple form of closed loop following the principles of insulin therapy. In [20], the authors use an actor-critic agent to tune both the insulin sensitivity and the insulin to carbohydrates ratio. In [13], the authors suggest a meal bolus calculator using Deep Deterministic Policy Gradient (DDPG) [21].

Full RL closed loops. Another group of papers directly trains reinforcement learning agents to calculate optimal insulin deliveries within a closed-loop system. In [12], deep Q-learning is used to compute the optimal basal value for glycemia control, while meal management is achieved using a traditional meal bolus calculator. Q-learning–using various neural network architectures–is used in [14] to train a full closed loop on virtual patients. This baseline work suffers limitations: meal information is not conveyed to the RL agents, there is no safety rule -even in the case of pending hypoglycemia- and the action space is discretized (into several fixed values of basal rates).

RL without a simulator. All of the works cited above use and rely on a virtual patient simulator. While it may be enough for a proof of concept, there remains too much uncertainty to test such agents on real patients -in particular with the management of unusual situations or edge cases. In [9], the authors investigate the use of offline RL for glycemia control. They use conservative Q-learning [22], Twin Delayed DDPG with Behavioral Cloning [23] and Batch Constrained Deep Q-learning [24]. To illustrate the potential of offline RL, they train agents on simulated data–collected using a simple proportional-integral-derivative controller–and evaluate the agent on a simulator. In this setting, they achieve promising results: the RL agents–and especially TD3-BC–largely improve over the collection policy. They also show how personalization can be made at the patient level, once again improving over the baseline RL agents. Their work suffers some limitations. First, their offline dataset was collected using a simulator and with additional noise to simulate exploration. This cannot be expected to translate to a real life dataset. Second, the patient state which they used does not contain any individual patient parameter or additional covariable, but merely the glycemia sequence, carbohydrates on board and insulin on board. Third, the UVA/PADOVA simulator [25] is an overly simplified benchmark for glycemia control [17]. While useful for demonstration purposes, it does not offer a representative panel of the variety of diabetic situations. In [15] for instance, the authors show that near perfect glycemia control can be reached. Finally, their work lack the use of offline policy evaluation methods, which enable to monitor closed loop performances without the use of a simulator.

3. Methods

3.1. Problem formalization

In a closed-loop system, the insulin pump operates with a CGM device -receiving real-time blood glucose level updates at fixed time intervals- and an insulin pump. Its primary objective is to maintain blood glucose within the range 70-180mg/dL. This task is accomplished by computing the appropriate quantity of insulin to deliver each time a new glycemia value is received. This complex decision-making process can be modeled using a Partially Observable

Markov Decision Process (POMDP). Therefore, reinforcement learning is a natural idea to build efficient closed loop systems. An *agent* -the algorithm overseeing the insulin pump's control- receives a description of the patient *state* at each time step, selects an *action* and receives the *new state* as well as a *reward*.

The state should contain any feature which contains information relative to the physiology and glucose level of the patient. Its construction will be detailed further in Section 3.4. While for most classical RL applications, the reward is fixed by the environment itself, we can here design any reward function to optimize. This choice is critical and will be explained in Section 3.6.

Meal boluses, ranging from 1 U to more than 15 U, constitute by far the largest instantaneous insulin deliveries. 99.7% of remaining insulin deliveries found in our data are below 10 U/h. Therefore, to decrease the risk associated with the use of RL, we choose to use a standard meal bolus calculator (as in [1]) to deal with announced meals, leaving the responsibility of all remaining insulin to the RL agent. We model each RL agent action as an insulin rate between 0 and 10 U/h. This way, the potential harm that may be caused by the RL agent is much lower than if the agent was allowed to prescribe meal boluses. To allow for some flexibility, we use a reasonably safe meal bolus calculator, which is unlikely to overestimate meal boluses. Doing so, the RL agents can still increase postprandial insulin, indirectly adjusting the meal bolus.

Additionally, since hypoglycemia events are the greatest immediate threats to the patients, in any situation where an impending risk of hypoglycemia is detected, all insulin deliveries are stopped. The criterion to measure the risk of hypoglycemia is a linear regression on the glycemia signal, and a condition on the predicted glycemia 15min-1h in the future.

Meal boluses and hypoglycemia prevention is used in all the experiments made in section 4.

3.2. RL algorithms

In Reinforcement Learning (RL), the primary objective for an agent is to maximize the cumulative return G defined as the sum of the discounted rewards $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, where R_t is the reward the agent receives at time step t.

The discount factor γ determines how much the agent values future rewards compared to immediate rewards, controlling the effective horizon. In our context, with $\gamma = 0.99$, the agent time horizon is of order $\frac{1}{1-\gamma} \simeq 8h$ since each time step lasts 5 minutes. This time horizon is relevant because close to the maximum duration of the effects of meals and boluses on the glycemia.

The state-action function, denoted as the Q-function, defined as $Q(s, a) = \mathbb{E}\big[G_t | s, a\big]$ is an analog of G. This function provides an estimate of the expected return when taking action a in state s.

While most RL applications to glycemia control are online and use a simulator, we argue that offline reinforcement learning offers much better perspectives. First, as discussed in the introduction, online approaches for glycemia control may lead to over-fitting of the biases of the simulator. Second, the now widely used closed loops systems have enabled the acquisition of vast quantities of data.

Third, the existing closed loop systems already give solid results on a wide variety of patients, and it is clear that all closed loop systems should follow policies which are reasonably close to existing systems. Indeed, existing systems perform well, are safe and follow a rationale from usual insulino-therapy, broadly validated by diabetologists. For all these reasons, offline RL algorithms constitute excellent approaches to improve on existing closed loop systems.

Capturing individualized glucose-insulin dynamics through model-based approaches is challenging due to inherent physiological variability. Therefore, model-based RL methods may be challenging. On the other hand, model-free methods offer a computationally efficient alternative, eliminating the need to predict these intricate dynamics at each decision step. Furthermore, the empirical performances of deep learning-backed model-free techniques, as evidenced by successes with DQN, A3C [26], and TRPO [27] in diverse domains, underscore their potential for this application. Coupled with the safety imperative of an offline approach—given the risks of online biases and catastrophic outcomes from incorrect dosing—model-free offline RL emerges as a robust solution. It leverages historical patient data, offering policies that are data-driven and individually adaptive without the pitfalls of model-induced errors. We focus on model-free offline RL methods in the rest of the paper.

3.3. Offline RL algorithms

Offline RL offers the advantage of learning from large real-world datasets but the impossibility to explore raises new challenges. A particular concern arising from this is distribution shift: as the newly trained policy is different from the behavior policy -policy used to collect the data-, the actions taken are also different from those the behavior policy would take. Different actions lead to different next states which can lead to a different distribution of encountered states -from the one observed from the behavior policy- and can lead to overconfidence in some states, posing potential risks.

Each offline algorithm used in this paper mitigates distributional shift differently:

- **Batch Constrained Q-learning (BCQ)**: a variational auto-encoder generates counterfactual queries, which represent a set of actions following the distribution of the training data, as described in [24]. The selected action is determined by choosing the one with the highest Q value, and clipped double Q learning is employed to limit Q-value overestimation.

- **Conservative Q-learning (CQL)**: Instead of learning an approximation of the Q-function, CQL estimate a lower bound function of the Q-function to reduce overestimation [22]. In-distribution state-action pair are assigned higher q values than out of distribution ones.

- **Twin Delayed DDPG with Behavioral Cloning (TD3-BC)**: An extension of the actor critic algorithm TD3 [28]. A behavioral cloning term in TD3 actor loss is used to select actions that are often seen in the

training data [23]. Clipped double Q-learning and Q-function smoothing are also used.

3.4. State construction

Given the data at hand, the available features to construct the state are described in Table 1.

Table 1: Data used to build the agent state.

Variables	**Description**
Glycemia History	Past records of blood sugar levels
Insulin History	Past records of insulin injection rates
Insulin Metrics	IOB (insulin on board, computed as in openAPS, see [29], and TDD (total daily dose, representing daily insulin needs)
Carbohydrate Metrics	COB (carbohydrate on board, indicating undigested sugar intake)
Time Metrics	Current time of day
Physiological Metric	Body weight

Glycemia history offers valuable insights into past blood sugar levels, enabling the RL agent to identify trends and patterns. This historical context is instrumental in predicting future glucose levels and assessing the success of prior insulin interventions. Insulin history complements glycemia history by providing a comprehensive record of insulin injection rates over time. Understanding how the patient has responded to insulin dosages in the past is useful for adapting recommendations to their evolving insulin sensitivity and individual insulin requirements. The IOB quantifies the insulin that has already been subcutaneously injected but has not yet had an action in the blood stream, influencing future glycemic responses. The TDD, on the other hand, offers an overview of daily insulin intake, aiding the agent in tailoring recommendations for glycemic control throughout the day. The COB introduces the crucial concept of undigested carbohydrate intake. When patients consume carbohydrates, blood sugar levels do not immediately reflect this intake. COB accounts for this delayed effect, allowing the agent to anticipate and mitigate potential glycemic spikes resulting from recent carbohydrate consumption. The consideration of time metrics, specifically the time of day, is essential. Sensitivity to insulin and liver activity naturally fluctuate throughout the day due to circadian rhythms and mealtime variations [30, 31]. Finally, the body weight can impact insulin sensitivity and metabolism.

The process of state construction involves careful optimization to determine the essential co-variables and the appropriate time horizon for time series data. After meticulous analysis, our research has revealed that the most effective state configuration comprises IOB, COB, TDD, time of day, and a one-hour window for the time series.

All features are normalized using zero-mean unit-variance or min-max scaling.

3.5. Metrics

In our performance assessment, we employ clinical metrics, as detailed in Table 2, universally recognized for evaluating glycemia control algorithms. The Time In Range (TIR) is the primary metric for closed loops. These metrics bridge machine learning agent to real-world outcomes, ensuring trained agents offer clinically satisfying glycemia control. The clinical targets for each metrics can be seen in the consensus found in [32].

Table 2: Blood Glucose Metrics

Metric	**Description**	**Target**
Time-in-Range (TIR)	Percentage of time where the glycemia is in the range: 70-180 mg/dL. Increased TIR is strongly associated with a reduced risk of developing micro-vascular complications [33].	> 70% [34]
Time-Below-Range (TBR)	Percentage of time where the glycemia is lower than 70 mg/dL also called time in hypoglycemia.	< 4% [34],[35]
Critical Time-Below-Range (TBR<54)	Percentage of time where the glycemia is lower than 54 mg/dL.	< 1% [36], [35]
Time-Above-Range (TAR)	Percentage of time where the glycemia is greater than 180 mg/dL, also called time in hyper glycemia.	< 25% [34] [35]
Coefficient of Variation (CV)	Relative dispersion of blood glucose values around their mean. A high value of the coefficient of variation entails a higher probability of vascular tissue damage [37].	< 36% [38], [35]
Mean glycemia	Mean blood glucose value (mg/dL). A high value increases the probability of dementia [39] and cardio-vascular tissue damage [37]	As low as possible

3.6. Reward function design

Unlike common RL tasks, there is no predefined reward function for glycemia control. The choice of reward function is absolutely crucial, as it drives entirely the behavior of the RL agents. For simplicity, we use reward functions which

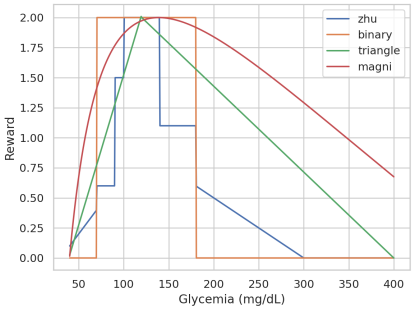

Figure 1: Candidate reward functions.

only depend on the current glycemia value. More sophisticated functions, depending on some history of glycemia or on the current rate of insulin may be used e.g. to penalize glycemia variations or too high insulin rates.

We show on Figure 1 a list of reward functions analyzed in this paper. The simplest is the binary reward whose maximization amounts to TIR maximization. Rewards from [12, 9, 1] are other alternatives. These rewards represent different trade-offs between hypoglycemia and hyperglycemia situations, as well as different tolerances to mild deviations from the ideal glycemia of 110 mg/dL.

To determine which reward function is the most appropriate without unnecessary computations, we look at how the performance metrics described in the previous section vary for each patient-day with respect to the sum of rewards on the same days. Such an analysis is shown on Figure 2. Magni and triangle rewards are the ones that correlate the least with TIR and TBR. Empirical testing showed that binary reward was not informative enough for high blood glucose values sometimes resulting in the inability to reduce the blood glucose level after meals.

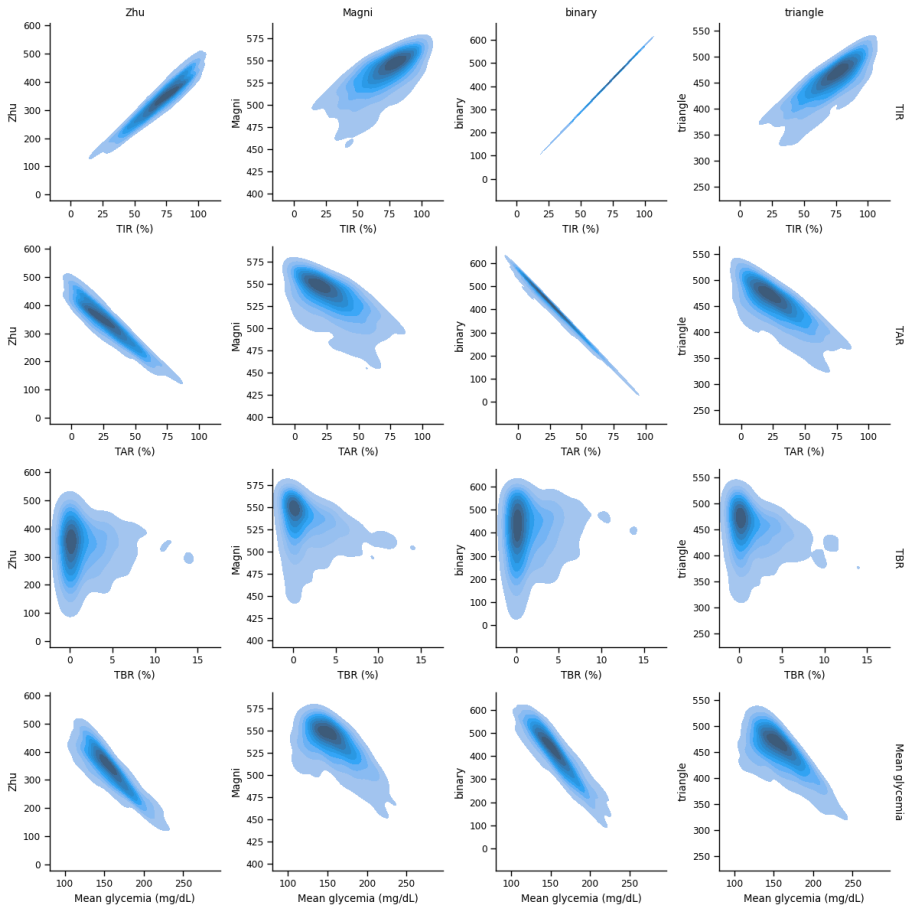

Figure 2: Correlation between clinical metrics for glycemia control against different reward functions, high correlation means that increasing the training reward will increase / decrease the clinical metric as wanted.

4. Experimental results: population models

We provide the implementation of all offline RL agents used and some of the training/evaluation pipelines on this repository.

4.1. Data acquisition and pre-processing

The data used to train and evaluate the RL agents has been collected through real life commercial usage of the DBLG1 artificial pancreas [1]. This closed loop system equips more than 10000 patients, with some patients wearing the system for more than 2 years. Among the ones agreeing to have their data collected, we selected 100 at random to form a training set for the RL agents. We filter to keep only days during which the closed loop was activated for more than 70

% of the time. Table 3 contains general characteristics regarding the data used. Overall, there are 6.9 million transitions in the dataset.

Note that the behavior policy is not *exactly* the DBLG1 algorithm as:

- Some open loop points may remain within the data (e.g. when disconnection with the pump or the CGM occurs)
- Even when the closed loop is activated, the patient has the possibility to modify boluses suggested by the algorithm (and does so in 1.4 % of cases) or to manually prescribe boluses (that represents 1.4 % of total boluses).
- Some parameters of the DBLG1 algorithm can be adjusted by the patients so as to improve the (perceived) quality of the closed loop. It may or may not improve the TIR/TAR/TBR metrics, depending on the patient particular sensitivity.

We argue that it is an advantage when using the data to train offline RL agents. Indeed, the state/action distribution in the data is larger than with the raw behavior policy, bringing more information during training of offline RL agents. An ablation study, removing the occurrences of manual and modified boluses for instance, would allow to quantify this effect, though beyond the scope of the work presented here.

Table 3: Training data details.

	Mean ± Std
Age	44.8 ± 13.0
Weight (kg)	77.1 ± 17.5
Total daily dose (U)	44.0 ± 18.1
Proportion of manually modified boluses (%)	1.4 ± 1.8
Proportion of manually prescribed boluses (%)	1.4 ± 2.4
TIR (%)	68.4 ± 9.9
TBR (%)	1.3 ± 1.6
TAR (%)	30.3 ± 10.1
CV	33.0 ± 4.5
Mean glycemia (mg/dL)	160.8 ± 14.6
Number of observed days	284.1 ± 161.2
Number of observations	69 000 ± 39 000

4.2. Algorithm comparison

We perform an hyper-optimization of TD3-BC, BCQ and CQL algorithms on the data presented in 4.1. Working with a limited compute budget -for economical and ecological reasons- we iteratively optimize several features in an A/B testing manner: the state composition (length of glycemia and insulin history), the reward function used and several hyper-parameters critical to each algorithm such as the RL/BC trade-off of TD3-BC.

Each trained agent performances are evaluated online on the simulator. While we argue that in silico validation should not be the gold standard of closed loop validations and design, it is still very informative, and we view it as a necessary -but not *sufficient*- condition for algorithm validation. In any case, we will show in the next section how to leverage offline policy evaluation methods to measure closed-loop candidate performances.

As mentioned before:

- When a meal is declared, the RL action is overridden by a meal bolus computed using the calculator in [1].
- When there an impeding risk of hypoglycemia is detected, the RL agent is deactivated and no can be sent to the patient.

The best performance metrics for each algorithm are shown on Table 4. TD3-BC and BCQ outperform the behavior policy in terms of TIR, TAR and mean glycemia. TD3-BC is better than the behavior policy for all metrics but the coefficient of variation, which remains within acceptable bounds. This shows the ability of offline RL algorithms to improve over the behavior policy, confirming the findings of [9].

Table 4: Comparison of the offline algorithm for glycemia control, values are Mean ± Std. Best value for each metric is in bold

	TIR	TBR	TBR<54
BCQ	70.4±7.59	3.87±4.17	0.98±1.85
CQL	57.8±10.52	9.93±9.57	5.13±6.71
TD3-BC	**74.38±7.3**	**2.73±3.71**	**0.86± 1.84**
Behavior policy	69.89± 7.97	3.59 ± 3.65	1.44 ± 2.37
	TAR	CV	Mean Glycemia
BCQ	25.72±7.01	39.38±7.84	**147.94±11.79**
CQL	32.27±11.76	42.94±11.92	150.75±20.5
TD3-BC	**22.89±5.86**	36.90± 7.79	148.62±10.82
Behavior policy	26.51 ±7.04	**33.55 ± 7.13**	156.59 ± 9.28

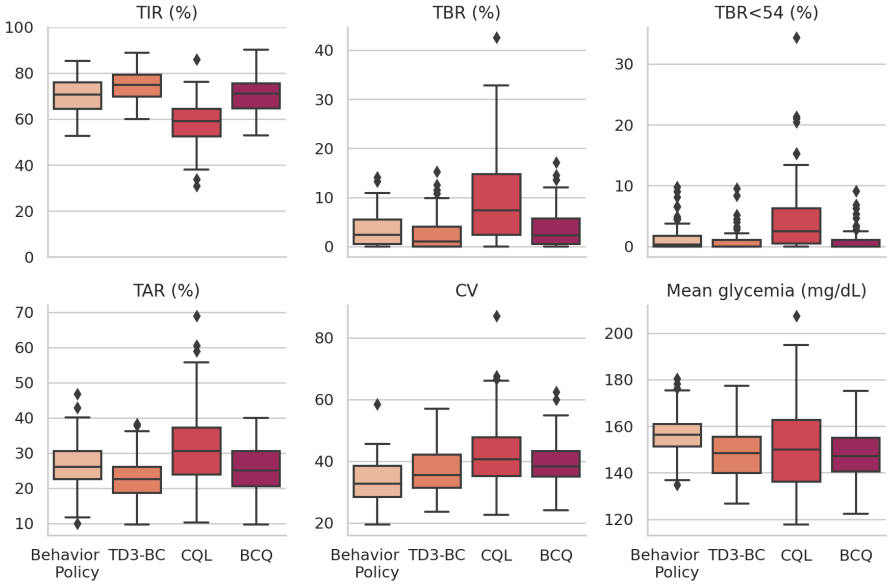

Figure 3: Comparison of the different policies on the simulator.

4.3. Population model analysis

The best-performing model -which we refer to as population model from now on-, TD3-BC, found in the comparison above has been compared on the simulator with the behavior policy. We can see from Table 4 that the TD3-BC agent is able to simultaneously improve the TIR by 4.49%, and the TBR by 0.86%. Note that in any case, the inter-patient performance variability remains quite high e.g. TIR range from around 50% to above 80%.

An example of control on the simulator is presented on Figure 4. The observed differences are informative. The RL agent is much more aggressive in hyperglycemia situations, but also seems to anticipate more when the glycemia is high but decreasing. Also, during the meal periods, the RL agent is more aggressive, especially around 1h after the meal.

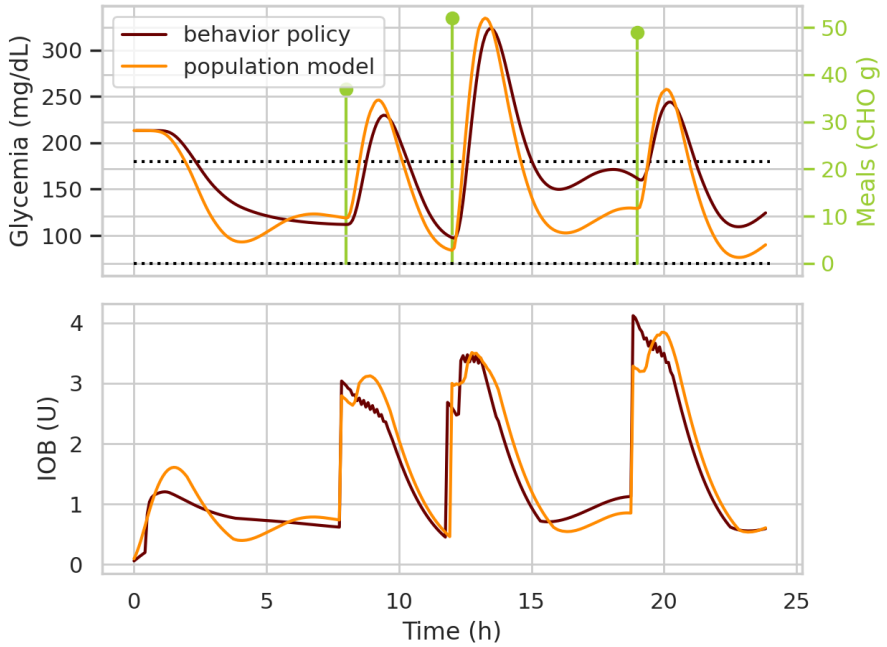

Figure 4: Comparison of the RL agent and the behavior policy on the simulator. The RL agent is more aggressive: at the start to reach a value close to 110 mg/dL, and also add more basal after the meal bolus. Mean glycemia is thus lower for the population model.

The ability for a closed loop to deal with unannounced meals is of a paramount importance as it further decreases the burden of the disease on diabetic patients. To ensure that the population model is capable of handling such cases, we run a simulation with unannounced meals. In this scenario, the virtual patients don't declare any meals: the COB feature given to the offline agent and the behavior policy is always null and no meal bolus is automatically administered.

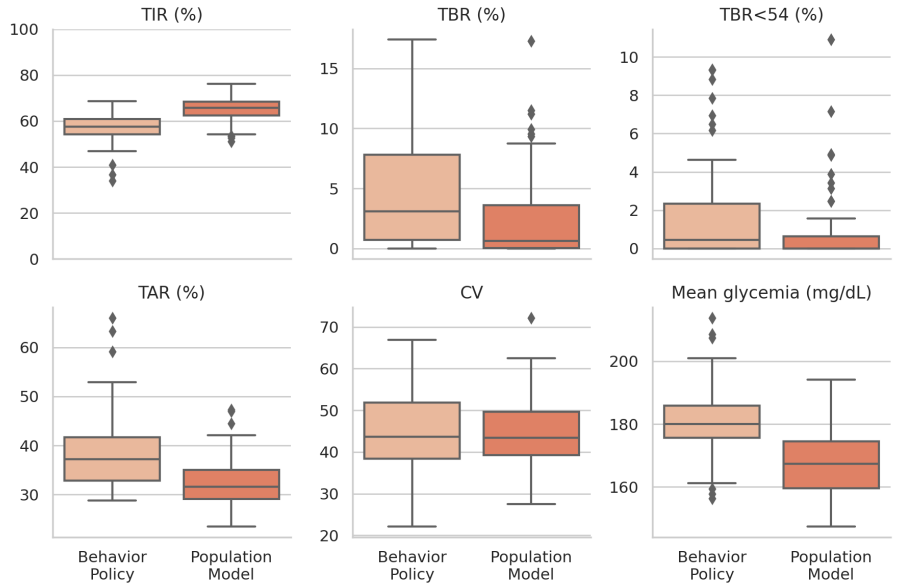

Figure 5: In silico comparison of behavior policy with personalized model in the case of unannounced meals. Simultaneous improvement of TIR, TBR, and mean glycemia.

Results presented on Figure 5, which compares the behavior policy and the population model. There is a significant improvement in the performance of our Reinforcement Learning (RL) agent compared to the behavior policy. Specifically, we observe a substantial 8.0% increase in the TIR and a 6.1% reduction in the mean TBR. Notably, the most significant improvement is observed in the mean glycemia, which decreases by an impressive 13.2 mg/dL. This reduction has promising implications for patient health, particularly in scenarii where meal announcements may be overlooked. These results mark an important initial step toward achieving a fully closed-loop glycemia control system, one that operates seamlessly without the need for meal declarations from patients.

5. An end-to-end pipeline for patient-wise personalization

In this section, we describe a pipeline for offline patient-wise model personalization. We start by proposing an application of fitted Q-evaluation which allows to estimate clinically relevant metrics. We then describe the experimental setup we propose and present the results.

5.1. Fitted Q evaluation

In order to evaluate an offline agent without the use of a simulator, off-policy evaluation methods can be used. Let π the new policy of the agent to

evaluate. Fitted Q evaluation (FQE) learns the Q function under the policy π using Bellman equation [40]:

$$Q(s_t, a_t) = c(s_t, a_t) + \gamma Q(s_{t+1}, \pi(s_{t+1}))$$

where (s_t, a_t, s_{t+1}) is a transition (state, action, next state) in the offline dataset and c is a reward function. When training FQE to estimate the Q function under the new policy, the reward function c can be different from the one used to train the agent. We leverage this possibility to devise a method which directly estimates TIR/TAR/TBR.

In particular, if the reward function $c = c_{\text{TIR}}$ is chosen as

$$c_{\text{TIR}}(x) = \begin{cases} 1 & \text{if } x \in [70, 180] \text{ mg/dL} \\ 0 & \text{else} \end{cases}$$

then

$$\mathbb{E}_\pi[c_{\text{TIR}}(a, s)] = \text{TIR}_\pi$$
$$Q(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k c_{\text{TIR}_k}(s_k, a_k)\right] = \frac{\text{TIR}_\pi}{1-\gamma}$$

where TIR_π is the TIR under the policy π, since the expected value of the reward is precisely the proportion of time spent in the normoglycemia range. Similarly, TBR and TAR can also be estimated with the following reward functions: $c_{\text{TBR}}(x) = \mathbb{1}_{x<70}(x), c_{\text{TAR}}(x) = \mathbb{1}_{x>180}(x)$. These choices of rewards for FQE allow to estimate clinically relevant metrics without any simulation. Note also that these estimations can be made in theory from any state, allowing for immediate short term predictions as well as global patient-wise estimations of TIR/TAR/TBR.

In [40], the authors showed that the estimated value obtained from the FQE algorithm is not always accurate but can effectively be used to rank different algorithms. This caveat is to keep in mind, and constitutes a limitation of this work. A detailed analysis of the performances of FQE depending on the distribution of data would be informative and will be part of further work.

5.2. Personalization protocol

A well-known limitation of current closed-loops is their high inter-patient variability, which can be seen on Figure 3 and Table 4.

To limit this issue, we propose a pipeline to personalize the agent to individual patient data. Starting from an offline RL agent, the training procedure is sketched on Figure 6:

1. The first 25% of the patient data is used as a data set for fine-tuning the agent,
2. The following 25% of the patient data is used to train a specific FQE model for each metric reward/TIR/TAR/TBR,

3. The following 25% of the patient data is used to evaluate the estimated metrics of the trained FQE models. The estimated reward serves for best checkpoint selection throughout the personalization, so as to prevent at best over-fitting and catastrophic forgetting,
4. At the end of the training, the final 25% of data is used to provide an unbiased measure of generalization performances, using the FQE models of the best checkpoint.

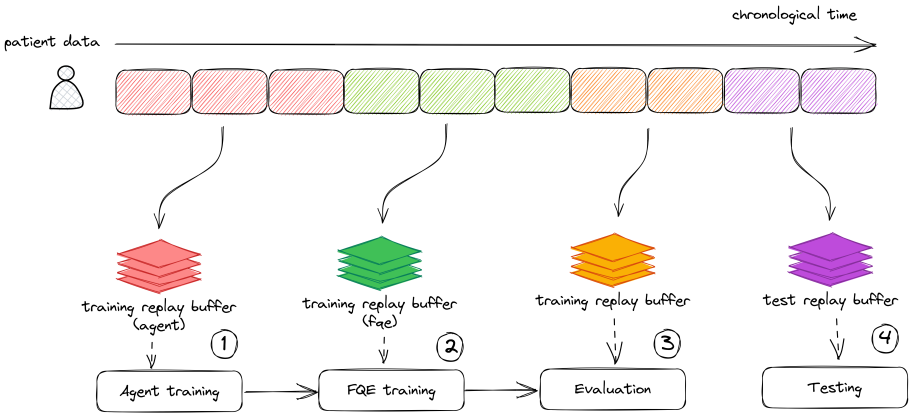

Figure 6: Personalization training procedure, each patient data is split into 4 chronological sets: one to train the agent on the patient data, one to train FQE model on, one to run validation during training and one to test the agent performance.

In this series of experiments, we do not use the meal boluses and the hypoglycemia cutting logic: we evaluate the performances of the sole RL agent. This gives us a more accurate insight into the effect of the personalization of the whole control. This explains why the baseline performances in this section are lower (in fact, achieving such high TIR with an insulin rate lower than 10 U/h is a success). Of course, there is no obstacle in also running FQE evaluation with these features enabled.

To enable a fair comparison of the personalized models with the population model, we also train FQE models for reward/TIR/TBR/TAR for the population model, on the union of all data sets used for the FQE of the personalized models. We then evaluate these FQE models on the last 25% data of each patient. These estimates of reward/TIR/TBR/TAR are the performances to beat.

5.3. Results

Overall results. We perform the personalization procedure on 25 patients separately. These patients are additional patients, not included in the training set of the population model. These patients have been observed on average 333 days, which means that about 3 months are used for the actual offline fine-tuning of the RL agent -the rest being used by the validation and test FQE for each metric.

We report the overall results on Figure 7. All metrics improve on average with the personalization. The average Q-value estimated for the training reward increases by 10, the average TIR increases of 1%, the average TBR increases by 1 % and the TAR average is unchanged. Notably, worst cases are greatly improved e.g. the worst TIR is estimated to increase from 38 % up to 50%.

Figure 8 gives the variations of each estimated metric for each patient between the population model and the personalized model. Note that although our checkpoint selection criterion throughout the personalization is based on the training reward, almost metrics do improve for almost all patients.

This shows that 3 months of data is enough to perform with success an offline patient-wise personalization of RL agents. Investigating more precisely the connection between the agents improvements and the quantity of data available would be another interesting continuation of this work.

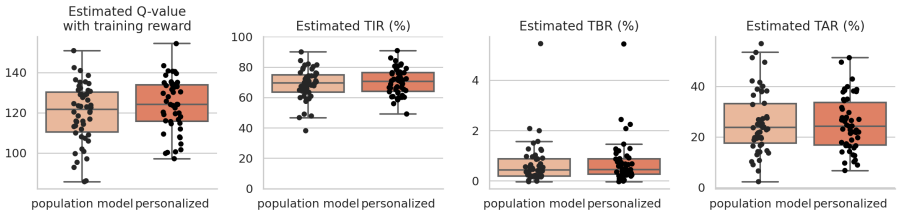

Figure 7: FQE estimation of the clinical metric before and after personalization, selecting the best model on training reward (Zhu).

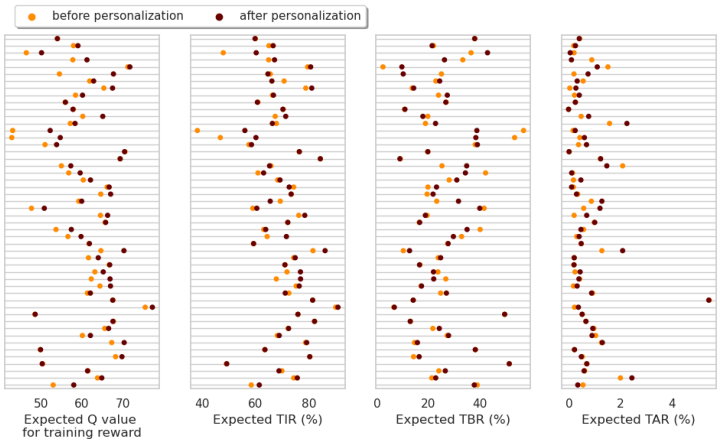

Figure 8: FQE estimation of each clinical metric before and after personalization for each patient. Almost all metrics improve for all patients.

Personalization analysis. To further evaluate how the personalization works, we use the fact that insulin ingested at time t has its largest impact on glycemia at

time around t + 30 minutes. Therefore, if for some state in the real data, the glycemia 30 minutes later was high (resp. low), then the new control improved over the existing control if it suggested to send an higher (resp. lower) quantity of insulin 30 minutes before. We make this analysis on Figure 9 which shows the basal rates delivered by the personalized models and the population model, with respect to the glycemia 30 minutes in the future. When the future glycemia is below 200 mg/dL, the personalized agents send less insulin on average. When the future glycemia is above 200 mg/dL, the personalized agents send more insulin on average. This constitutes another validation of the quality of the personalization.

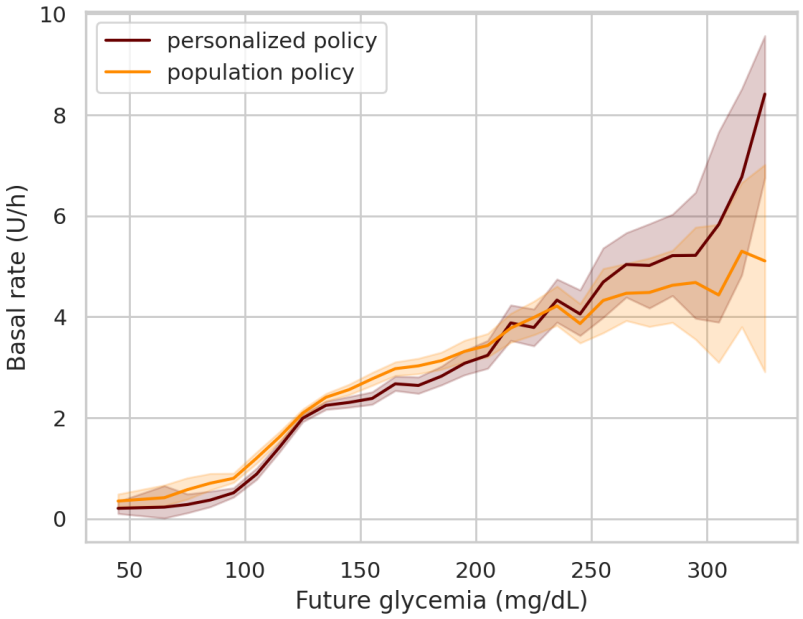

Figure 9: Variation of the prescribed basal rate between the population and personalized policy. X axis is the glycemia 30 minutes in the future. If the glycemia at t + 30 min is high (resp. low), the personalization is successful if it increased (resp. decreased) the basal rate with respect to the population model. This is what we observe here: the personalization led to increased basal rates above 200 mg/dL, and decreased basal rates below.

6. Conclusion

We performed an extensive comparison of offline RL algorithms for glycemia control on real data. The offline RL often outperforms the behavior policy. For example, the best TD3-BC model we trained has +7% TIR, -1% TBR and -12 mg/dL of mean glycemia on average, across a family of in silico patients. Further

in silico evaluations with no announced meals illustrate the high robustness and accuracy of the RL control.

Going further, we showed how personalization of such RL agents can be made on individual patients in a realistic setting. This is illustrated using OPE methods, in a way that enables to recover key diabetics metrics directly -instead of a hard-to-interpret Q-value estimate. A couple of months of observations allows to improve the patients' TIR by an estimated 1 % on average and to drastically reduce the variance of inter-subject performances. Such an improvement of worst cases of glycemia control in closed loop systems is of particular interest, as it is often a limitation of current commercial articifical pancreas.

Continuation of this work could include an ablation study to see if manual patient actions within the dataset do improve to the offline RL trainings. Additionally, a more rigorous evaluation of the FQE method applied to TIR/TBR/TAR estimation may be interesting.

Acknowledgements

The completion of this paper has been made possible thanks to the support of Diabeloop SA. The data used in the paper has been provided by Diabeloop SA while complying with the RGPD reglementation. We also thank Pierre Gauthier for his contributions on the simulator-related experiments in the paper.

References

[1] C. Amadou, S. Franc, P.-Y. Benhamou, S. Lablanche, E. Huneker, G. Charpentier, A. Penfornis, Diabeloop dblg1 closed-loop system enables patients with type 1 diabetes to significantly improve their glycemic control in real-life situations without serious adverse events: 6-month follow-up, Diabetes Care 44 (3) (2021) 844–846.

[2] B. Pintaudi, I. Gironi, R. Nicosia, E. Meneghini, O. Disoteo, E. Mion, F. Bertuzzi, Minimed medtronic 780g optimizes glucose control in patients with type 1 diabetes mellitus, Nutrition, Metabolism and Cardiovascular Diseases 32 (7) (2022) 1719–1724.

[3] E. C. Cobry, C. Berget, L. H. Messer, G. P. Forlenza, Review of the omnipod® 5 automated glucose control system powered by horizon™ for the treatment of type 1 diabetes, Therapeutic Delivery 11 (8) (2020) 507–519.

[4] K. K. Hood, N. Garcia-Willingham, S. Hanes, M. L. Tanenbaum, J. Ware, C. K. Boughton, J. M. Allen, M. E. Wilinska, M. Tauschmann, L. Denvir, et al., Lived experience of camaps fx closed loop system in youth with type 1 diabetes and their parents, Diabetes, Obesity and Metabolism 24 (12) (2022) 2309–2318.

[5] L. Ekhlaspour, M. J. Schoelwer, G. P. Forlenza, M. D. DeBoer, L. Norlander, L. Hsu, R. Kingman, E. Boranian, C. Berget, E. Emory, et al., Safety and performance of the tandem t: slim x2 with control-iq automated insulin delivery system in toddlers and preschoolers, Diabetes Technology & Therapeutics 23 (5) (2021) 384–391.

[6] S. K. Garg, M. Kipnes, K. Castorino, T. S. Bailey, H. K. Akturk, J. B. Welsh, M. P. Christiansen, A. K. Balo, S. A. Brown, J. L. Reid, et al., Accuracy and safety of dexcom g7 continuous glucose monitoring in adults with diabetes, Diabetes Technology & Therapeutics 24 (6) (2022) 373–380.

[7] A. Blum, Freestyle libre glucose monitoring system, Clinical Diabetes 36 (2) (2018) 203–204.

[8] C. Dalla Man, R. A. Rizza, C. Cobelli, Meal simulation model of the glucose-insulin system, IEEE Transactions on biomedical engineering 54 (10) (2007) 1740–1749.

[9] H. Emerson, M. Guy, R. McConville, Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes, arXiv preprint arXiv:2204.03376 (2022).

[10] E. Daskalaki, P. Diem, S. G. Mougiakakou, An actor–critic based controller for glucose regulation in type 1 diabetes, Computer methods and programs in biomedicine 109 (2) (2013) 116–125.

[11] E. Daskalaki, P. Diem, S. G. Mougiakakou, 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), IEEE, 2013, pp. 3487–3490.

[12] T. Zhu, K. Li, P. Herrero, P. Georgiou, Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation, IEEE Journal of Biomedical and Health Informatics 25 (4) (2020) 1223–1232.

[13] T. Zhu, K. Li, L. Kuang, P. Herrero, P. Georgiou, An insulin bolus advisor for type 1 diabetes using deep reinforcement learning, Sensors 20 (18) (2020) 5058.

[14] I. Fox, J. Wiens, Reinforcement learning for blood glucose control: Challenges and opportunities (2019).

[15] M. Louis, H. R. Ugalde, P. Gauthier, A. Adenis, Y. Tourki, E. Huneker, Safe reinforcement learning for automatic insulin delivery in type i diabetes (2022).

[16] R. Hovorka, F. Shojaee-Moradie, P. V. Carroll, L. J. Chassin, I. J. Gowrie, N. C. Jackson, R. S. Tudor, A. M. Umpleby, R. H. Jones, Partitioning glucose distribution/transport, disposal, and endogenous production during ivgtt, American Journal of Physiology-Endocrinology and Metabolism 282 (5) (2002) E992–E1007.

[17] G. M. Steil, Best use of models to advance the artificial pancreas (2018).

[18] S. Levine, A. Kumar, G. Tucker, J. Fu, Offline reinforcement learning: Tutorial, review, and perspectives on open problems, arXiv preprint arXiv:2005.01643 (2020).

[19] H. Emerson, M. Guy, R. McConville, Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes, Journal of Biomedical Informatics 142 (2023) 104376.

[20] E. Daskalaki, P. Diem, S. G. Mougiakakou, Model-free machine learning in biomedicine: Feasibility study in type 1 diabetes, PloS one 11 (7) (2016) e0158722.

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971 (2015).

[22] A. Kumar, A. Zhou, G. Tucker, S. Levine, Conservative q-learning for offline reinforcement learning, Advances in Neural Information Processing Systems 33 (2020) 1179–1191.

[23] S. Fujimoto, S. S. Gu, A minimalist approach to offline reinforcement learning, Advances in neural information processing systems 34 (2021) 20132–20145.

[24] S. Fujimoto, D. Meger, D. Precup, Off-policy deep reinforcement learning without exploration, in: International conference on machine learning, PMLR, 2019, pp. 2052–2062.

[25] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, C. Cobelli, The uva/padova type 1 diabetes simulator: new features, Journal of diabetes science and technology 8 (1) (2014) 26–34.

[26] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: International conference on machine learning, PMLR, 2016, pp. 1928–1937.

[27] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International conference on machine learning, PMLR, 2015, pp. 1889–1897.

[28] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International conference on machine learning, PMLR, 2018, pp. 1587–1596.

[29] OpenAPS, Understanding insulin on board (iob) calculations, accessed: 2023-10-01 (2023).
URL https://openaps.readthedocs.io/en/latest/docs/While%20You%20Wait%20For%20Gear/understanding-insulin-on-board-calculations.html

[30] A. D. Grant, D. M. Lewis, L. J. Kriegsfeld, Multi-timescale rhythmicity of blood glucose and insulin delivery reveals key advantages of hybrid closed loop therapy, Journal of Diabetes Science and Technology 16 (4) (2022) 912–920.

[31] P. Belsare, A. Bartolome, C. Stanger, T. Prioleau, Understanding temporal changes and seasonal variations in glycemic trends using wearable data, Science Advances 9 (38) (2023) eadg2132.

[32] M. de Bock, E. Codner, M. E. Craig, T. Huynh, D. M. Maahs, F. H. Mahmud, L. Marcovecchio, L. A. DiMeglio, Ispad clinical practice consensus guidelines 2022: Glycemic targets and glucose monitoring for children, adolescents, and young people with diabetes, Pediatric diabetes 23 (8) (2022) 1270.

[33] R. W. Beck, R. M. Bergenstal, T. D. Riddlesworth, C. Kollman, Z. Li, A. S. Brown, K. L. Close, Validation of time in range as an outcome measure for diabetes clinical trials, Diabetes care 42 (3) (2019) 400–405.

[34] T. Battelino, T. Danne, R. M. Bergenstal, S. A. Amiel, R. Beck, T. Biester, E. Bosi, B. A. Buckingham, W. T. Cefalu, K. L. Close, et al., Clinical

targets for continuous glucose monitoring data interpretation: recommendations from the international consensus on time in range, Diabetes care 42 (8) (2019) 1593–1603.

[35] R. I. Holt, J. H. DeVries, A. Hess-Fischl, I. B. Hirsch, M. S. Kirkman, T. Klupa, B. Ludwig, K. Nørgaard, J. Pettus, E. Renard, 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 44 (11) (2021) 2589–2625.

[36] A. D. Association, 6. glycemic targets: standards of medical care in diabetes—2021, Diabetes Care 44 (Supplement_1) (2021) S73–S84.

[37] A. Ceriello, L. Monnier, D. Owens, Glycaemic variability in diabetes: clinical and therapeutic implications, The lancet Diabetes & endocrinology 7 (3) (2019) 221–230.

[38] T. Danne, R. Nimri, T. Battelino, R. M. Bergenstal, K. L. Close, J. H. DeVries, S. Garg, L. Heinemann, I. Hirsch, S. A. Amiel, et al., International consensus on use of continuous glucose monitoring, Diabetes care 40 (12) (2017) 1631–1640.

[39] P. K. Crane, R. Walker, R. A. Hubbard, G. Li, D. M. Nathan, H. Zheng, S. Haneuse, S. Craft, T. J. Montine, S. E. Kahn, et al., Glucose levels and risk of dementia, New England Journal of Medicine 369 (6) (2013) 540–548.

[40] H. Le, C. Voloshin, Y. Yue, Batch policy learning under constraints, in: International Conference on Machine Learning, PMLR, 2019, pp. 3703–3712.