Universidad Politécnica de Cartagena

UPCT Escuela Internacional de Doctorado

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

TESIS DOCTORAL

# EVALUACIÓN DEL CONTROL DEL NIVEL DE GLUCOSA EN SANGRE EN PACIENTES DIABÉTICOS TIPO 1 USANDO APRENDIZAJE REFORZADO

Presentada por Phuwadol Viroonluecha para optar al
grado de Doctor
por la Universidad Politécnica de Cartagena

Dirigida por:
Dr. Esteban Egea López

Codirigida por:
Dr. José Santa Lozano

Cartagena, 2023

Universidad Politécnica de Cartagena

UPCT ESCUELA INTERNACIONAL DE DOCTORADO

DOCTORAL PROGRAMME IN INFORMATION AND COMMUNICATION TECHNOLOGIES

PhD THESIS

# EVALUATION OF BLOOD GLUCOSE LEVEL CONTROL IN TYPE 1 DIABETIC PATIENTS USING ONLINE AND OFFLINE REINFORCEMENT LEARNING

Presented by Phuwadol Viroonluecha
to the Technical University of Cartagena in fulfilment of
the thesis requirement for the award of PhD

Supervisors:
Dr. Esteban Egea López
Dr. José Santa Lozano

Cartagena, 2023

# Acknowledgments

I would like to extend my sincere appreciation to my supervisors, Professor Esteban Egea-López and Professor José Santa, for their unwavering guidance, support, and encouragement throughout my Ph.D. journey. Despite their busy schedules, they were always there to provide insightful advice, even during leisure and holiday periods and were prompt in responding to emails within a few hours. Their dedication has been truly invaluable to me and my PhD journey. I am also grateful to Professor Pablo Pavón Mariño who invited me to pursue my PhD in the GIRTEL group at the Technical University of Cartagena.

I would like to express my thanks to Professor Mª Victoria Bueno Delgado, Dr. José Luis Romero Gázquez, Dr. Miquel Garrich Alabarce, and other GIRTEL members for their support and for facilitating my adaptation to life in Spain. I am also grateful to Professor Nuno Manuel Garcia and the ALLab team at the University of Beira Interior in Portugal for their warm welcome during my research stay.

Thank you to my family and friends who have always enquired about the progress and the completion of my doctoral studies, despite their lack of knowledge regarding the topic of my research. Your support and encouragement have been invaluable to me throughout this journey.

I would like to express my sincere gratitude to my dear Auggareema for providing unwavering emotional support throughout the challenging pandemic period, which helped me stay sane and focused during the writing of this thesis.

Last but not least, I am grateful for the financial support provided by the following grants that supported the work carried out in this thesis:

- AEI Formación de Personal Investigador (FPI) Predoctoral Grant (PRE2018-084260).
- Agencia Estatal de Investigación (AEI), Fondo Europeo de Desarrollo Regional (FEDER), and Unión Europea (UE) (TEC2017-84423-C3-2-P (ONOFRE-2)).
- Project PID2020-112675RB-C41 (ONOFRE-3) funded by Agencia Estatal de Investigación (AEI), MCIN/AEI/10.13039/501100011033

*"I walk slowly, but I never walk backward."*

Abraham Lincoln

*"If you can't run, then walk. If you can't walk, then crawl, but whatever you do, you have to keep moving."*

Martin Luther King Jr.

# Resumen

Los pacientes con diabetes tipo 1 deben monitorizar de cerca sus niveles de glucemia y administrar se insulina para controlarlos. Se han propuesto métodos de control automatizado de la glucemia que eliminan la necesidad de intervención humana y, recientemente, el aprendizaje por refuerzo, un tipo de algoritmo de aprendizaje automático, se ha utilizado como un método efectivo de control en entornos simulados.

Actualmente, los métodos utilizados para los pacientes con diabetes, como el régimen basal-bolus y los monitores continuos de glucemia, tienen limitaciones y todavía requieren intervención manual. Los controladores PID se utilizan ampliamente por su simplicidad y robustez, pero son sensibles a factores externos que afectan su efectividad. Los trabajos existentes en la literatura se han enfocado principalmente en mejorar la precisión de estos algoritmos de control. Sin embargo, todavía hay margen para mejorar la adaptabilidad al perfil de cada paciente. La siguiente fase de investigación tiene como objetivo mejorar los métodos actuales y adaptar los algoritmos para controlar mejor los niveles de glucemia. La aplicación del aprendizaje máquina en el campo del control de la diabetes ha demostrado ser un campo con un gran desarrollo en los últimos años. Por ello, se opina que una solución con gran potencial es usar el aprendizaje por refuerzo (RL) para entrenar los algoritmos en base a datos individuales del paciente.

En esta tesis, proponemos un control en lazo cerrado para los niveles de glucemia basado en el aprendizaje profundo por refuerzo. Describimos la evaluación inicial de varias alternativas llevadas a cabo en un simulador realista del sistema glucorregulador y proponemos una estrategia de implementación particular basada en reducir la frecuencia de las observaciones y recompensas pasadas al agente, y usar una función de recompensa simple. El trabajo se centra en entrenar agentes con esa estrategia para tres grupos de clases de pacientes, evaluarlos y los compararlos con otras alternativas. Nuestros resultados muestran que nuestro método con *Proximal Policy Optimization* es capaz de superar a los métodos tradicionales, así como a propuestas similares recientes, al lograr períodos más prolongados de estado glicémico seguro y de bajo riesgo.

Como extensión del aporte anterior, constatamos que la aplicación práctica de los algoritmos de control de glucemia requeriría interacciones de prueba y error con los pacientes, lo que es una limitación para entrenar el sistema de manera efectiva. Como alternativa, el aprendizaje reforzado sin conexión no requiere interacción con humanos y la investigación previa sugiere que se pueden lograr resultados prometedores con conjuntos de datos obtenidos sin interacción, similar a los algoritmos de aprendizaje automático clásicos. Sin embargo, aún no se ha evaluado la aplicación del aprendizaje reforzado sin conexión al control de la glucemia. Por lo tanto, en esta tesis, evaluamos exhaustivamente dos algoritmos de aprendizaje reforzado sin conexión para el control de glucemia y examinamos su potencial y limitaciones. Evaluamos el impacto del método utilizado para generar los conjuntos de datos de entrenamiento, el tipo de trayectorias empleadas (secuencias de estados, acciones y recompensas experimentadas por un agente en un entorno, la calidad de las trayectorias y el tamaño de los conjuntos de datos en el entrenamiento, y el rendimiento, y los comparamos con las alternativas como PID y *Proximal Policy Optimization*. Nuestros resultados demuestran que uno de los algoritmos de aprendizaje reforzado sin conexión evaluados, *Trajectory Transformer*, es capaz de rendir al mismo nivel que los algoritmos de aprendizaje reforzado convencionales, pero sin necesidad de interacción con pacientes reales durante el entrenamiento.

# Abstract

Patients with Type 1 diabetes are required to closely monitor their blood glucose levels and administer insulin to manage them. Automated glucose control methods that eliminate the need for human intervention have been proposed, and recently, reinforcement learning, a type of machine learning algorithm, has been used as an effective control method in simulated environments.

Currently, the methods used for diabetes patients, such as the basal-bolus regime and continuous glucose monitors, have limitations and still require manual intervention. The PID controllers are widely used for their simplicity and robustness, but they are sensitive to external factors affecting their effectiveness. The existing works in the research literature have mainly focused on improving the accuracy of these control algorithms. However, there is still room for improvement regarding adaptability to individual patients. The next phase of research aims to further optimize the current methods and adapt the algorithms to better control blood glucose levels. Machine learning proposals have paved the way partially, but they can generate generic models with limited adaptability. One potential solution is to use reinforcement learning (RL) to train the algorithms based on individual patient data.

In this thesis, we propose a closed-loop control for blood glucose levels based on deep reinforcement learning. We describe the initial evaluation of several alternatives conducted on a realistic simulator of the glucoregulatory system and propose a particular implementation strategy based on reducing the frequency of the observations and rewards passed to the agent, and using a simple reward function. We train agents with that strategy for three groups of patient classes, evaluate and compare it with alternative control baselines. Our results show that our method with Proximal Policy Optimization is able to outperform baselines as well as similar recent proposals, by achieving longer periods of safe glycemic state and low risk.

As an extension of the previous contribution, we have noticed that, practical application of blood glucose control algorithms would necessitate trial-and-error interaction with patients, which could be a limitation for effectively training the system. As an alternative, offline reinforcement learning does not require interaction with subjects and preliminary research suggests that promising results can be achieved with datasets obtained offline, similar to classical machine learning algorithms. However, application of offline reinforcement learning to glucose control has to be evaluated yet. Thus, in this thesis, we comprehensively evaluate two offline reinforcement learning algorithms for blood glucose control and examine their potential and limitations. We assess the impact of the method used to generate training datasets, the type of trajectories employed (sequences of states, actions, and rewards experienced by an agent in an environment over time), the quality of the trajectories, and the size of the datasets on training and performance, and compare them to commonly used baselines such as PID and Proximal Policy Optimization. Our results demonstrate that one of the offline reinforcement learning algorithms evaluated, Trajectory Transformer, is able to perform at the same level as the baselines, but without the need for interaction with real patients during training.

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **ANNs** | Artificial Neural Networks. *(pp. 7)* |
| **AP** | Artificial Pancreas. *(pp. 6, 14, 18, 58)* |
| **BB** | Basal-bolus Regime. *(pp. 5, 6, 10, 14, 20, 31, 33–36, 68)* |
| **BB-CD** | Basal-bolus Regime with Cooldown. *(pp. 20, 32–36, 68)* |
| **BCQ** | Batch Constrained Deep Q-learning. *(pp. 20)* |
| **BG** | Blood Glucose. *(pp. 5–10, 13–20, 22–25, 28–31, 33, 35–38, 41, 43, 50, 55–58, 63, 65, 68–70)* |
| **BGRI** | Blood Glucose Risk Index. *(pp. 23, 24, 27, 28)* |
| **BMR** | Basal Metabolic Rate. *(pp. 41)* |
| **CF** | Correction Factor. *(pp. 32)* |
| **CGM** | Continuous Glucose Monitor. *(pp. 6–8, 10, 13, 14, 18, 19, 22, 23, 27–30, 37, 41, 50, 56, 57, 70)* |
| **CHO** | Carbohydrate. *(pp. 23, 28)* |
| **CMA-ES** | Covariance matrix adaptation evolution strategy. *(pp. 42, 43)* |
| **CQL** | Conservative Q-learning. *(pp. 20)* |
| **CR** | Carbohydrate Ratio. *(pp. 32)* |
| **DDE** | Nonlinear Differential Difference Equation. *(pp. 18)* |
| **DDPG** | Deep Deterministic Policy Gradient. *(pp. 19, 20)* |
| **Decis-PID-IF** | Decision Transformer with the dataset generated from Proportional Integrative Derivative with Insulin Feedback. *(pp. 55, 58–60)* |

| | |
|---|---|
| **Decis-PPO** | Decision Transformer with the dataset generated from Proximal Policy Optimization. *(pp. 55, 58, 59, 61)* |
| **DH** | Dual Hormone. *(pp. 12)* |
| **dL** | Decilitre. *(pp. 24, 33, 37, 50, 55, 57, 58, 67)* |
| **DM** | Diabetes Mellitus. *(pp. 5)* |
| **DNN** | Deep Neural Network. *(pp. 15, 19, 26)* |
| **DQN** | Deep Q-network. *(pp. 19)* |
| **DRL** | Deep Reinforcement Learning. *(pp. 9, 15, 16, 19, 20, 22–24, 38, 68)* |
| **DT** | Decision Transformer. *(pp. 17, 20, 55–57, 59, 60, 63, 64, 66, 67)* |
| **FDA** | The United States Food and Drug Administration. *(pp. 6, 13, 18)* |
| **GP** | Gaussian Process. *(pp. 42, 43)* |
| **HBGI** | High Blood Glucose Index. *(pp. 35)* |
| **IDR** | Insulin Delivery Rate. *(pp. 18)* |
| **IF** | Insulin Feedback. *(pp. 18, 43, 47)* |
| **k** | Kilo, $10^3$. *(pp. 63, 64, 67)* |
| **LBGI** | Low Blood Glucose Index. *(pp. 35)* |
| **LSTM** | Long Short Term Memory Network. *(pp. 18, 26)* |
| **M** | Mega, Million, $10^6$. *(pp. 63, 64, 67)* |
| **MDP** | Markov Decision Process. *(pp. 8, 15, 16, 23)* |
| **mg** | Milligram. *(pp. 24, 33, 37, 50, 55, 57, 58, 67)* |
| **ML** | Machine Learning. *(pp. 7, 14–18)* |
| **MPC** | Model Predictive Control. *(pp. 6–8, 18)* |
| **OF** | Observation Frequency. *(pp. 9, 40, 42–45, 47, 50, 69)* |
| **PID** | Proportional-Integral-Derivative. *(pp. 6–10, 14, 18–20, 31–36, 38, 40–45, 47, 50, 54, 56, 65, 67–69)* |

| | |
|---|---|
| **PID-Har** | Proportional Integrative Derivative with Harrison-Benedict Meal Generation Algorithm. *(pp. 44, 45)* |
| **PID-IF** | PID-Har with Insulin Feedback. *(pp. 20, 32–34, 38, 42–45, 47, 50, 55–59, 66, 68)* |
| **PID-OF** | PID-IF with personalized observation frequency. *(pp. 45, 47, 50)* |
| **PLGS** | Predictive Low Glucose Suspend System. *(pp. 6, 18)* |
| **POMDP** | Partially Observable Markov Decision Process. *(pp. 8, 15, 16, 22, 23, 28, 38, 70)* |
| **PP55** | Mixed dataset of Proximal Policy Optimization and Proportional Integrative Derivative with Insulin Feedback with five to five ratio. *(pp. 59)* |
| **PP82** | Mixed dataset of Proximal Policy Optimization and Proportional Integrative Derivative with Insulin Feedback with eight to two ratio. *(pp. 59)* |
| **PPO** | Proximal Policy Optimization. *(pp. 9, 16, 19, 26–28, 30, 31, 34, 56–59, 61, 65, 67, 69)* |
| **PPO-RNN** | Proximal Policy Optimization with Recurrent Neural Network. *(pp. 20, 26, 27, 30, 33–37, 55, 56, 58, 68)* |
| **RI** | Risk Index. *(pp. 35, 36, 60, 64)* |
| **RL** | Reinforcement Learning. *(pp. 7–10, 12, 15–21, 30, 38, 39, 55–58, 65, 67, 69–71)* |
| **RNN** | Recurrent Neural Network. *(pp. 15, 16, 18, 26, 28, 31, 38, 69)* |
| **SABR** | Simulation-Augmented Batch Reinforcement Learning. *(pp. 20)* |
| **SAC** | Soft Actor-Critic. *(pp. 9, 16, 19, 26–28, 30, 31)* |
| **SAC-RNN** | Soft Actor-Critic with Recurrent Neural Network. *(pp. 26, 27, 30)* |
| **SH** | Single Hormone. *(pp. 12)* |
| **T1D** | Type 1 Diabetes Mellitus. *(pp. 5, 8–14, 18–20, 24, 31, 35, 41, 55, 65, 68, 69)* |
| **T1D-VPP** | Type 1 Diabetes Virtual Patient Population. *(pp. 12, 23)* |
| **T2D** | Type 2 Diabetes Mellitus. *(pp. 5, 18, 31)* |
| **TD3** | Twin Delayed Deep Deterministic Policy Gradient. *(pp. 19)* |
| **TD3-BC** | Twin Delayed Deep Deterministic Policy Gradient with Behavioural Cloning. *(pp. 20)* |

**TIR** Time in Range. The target glycemic level range between 70 and 180 mg/dL. *(pp. 40, 43, 47, 50, 54, 55, 57, 64, 66, 67)*

**TPE** Tree-structured Parzen Estimator. *(pp. 42, 43)*

**Traj-PID-IF** Trajectory Transformer with the dataset generated from Proportional Integrative Derivative with Insulin Feedback. *(pp. 55, 58, 59)*

**Traj-PPO** Trajectory Transformer with the dataset generated from Proximal Policy Optimization. *(pp. 55, 58, 59, 61, 63, 65, 66)*

**TT** Trajectory Transformer. *(pp. 17, 20, 55–57, 59, 60, 63, 64)*

**UVA/PADOVA** University of Virginia and University of Padova. *(pp. 12, 13, 22, 41, 56, 68)*

# 1
# Introduction

## 1.1. The problem of diabetes management

Diabetes mellitus (DM) is a disease associated with abnormally high levels of blood glucose (BG) due to lack of insulin (type 1 diabetes - T1D) or insulin resistance (type 2 diabetes - T2D). In 2019, approximately 463 million adults worldwide were suffering from DM, which is increasing continuously [1]. More than 1.1 million children and adolescents are living with type 1 diabetes. In addition, there are 4.2 million deaths caused by DM.

T1D is an autoimmune system disorder involving the destruction of liver $\beta$ cells of the pancreatic islets of Langerhans due to insulin deficiency. Without enough insulin, glucose cannot enter the cells to transform it into energy. People with T1D need to monitor their BG levels regularly and take insulin to keep their blood sugar levels within a normal range. *Higher (hyperglycemia)* or *lower (hypoglycemia)* blood glucose levels can cause serious health problems. On the other hand, low blood glucose levels can lead to short-term complications, such as drowsiness, shakiness, confusion, loss of consciousness, seizure, or even coma or death [2, 3]. On the other hand, too little insulin can result in *hyperglycemia*, that is, high blood glucose levels can cause long-term chronic diseases, including retinopathy, nephropathy, and neuropathy [3]. Thus, people with T1D must monitor their blood glucose levels and inject insulin to prevent them. There are several insulin delivery methods both manual and automated. The usual insulin delivery method to manage glucose levels is the basal-bolus (BB) regime, which involves taking insulin before meals and at bedtime. A continuous Glucose Monitor (CGM) is a device that measures human plasma glucose levels in real-time. A CGM typically consists of a small sensor that is inserted under the skin, a transmitter that sends the data to a receiver or smartphone, and an application or other interface that displays the glucose levels in real-time. This device help patients monitor their glucose levels, but even combined with a CGM, the disadvantage of BB is the need for manual injection several times per day, especially for children when they are at school [4].

## 1.2. Current methods and systems in diabetes management

Several methods for automated glucose control [5] have been developed. These control algorithms can be classified into: (1) open-loop controls, which require patient intervention and/or external information, such as meal or exercise announcement; and (2) closed-loop controls, which do not require the patient intervention to regulate the dosage [6] but some external information can be useful in avoiding rapid BG growth [7]. In this work we consider a restricted definition of closed-loop controller in which any information that cannot be automatically passed to the controller and requires the intervention of the user is not a closed-loop controller, a point of view shared by similar works [8, 6].

An artificial pancreas (AP) is a medical device designed to automate the management of insulin delivery for people with type 1 diabetes. The AP system consists of a CGM device that measures glucose levels in real-time and a insulin pump that delivers insulin based on the glucose readings. The device operates using advanced control algorithms that automate the decision making process of insulin delivery, mimicking the functions of a healthy pancreas. The currently available AP systems for controlling blood glucose levels in devices rely on PID and MPC control algorithms, as mentioned in the previous works [7, 9, 10, 11, 12, 13, 14]. A PID controller adjusts insulin release to maintain stable blood glucose levels through proportional, integral, and derivative components which consider current error, accumulated error over time, and future errors based on the rate of change, respectively. On the other hand, MPC controllers predict blood glucose levels, are more proactive than PID controllers but they require a minimal compact mathematical model to perform well. MPC controlllers use nonlinear differential-difference equation models to accurately account for the endogenous insulin delivery rate. These algorithms work as a hybrid closed-loop system and require input from the user regarding carbohydrate intake and physical activity [10]. FDA-approved systems, such as the Metronics 670G and 770G, which use PID [9, 10, 15], and Tandem Control-IQ, which uses MPC [9, 10], are commercially available. To prevent hypoglycemia overnight, most commercial products employ Predictive Low Glucose Suspend (PLGS) technology [10], which predicts glucose concentration trends and suspends insulin delivery prior to hypoglycemia.

## 1.3. Literature review

Several methods for closed-loop controls can be found in the literature. As mentioned previously, the utilized control algorithms usually are predictive integral derivative controllers (PID) [16, 11, 12, 17, 18] and model predictive controllers (MPC) [19, 20, 13, 14, 21] and expert-based (fuzzy logic) approaches [22]. In brief, PID, one of the most effective and widely used solutions both in commercial and research because of its simplicity and robustness [7, 23]. PID use previous BG samples as feedback to determine the insulin needed to drive the desired glucose concentration in human blood. Their main disadvantage is a poor adaptation to meal disturbances [8, 24] and inability to individual treatment. MPC requires a mathematical model to predict future glucose concentration using current BG, insulin delivery and meal intake; then the algorithm calculates the appropriate insulin infusion rate by minimizing the difference between estimated glucose concentration from the model and the target glucose concentration on a prediction time window [19]. These methods depend obviously on the quality of the model and are also sensitive to external disruptions such as food intake or physical activity that cannot be accurately modeled [13, 14]. Expert-based approaches implement case-based logic using experience from medical experts to decide when and how much insulin to deliver [22]. The model requires an expert to create, adjust and evaluate the model, which can lead to human errors. Also, existing empirical models of a patient metabolism cannot be applied to these approaches, hence there are no theoretically based performance guarantees.

Recently, machine learning (ML), including reinforcement learning (RL), has gained attention in diverse domains such as finance, robots, computer vision, language recognition as well as medicine and healthcare. ML predictive models can be applied to time series data to understand changes in glycemic state and determine the amount of insulin to deliver. Among them, Artificial Neural Networks (ANNs) involve machine learning algorithms that have been already used for diabetes purposes. They need labeled training data from experts to predict blood glucose concentration based on supervised learning and avoid human error. ANNs perform well for short-term prediction [25]. However, improving the prediction of supervised learning approaches implies high volumes of training labeled data and there still remains the problem of designing an appropriate controller from the predicted BG level. RL has been suggested as a more promising alternative [6, 8]. RL is a branch of ML that lets the agent learn by interacting with the environment, usually an artificial patient in a simulation [6]. In RL, a software agent makes observations and takes actions within an environment and receives rewards from its actions. By appropriately shaping the reward function, the agent can self-learn the desired goal. Their main advantage is that they are model-free, up to some extent since the environment provides the implicit model, and can learn latent disturbances and adapt to them. The RL agent gathers rewards from outcomes of the agent's action, which are used to learn to take better decisions. Thus, RL algorithms can use physiological data gathered from CGM systems to train the agent. We evaluate and discuss this RL process, called *online RL*, in Chapter 3.

## 1.4. Issues/gaps/challenges

Even though different RL approaches have been increasingly proposed and discussed [6, 8, 24], effective training of agents for BG control has proved to be difficult [8, 26, 27, 20]. Several factors may explain the difficulties. First, most of the RL algorithms are designed to approximately solve a Markov Decision Process (MDP) with a fully observable state space [28], but realistic simulators of the glucoregulatory systems cannot be considered fully observable. Therefore, the environment is at most a Partially Observable MDP (POMDP) and, in spite of their ubiquity in many fields, only very few recent algorithms have been developed specifically for POMDPs [29].

Moreover, POMDPs require a mapping from true environment states to observable variables that have to be defined by the control algorithm designer [6], usually with a high degree of arbitrariness. This may have a serious influence on the learning ability of the agent because a bad choice of observable states makes state changes and associated rewards not directly related to agent actions. For instance, a delayed response to insulin is a realistic feature of a simulator but it is also related to the choice of POMDP state mapping, because a CGM reading (observation) after an insulin injection (action) does not reflect the actual change of state. Second, compared to other learning environments, there is a remarkable number of design alternatives whose influence is not clear and usually require careful trial and evaluation. Those involve the choice of the RL algorithm (agent) and its underlying neural network architecture, the tuning of agent hyperparameters, the selection of an appropriate reward function, and even the design of the action space that may be adapted to patient specific data [8, 27]. For the sake of conciseness, the design choices related to the aforementioned issues are called implementation strategy from now on.

From the above discussion, it is clear that different implementation strategies can lead to effective RL-based closed-loop controls. They may result in a viable controller or not, and with widely different performance, so the implementation strategy is subject to further investigation, as it is addressed in Chapter 3. In fact, straightforward implementations, as we discuss and show in the following sections, do not work properly. Therefore, the RL approach to control is not different to the other main approaches to the control problem, PID and MPC, in the sense that it can be considered a generic approach, with many potential different implementations which are proposed and evaluated [13, 14]. In Chapter 3, we evaluate two different reinforcement learning algorithms to control *in silico* blood glucose levels in T1D and compare them with other well-known alternatives, including a PID controller. We describe and discuss our implementation strategy and related problems and compare it with recent proposals using a different implementation strategy [8, 27].

## 1.5. Proposal and hypothesis

Our work in Chapter 3 shares with other proposals [26, 8, 27, 20] the initial premise: to train state-of-the-art Deep Reinforcement Learning (DRL) algorithms, such as Soft Actor-Critic (SAC) [30] or Proximal Policy Optimization (PPO) [31], as a T1D BG closed-loop control, but it differs in several aspects of the implementation strategy that we discuss in Chapter 3. This process is called online RL in the remaining of the thesis.

In Chapter 3, we investigate the impact of observation frequency (OF) on the performance of reinforcement learning algorithms for blood glucose control in T1D and we find that implementing OF significantly improves the performance of the PPO algorithm. Given these promising results, it is natural to ask whether this improvement can be extended to other control methods, such as the widely-used PID controller. Thus, in Chapter 4, we explore the effect of OF on the performance of the PID controller for blood glucose control. This is particularly relevant, as PID control is often used in practice due to its simplicity and robustness. By investigating the impact of OF on PID control, we hope to gain a better understanding of how this parameter can be tuned to improve blood glucose control in T1D.

Finally, online RL requires extensive trial and error interaction with the environment, which is the real patient in this case, something that is obviously not possible. Therefore, online RL has been so far successfully used to automatically control BG [32, 8] but only in *in silico* tests and there is no clear way of bringing it to clinical trials because of the high risk involved when working on real patients. In contrast, *offline RL* [33], a recent approach, could solve that problem. Offline RL requires only pre-obtained data to make an agent learn a policy for a particular environment. This data can come from real measurements taken from patients. Thus, this approach does not involve actual interaction with the environment (patient) during the training phase. The suitability of offline RL for BG control has only been started to be discussed in the literature [18]. To cover this gap, in this thesis, we evaluate the use of offline RL as a method for effective BG control, demonstrating its potential and discussing its shortcomings. A goal of the work in Chapter 5 is to determine whether offline RL can be a realistic alternative for data-driven BG control, before attempting clinical trials with real patient data.

We have selected two recent offline RL algorithms, Trajectory Transformer [34] and Decision Transformer [35], and evaluated its performance compared to online RL and PID baselines. But, in addition to the algorithm, there are a number of factors that have influence on the ability of offline RL agents to learn, such as the size and quality of the datasets used for training. So we have extensively evaluated this aspect by: trying different dataset sizes, using two types of datasets and mixing them and selecting the best subset of the past experiences (trajectories) that lead the RL agent to learn certain behavior.

Therefore, the initial hypothesis of this thesis is that reinforcement learning methods can be an effective method to automatically control BG levels in T1D patients. In the remaining of the thesis we evaluate this hypothesis.

## 1.6. Objectives

The general objective of this thesis is to design and evaluate more effective control algorithms for artificial pancreas systems used by patients with T1D. The focus will be on improving diabetes management through the use of a closed-loop system that uses data from both a CGM and an insulin pump. To achieve this objective, the following sub-objectives will be addressed:

1. Analyze the existing problems and challenges in the current artificial pancreas systems for T1D patients.
2. Investigate the feasibility of using online and offline RL algorithms to address these challenges.
3. Eliminate the necessity for online RL to interact with patients by leveraging offline RL and pre-collected data.
4. Evaluate the performance of different RL algorithms in controlling blood glucose levels.
5. Compare the performance of these algorithms with traditional control methods, such as BB and PID, to determine their effectiveness in improving diabetes management.

These sub-objectives will help the research to address the main objective by providing a comprehensive evaluation of the use of RL in artificial pancreas systems, and by comparing its effectiveness with traditional control methods. The results of this research will contribute to the development of better and more effective control algorithms for artificial pancreas systems, improving diabetes management for T1D patients.

## 1.7. Thesis organization

This thesis is organized as follows:

**Chapter 1: Introduction**. The background of the research, the motivation, and the problem statement are presented. The aim of the research and its significance in the field of diabetes management are also discussed in detail.

**Chapter 2: Background and Related Work**. It is focused on the blood glucose control issue in patients with T1D and presents comprehensive examination of the current advancements in glucose regulation for these patients.

**Chapter 3: Deep Reinforcement Learning for Blood Glucose Control**. An evaluation of the use of deep reinforcement learning for controlling blood glucose levels in T1D patients is presented. A meticulous discussion of the research's techniques and results is carried out.

**Chapter 4: Optimization of PID parameters for blood glucose control**. The optimization of the PID (Proportional Integral Derivative) method for blood glucose control is presented as a baseline for comparison with other algorithms in the research. A comprehensive overview of the optimization process and its results is provided, which will serve as a benchmark for evaluating the performance of alternative methods.

**Chapter 5: Offline Reinforcement Learning for Blood Glucose Control**. The examination shifts to evaluating the effectiveness of offline reinforcement learning in regulating blood glucose levels for patients with T1D. A detailed account of the techniques and results from the research is presented.

**Chapter 6: Conclusion and Future Work**. The main findings of the research are summarized and the limitations and challenges of the methods proposed are discussed. Finally, future lines of research are proposed to continue advancing in the field of diabetes management.

# 2

# Background and related work

The aim of this chapter is to provide a comprehensive understanding of the current state-of-the-art in blood glucose control for patients with T1D and to examine the related work in this field. In addition, the chapter will provide an overview of the current advancements in glucose control algorithms and their limitations. This information will provide a foundation for the research carried out in the later chapters of this thesis. The background information provided in this chapter will be essential in understanding the need for this research and the potential impact that the results may have on the field of T1D management.

## 2.1. T1D simulation and models

For safety reasons, biomedical experiments with machine learning algorithms have been done and pre-evaluated *in silico* through computer simulation. Currently, there are several T1D simulators available, with both free and paid versions, as for instance, AIDA [36], Type 1 Diabetes Virtual Patient Population (T1D-VPP) [37], Dosing-RL Gym [38], and the UVA/PADOVA Simulator [39]. AIDA is a free software simulating human plasma insulin and blood glucose for education and research purposes. T1D-VPP involves single (SH) and dual hormone (DH) mathematical models which generate a T1D diabetes virtual population of patients and model the effect of exercise in the glucoregulatory system. Dosing-RL Gym is based on an expanded version of the Bergman minimal model, which includes meal disturbances [38].

The UVA/PADOVA Type 1 Diabetes Simulator can be used as a substitute for preclinical testing of closed-loop control strategies. The simulator was developed in 2007 by the Universities of Virginia (UVA) and Padova and has been approved *in silico* T1D model by the United States Food and Drug Administration (FDA) [39]. It is the most used simulator among *in silico* software, according to [6] and [40]. There are four main components of the simulation, which are depicted in Fig. 2.1: (1) *In silico* patient – a model of the glucose-insulin system in a patient; (2) *In silico* sensor – a model of the sensor to measure BG including its error; (3) Controller – a model used to estimate the amount of insulin to maintain blood sugar; and (4) *In silico* pump – a model of discrete insulin delivery and subcutaneous kinetics. In this thesis, we use *SimGlucose*, an open-source Python implementation of the UVA/PADOVA simulator [41], previously used in similar studies [6, 8, 42, 43]. The simulation environment implements the OpenAI gym interface [44], which makes it can be seamlessly integrated with multiple machine-learning libraries. The simulator provides virtual patients in three age groups: adults, adolescents, and children, with 10 patients per group in the free version. It also simulates different noisy CGM sensors, insulin pumps and a random meal scheduler.

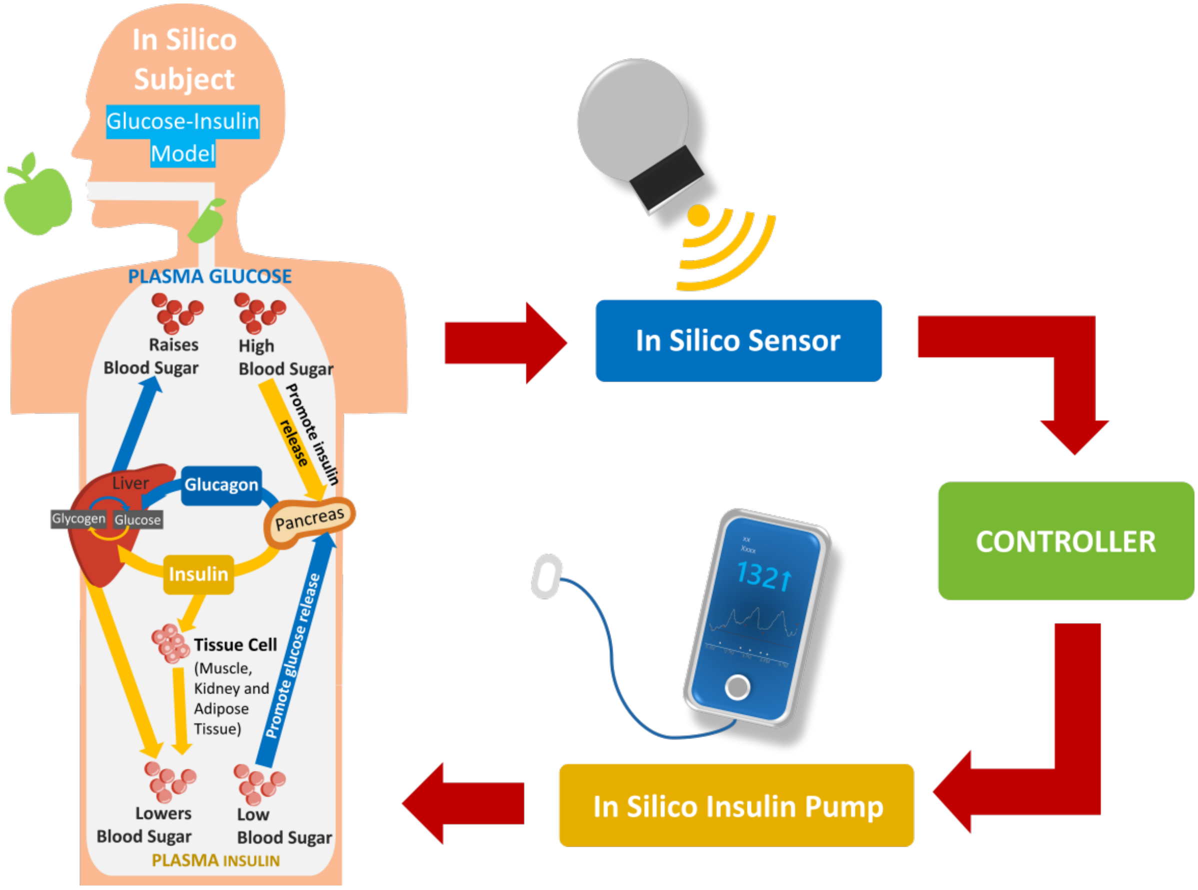


**Figure 2.1:** Principal components of the computer simulation environment.

## 2.2. Methods for glycemic regulation

T1D conditions typically develop in children or young adults and require lifelong treatment with insulin injections. Several insulin regimes are used to control blood sugar. The traditional ones involve one or two injections per day. But patients must control their food intake to be constant throughout the three meals a day. Multiple daily injection therapy, or basal-bolus (BB), offers more flexibility in diet and dosage, but patients still need to control carbohydrate intake and insulin injections [45]. Automatic insulin pumps with integrated continuous glucose monitors (CGMs) have been developed to alleviate the burden of glycemic control and deliver optimal insulin according to current blood glucose levels, allowing patients to live independently without having to worry about delivering insulin. A system that does not requires any human intervention is usually called a closed-loop controller. Such a system is also called an Artificial Pancreas (AP).

Currently, most of the commercially available insulin pumps use a PID (proportional-integral-derivative) algorithm to control blood sugar levels. A PID controller is a control system that uses feedback to adjust a system's output in order to achieve the desired outcome. In the context of blood glucose control, a PID controller is used to regulate the release of insulin in order to maintain a stable blood glucose level [10]. The proportional component of the PID controller adjusts the output based on the current error between the desired and actual blood glucose levels, while the integral component considers the accumulated error over time and the derivative part predicts future errors based on the current rate of change. By combining and tuning these three components, a PID controller can control blood glucose levels, but they usually have problems to adapt to disturbances in food intake and need to be customized to individual patients [8, 24]. The general concept of a PID control system can be mathematically represented as a linear combination of three terms:

$$a_k = K_p P(s_k) + K_i I(s_k) + K_d D(s_k) \tag{2.1}$$

where $P(s_k) = s_k - s_t$, $I(s_k) = \sum_{i=0}^{k}(s_i - s_t)$ and $D(s_k) = |s_k - s_{k-1}|$.

Therefore, there is a target value, $s_t$, and three parameters ($K_p$, $K_i$, and $K_d$) which can be tuned to achieve the desired control in the proper way.

## 2.3. Machine learning in BG regulation

ML is gaining momentum in AP research [46, 47, 48]. ML algorithms can be used theoretically in the field of blood glucose control to develop systems that are able to automatically regulate blood glucose levels according to the individual needs. As other data-driven methods, the idea is to collect labeled data from CGMs and other devices and train a ML model. Through the training process, ML algorithms would ideally identify patterns and trends in order to learn how to predict blood glucose (BG) levels and adjust insulin levels accordingly. At this point, there are several alternatives. The first one is to use the ML model to just predict the expected BG level ahead of time and then use some other method to decide the insulin dose required to keep BG at the desired level [48]. However, the human response to insulin is highly non-linear and it is also difficult to predict the response to the insulin injection. Therefore, another alternative is to learn with ML methods that response such as Reinforcement Learning (RL) [8, 27, 6].

## 2.4. Reinforcement learning in T1D management

Reinforcement Learning (RL) refers to a class of methods to optimize the decision process of an *agent* operating on a given *environment*. The decision process is usually considered to be a *Markov Decision Process* (MDP), a discrete-time stochastic control process. Formally, an MDP is defined by a 4-tuple $(S, A, P, R)$ of states, $s \in S$, actions, $a \in A$, a state-transition function $Pa(s, s') = Pr(s_{t+1} = s'|s_t = s, a_t = a)$, and reward function $R_a(s, s')$. The agent is the learner entity that seeks the optimal behavior and is able to perform an action $a(s)$, which changes the state. In this change of state from $s$ to $s'$, the agent obtains a reward $r$, considered as the feedback from the environment. MDP-solving algorithms employ what is called a policy, denoted as $\pi$, which is a mapping between states and actions; that is $\pi : s \to a$. Their goal is to reach an optimal policy $\pi^*$, which maximizes the accumulated sum of rewards over the entire lifespan of the agent. This decision policy can be determined by the state-action function, also called Q-function, $Q(s, a)$, which can be approximated using Deep Neural Networks (DNN). Deep Reinforcement Learning (DRL) refers to algorithms and methods that use DNN to approximate the Q-function or optimal policy.

It is commonly assumed that the MDP has a fully observable state space S, that is, that the agent has access to observations that fully represent the state of the environment. However, the observation may just be a partial representation of the underlying state. A Partially Observable Markov Decision Process (POMDP) is an extension of an MDP, where the agent cannot fully observe the system state. In that case, the MDP 4-tuple is extended with a space of observations, $o \in O$, and a usually unknown and potentially stochastic function that maps the observations to true underlying states, $T : o \to s$. Partial observability may stem from many factors, including limited sensing capabilities or unknown environment dynamics [29]. Let us remark that the agent with partial observability cannot know which is the real state corresponding to the reward received [49]. Despite the ubiquity of POMDPs in many practical systems [29], most of the DRL algorithms assume an underlying MDP [28, 31]. POMDPs are usually addressed in DRL by augmenting the observation space with the history of past observations and actions and the use of Recurrent Neural Networks (RNN) [29] in the architecture of the learning algorithm. Only recently, a few algorithms have been specifically designed to deal with POMDPs [29]. Moreover, the state transition in some environments is determined not only by agent actions, but also by exogenous stochastic input actions [50]. Efficient methods to deal with this kind of environments are discussed by Mao *et al.* [50]

Most of the state-of-art DRL algorithms used for MDPs are based on actor-critic methods: temporal difference learning algorithms that separate representations of value functions and policies explicitly [28]. The actor selects actions in the action space, while the critic estimates the value function from the action made by the actor. They can be applied to either discrete or continuous action spaces. However, these methods show poor sample efficiency and stability convergence properties. A variety of techniques have been developed to address those problems [51, 28, 31].

In this thesis, we have used the folowing particular online RL algorithms:

**Soft Actor-Critic (SAC)**, a widely used continuous-state DRL algorithm [30, 8], whose policy maximizes a trade-off between the expected return and entropy, a measure of randomness in the policy, which ensures higher robustness and stability [28]. Maximum entropy policies have been shown to solve POMDP with unobserved rewards [49].

Besides SAC, we have used **Proximal Policy Optimization (PPO)**, another popular DRL algorithm [31]. PPO ensures that its policy does not change much from the previous policy updates, leading to smooth learning and avoiding variance in training. The tradeoff between SAC and PPO is stability and sample efficiency.

PPO tends to be more stable and uses more data, whereas SAC tends to be the opposite. PPO is also claimed to work well on POMDPs [52]. Both allow the use of RNNs in their architecture.

## 2.5. Offline reinforcement learning in T1D management

However, the main drawback of DRL approach is that it is not clear at all how to apply it to real patients, that is, how to transfer the learning from the *in silico* environment to real patients. Although data (BG level, physical activity, etc.) can be automatically collected from real patients from electronic devices, RL agents still need to experiment with the patient response in order to learn.

To solve this issue, a more recent approach, called *offline reinforcement learning*, has emerged. In offline reinforcement learning, the agent is not able to receive any feedback from its environment during the learning process, and must instead learn only from previously collected data [33]. This means that the agent must learn to make decisions based on the information that is available, without being able to receive any new information or adjust its actions based on its current situation. Note that the main difference with other ML methods is that with offline RL the actions and rewards are also given as input data. For example, a typical supervised ML algorithm uses collected BG levels (as well as other context data) to train and is able to predict the next BG level, given a certain input BG history. On the contrary, to train an offline RL agent we need to use BG levels, actions taken and observed rewards, and, once trained, it is able to predict the required action, given a certain BG history as input.

The advantage is that it is useful in situations where it is not possible or practical to experiment with the environment, such as when working with historical data or in safety-critical environments. As a drawback, note that, although it removes the need to interact with the environment to learn, it still leaves open the question of how to collect the required states, actions and rewards for training, which is not obvious for many practical situations. In Chapter 5, since we can collect those data from simulations, we put aside temporarily this question and focus on evaluating how effective is offline RL for BG control. Let us remark that the value of offline RL is that it is able to effectively generalize, that is, to apply the appropriate action to an input not previously seen in the training dataset. In other contexts, ML has proved to be very effective generalizing [53], but to the best of our knowledge, the generalizing performance of offline RL for BG control has only been started to be discussed in the literature [18]. Our goal in Chapter 5 is to evaluate it and discuss factors that may have an influence in the learning and prediction performance.

In particular, we evaluate the following offline RL algorithms:

**Decision Transformer (DT) [35].** In RL, a trajectory representation is typically a sequence of states (s), actions (a), and rewards (r). DT, instead of reward, uses return-to-go to feed as an input. Return-to-go describes as future desired returns $\hat{R}_t = \sum_{t=1}^{T} r_t$. So the DT trajectory representation is $\tau = \{\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, ..., \hat{R}_T, s_T, a_T\}$. The input of DT is a subset of the trajectory $\tau$ consisting of the $K$ most recent time steps which allow the previous values to be taken as input for long history.

**Trajectory Transformer (TT) [34].** The trajectory representation of TT is just slightly different from DT as $\tau = \{s_t^1, s_t^2, ..., s_t^N, a_t^1, a_t^2, ..., a_t^M, r_t\}_{t=0}^{T-1}$. TT use discretized states and actions as input as well as a scalar reward. But, for planning, that is offlineRL, it also augments the trajectory with a return-to-go as DT and uses a beam search algorithm [54].

Both DT and TT uses as architecture for action prediction a transformer network. The transformer is a type of deep learning model that is designed to process sequential data which was introduced by Vaswani *et al.* in 2017 [55]. The transformer architecture is based on the idea of using self-attention mechanisms to process input data, rather than using traditional convolutional or recurrent layers. This allows the model to capture long-range dependencies in the data and to process the input sequence in parallel, which makes it faster and more efficient than many other types of models. A key aspect determining the performance of offline RL algorithms is the quality of the datasets used for training. In fact, their performance is usually validated separately according to the quality of the trajectories included in the dataset. For instance, the quality of the dataset can range from randomly (random dataset) generated trajectories to trajectories generated by the best-performing algorithm (expert dataset) or a mixture of them [35, 34].

## 2.6. Related work

### 2.6.1. Common methods in blood glucose control

State-of-the-art control algorithms for AP systems on devices already on the market are based on PID and MPC approaches [7, 9, 10, 11, 12, 13, 14]. PID and MPC controllers usually work as hybrid closed loop system, requiring announcements of meal carbohydrates amount and exercise activity [10]. Two commercially available FDA-approved systems are Metronics 670G and 770G using PID [9, 10, 15] and Tandem Control-IQ using MPC [9, 10]. With PID and MPC, most commercial products avoid hypoglycemia overnight by utilizing Predictive Low Glucose Suspend (PLGS) [10]. PLGS technology predicts glucose concentration trends, then suspends insulin delivery before hypoglycemia occurs.

A PID controller is simple but have problems to adapt to meal consumption [8, 7]. Several variations to the basic PID approach have been proposed, as the use of insulin feedback (IF), which improves its performance [12, 11]. MPC controllers are more proactive than PID in insulin delivery by predicting BG levels, but they need a minimal compact mathematical model. Di Ferdinando *et al.* [13] and Borri *et al.* [14] use nonlinear differential difference equation (DDE) models for the endogenous insulin delivery rate (IDR), which is better accounted for in these models. Since the IDR cannot be neglected for T2D patients, and the DDE model reproduces it accurately, MPC that use DDE usually address T2D. Their results show that MPC provides good performance as long as a minimal compact model is available. However, as the complexity of the model increases, MPC approaches are not tractable and one has to resort to other control methods.

### 2.6.2. Machine learning in blood glucose control

Among these methods, the number of data-driven models for prediction of BG in T1D is increasing [40]. ML has been used as a tool for the prediction of diabetes [46, 47, 56, 48], but also for glycemic control in an insulin pump, and such techniques are growing rapidly within the artificial pancreas research community. Most ML experiments are done *in silico*, through computer simulation. As CGM data are time series, non-linear autoregressive neural networks are used for BG prediction in [37], while [57, 58, 59, 60, 61] use recurrent neural networks (RNNs) and long short-term memory (LSTM).

### 2.6.3. Reinforcement learning in blood glucose control

Reinforcement learning is being used in recent research works in the field of health care. For instance, the RL Q-Learning algorithm was applied on discrete action space simulation for radiotherapy to understand scenarios of tumor growth and its treatment plan [62]. RL agent-based models have been used on continuous state and action spaces problems to find cytokine therapy for sepsis, reducing mortality from 49% on average to 0.8% [63]. RL is suitable for time sequence problems, as in the glucoregulatory system. Furthermore, the agent can learn the policies without the need for labeled data as in supervised learning methods [6, 24]. The ability of RL to capture food intake patterns without human input makes it a good candidate for a fully closed-loop system and more responsive and safer policies [8]. Other works that use reinforcement learning in glycemic control include: double score strategy [64], Q-learning[65, 66, 67], Deep Q-network (DQN) [68], Deep Deterministic Policy Gradient (DDPG) [69] and its improvement Twin Delayed DDPG (TD3) [70], Soft Actor-Critic (SAC) [8, 27] and Proximal Policy Optimization (PPO) [32]. These RL methods are called online RL, since the agent interacts with the environment to collect data.

Particularly about T1D control, a recent review discusses most of the approaches used so far in this topic [6]. This review exposes the wide variety of alternatives used in almost all the defining elements of the RL framework, such as the definition of the state space, the action space, class of RL algorithms used and the reward function, what we have called the implementation strategies. To mention a few which differ in the class of RL algorithms, in [24] a control based on RL is proposed and the value function is not approximated by a DNN, but by a quadratic function, and in [71] the value function is approximated by a Gaussian process. In both cases, robust solutions are provided, but simplified glucose models are used. On the contrary, in Chapter 3 we use a more realistic simulator which generally requires DRL to approximate the value function.

Fox *et al.* adopt an approach similar to the one adopted by us in Chapter 3 in two recent papers [26, 8]. In fact, they share the basic premise with ours in Chapter 3: training a DRL agent for BG closed-loop control. However, they employ different implementation strategies. In their first one [26], the state space comprises the previous 24 hours of CGM samples and insulin doses at 5-minute intervals, but the action space is discrete and made of only three insulin doses. Three relatively simple DNN architectures were used to approximate the value function and the improvements over the PID baseline were not particularly noticeable. In their second work [8], the state space is also made of CGM and insulin samples but only from the four previous hours and the action state is continuous, so the SAC algorithm is used. Additionally, the reward functions differ in both cases. In contrast, in this work we use only the last CGM sample as state but take a relatively long interval of 30 to 60 minutes between observations and actions and define a simpler reward function. A hybrid model-based approach is discussed by Yamagata *et al.* [20], which uses a discrete action space combined with meal announcement.

Recently, Lim *et al.* [27] proposes a combination of machine learning methods for BG control: the controller uses a DRL SAC agent which is driven by a PID control as an initial policy and, in addition, the observation state is extended by the predictions of a dual attention network. Finally, the actions are also regulated by an adaptive safe action. The results of the last three aforementioned methods [8, 20, 27] are further compared with ours in the Discussion section in Chapter 3.

### 2.6.4. Offline reinforcement learning in blood glucose control

Our work in Chapter 3 shows a simple RL implementation strategy that outperforms PID with insulin feedback for BG control in *in silico* tests. However, as mentioned previously, online RL is not suitable for safety-critical environments, where interaction with the environment (the real patient) is not possible. Therefore, recently, researchers have paid more attention to offline RL. Offline RL is similar to online RL, but the offline RL agent does not need to interact and receive any new information from the environment during the learning process [33]. This means that the agent instead learns from previously collected data, which is safer and more useful for medical and healthcare research. Only a few works have evaluated the use of offline RL for BG control, such as [72], which uses Simulation-Augmented Batch RL (SABR), and [18], which applies and compares three offline RL techniques: Batch Constrained Deep Q-learning (BCQ), Conservative Q-learning (CQL) and Twin Delayed DDPG with Behavioural Cloning (TD3-BC). The work of Fox demonstrates how offline RL can reduce risks over two months and two years of evaluation. The work of Emerson *et al.* shows that TD3-BC outperformed PID across all patients.

## 2.7. Proposed solution

In this thesis, we propose a closed-loop glucose level control approach based on Deep Reinforcement Learning. We examine the unique features of a realistic simulator of the glucose regulation system as a training ground for DRL algorithms, and the challenges in training these algorithms in such an environment. To overcome these difficulties, we assess several implementation strategies for the learning process and, based on the evaluation results, suggest a specific strategy that involves reducing the frequency of observations and rewards, and using a straightforward reward function. Our proposed approach was applied to three patient groups using PPO-RNN agents, which were trained using the chosen strategy, evaluated, and compared with traditional control methods such as PID, PID with insulin feedback PID-IF, BB, and BB with cooldown BB-CD.

In addition, one of the major drawbacks of online RL is that it requires continuous interaction with patients and frequent updates to the model, something that is not possible to do safely at the moment. Offline RL eliminates this requirement, as the learning process can be conducted entirely from recorded data, without the need for patient interaction. Therefore, we propose using offline RL in T1D in Chapter 5. This proposal is similar to the work of Emerson *et al.* and Fox *et al.*, but there are significant differences. First, we evaluate more recent offline RL algorithms (DT and TT), which have shown better results than the ones used by Emerson *et al.* Second, their work only evaluates 9 patients, 3 from each of the three group ages available at SimGlucose, while we evaluate all the virtual patient population, 30 patients. Finally, their training dataset only contains $10^5$ samples generated by PID for each patient, while our datasets contain one million sample per patient and have been generated with PID-IF and our previous online RL implementation.

As we said, for offline RL it is key to evaluate the influence of the training dataset, so we have extensively evaluated this aspect by: analyzing different dataset sizes, source of datasets, potential of mixing them, and selecting the best subset of trajectories.

# 3

# Evaluation of blood glucose level control in T1D patients using deep reinforcement learning

In this chapter, we describe and discuss the design process and choices for our implementation strategy for a BG closed-loop control based on DRL. Our goal is to balance blood glucose as long as possible with low risks. First, we define the state and action space and discuss the reward function. Afterwards, we conduct an initial evaluation based on naive strategies to determine the features of the environment that may have more influence on the agent learning. This leads us to refine our design and propose an implementation strategy that is evaluated in Section 3.2.

## 3.1. Implementation strategy and methodology

### 3.1.1. Analysis of environment and initial design

The SimGlucose simulator environment implements the UVA/PADOVA glucose model [73] and provides CGM sensors that sample the BG level through a noisy (stochastic) process as well as a random and patient-dependent meal schedule. From this description, it is clear that the environment should be considered a POMDP, since the CGM observations of BG levels include noise reads from sensors. In fact, the underlying hidden states of the environment, $s$, are given by the glucose model states [73], rather than the BG level. Moreover, the dynamics of the environment, that is, the state transitions, are not only determined by the actions taken by the agent (insulin dose injected), but also by exogenous stochastic input actions [50], such as the intake of CHO (meals) or physical exercise. However, since SimGlucose does not consider physical exercise, unlike T1D-VPP [36], we restrict our study to external meals.

All the above considerations have an influence on the DRL controller design. We first define the elements of the DRL algorithm for our problem as follows:

**Observations and state:** The mapping of the observable variables may determine whether the environment is a POMDP or an MDP. As an example, the number of frames included in the observations in the Atari Pong game makes it become a POMDP or an MDP [74]. We start by using only CGM samples as observation variables, since they are readily available. Unlike the work in [8], we do not use past actions (insulin doses) in the observations. Since we target a closed-loop controller, we do not include CHO intake as part of the observation, which has to be announced by the user, even though some devices may facilitate its announcement [75]. We start by using only the current observation, given by the last CGM sample, $o \in \mathbb{R}^{+}$. The CGM sample frequency is three minutes per environment step.

**Action:** The action is the amount of basal insulin that the patient gets injected. It is a decimal number, ranging between 0 to 30 units, $a \in [0, 30]$, according to the specifications of the insulin pumps implemented in the simulator.

**Goals and risk metrics:** Safety is crucial in healthcare applications. The main goal of our BG controller is to balance the BG level for as long as possible with low health risks. A commonly used metric of risk associated with BG levels is the blood glucose risk index (BGRI), and it has also been used to measure the performance of control methods [76]. BGRI is a measure of glucose variability and associated risks and it is based on a symmetrization of the BG measurement scale [76]. The Clarke BGRI is defined as $BGRI = LBGI + HBGI$, where $LBGI$ and $HBGI$ are computed over a series of n CGM samples as:

$$LBGI = \frac{1}{n}\sum_{t=1}^{n} rl(BG_i) \tag{3.1}$$

and

$$HBGI = \frac{1}{n}\sum_{t=1}^{n} rh(BG_i)) \tag{3.2}$$

where $LBGI$ and $HBGI$ represent the risk associated with low and high BG levels. They are computed from the following function:

$$f(BG) = 1.509 \times [log(BG)^{1.084} - 5.381] \tag{3.3}$$

Noted that BG is measured in mg/dL. $f(BG)$ is the basis to calculate the BG risk function using the formula $r(BG) = 10 \times f(BG)^2$ and separating it as low $rl$ and high $rh$ as follows:

$$rl(BG) = \begin{cases} r(BG), & \text{if } f(BG) < 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

$$rh(BG) = \begin{cases} r(BG), & \text{if } f(BG) > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

Similar risk measures have been defined and used in the literature. For instance, in [8] the Magni risk function is used, defined as:

$$ri(BG) = 10[3.35506((ln(BG))^{0.8353} - 3.7932]^2 \tag{3.6}$$

The curves from Clarke and Magni are shown in Fig.3.1. As can be seen, the BGRI adequately captures the increased risk associated with hypoglycemia for the patients. It is common practice to define the reward function in terms of the Clarke or Magni RI [26, 8]. But, as we discuss next, they may not be adequate to capture our intended goal.

**Termination limits and safety:** In DRL, an environment finishes when some condition is met. For instance, in standard OpenAI gyms such as the BipedalWalker [77] the episode finishes when the robot falls. In SimGlucose, the episode ends when the BG level goes out of a predefined range. T1D patients should aim for a target range of 70–180 mg/dL [78]. We have configured SimGlucose to end the episode when BG $<$ 70 mg/dL or BG $>$ 350 mg/dL in order to try to avoid dangerous BG levels. Let us note that this is a quite conservative range. In contrast, in [8], episodes are done when the BG falls below 10 mg/dL or raises over 1000 mg/dL.

**Reward:** A crucial, and problematic, aspect of DRL is the need to design a reward function that helps the agent learn the intended goal [74]. In our solution, we have tried different approaches. In particular, we have tried using the negative of the Clarke BGRI as reward function, in order to keep the BG at the desired level, but in this case negative rewards induce the agent to terminate early, leading to dangerous regimes. Basically, the agent learns to inject more insulin to avoid keeping receiving negative rewards, provoking hypoglycemia. A termination penalty to correct for this behavior is usually introduced, as is done in [8]. From our point of view, this solution is not satisfactory, because the value of the reward at termination becomes effectively another hyperparameter. It requires to be tuned for the expected duration of the episodes. In fact, as an extreme case, since the desired goal is to avoid termination at all, the value should be set to infinity, or at least to a value high enough to counteract the expected lifetime of the patient. We have tried a different strategy: combined with the conservative termination limits mentioned above, we use

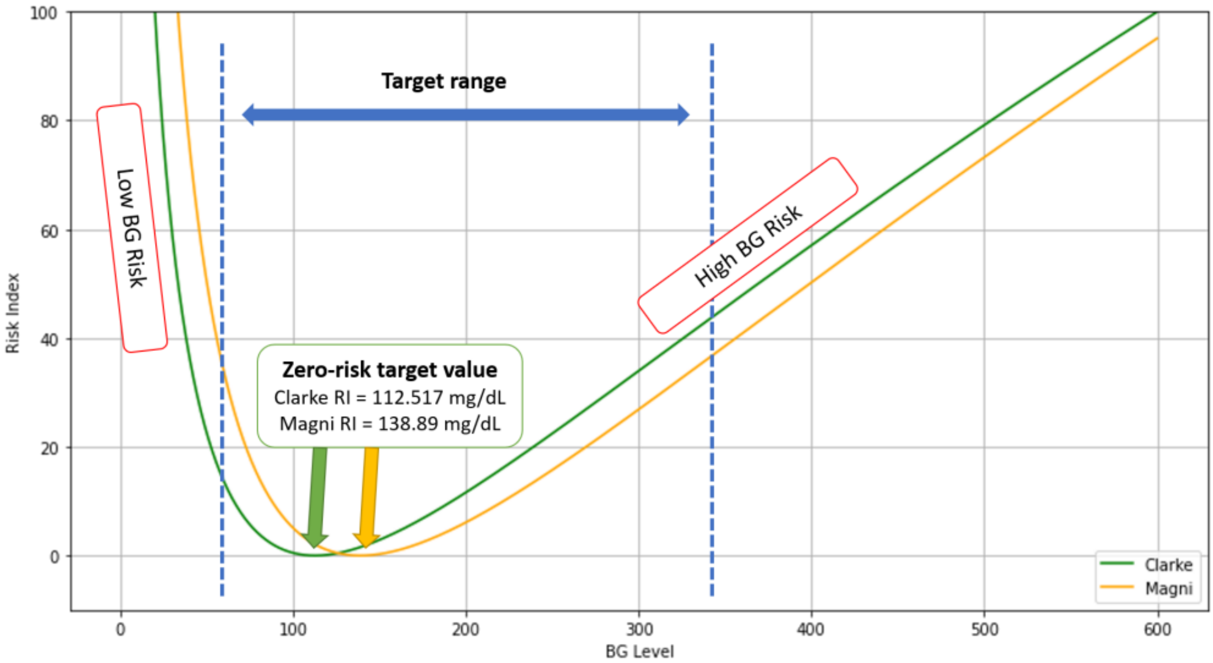


**Figure 3.1:** Blood glucose risk index function. Our target value is BG level at 112.517 mg/dL for Clarke BGRI and 138.89 mg/dL for Magni RI and having a zero BGRI implies that there is no risk for a patient at this point (Zero-risk). This plot shows graphically this point, including the left and right sides of the zero-risk target value, i.e., LBGI and HBGI, respectively. BG level at 70-350 mg/dL is considered as the target range.

a simple reward to encourage large episodes, which results in extended periods within adequate BG level regimes. Of course, our reward function can be refined, for instance, considering more sophisticated safe zones. Therefore, the reward is simply:

$$reward = \begin{cases} 1, & \text{if } BG \in [70, 350] \text{ mg/dL}. \\ 0, & \text{if } BG \in [10, 69] \text{ or } [351, 1000] \text{ mg/dL}. \\ -100, & \text{otherwise}. \end{cases} \tag{3.7}$$

**Discount factor:** the discount factor helps to balance the importance of immediate and future rewards. Since the effect of insulin on the BG is usually delayed, we set it to a relatively large value of $\gamma = 0.999$.

**Virtual population:** The UVA/PADOVA simulator provides parameters for fully specifying the glucoregulatory system of patients in three groups: children, adolescents, and adults, each category including 10 patients. According to the *SimGlucose* developer [79], the patient parameters correspond to the 30-patient subset available for the academic edition of the 2008 commercial UVA/PADOVA simulator. The commercial version provides a virtual population of 100 patients in each group.

Although 30 patients cannot cover all the possible variations in a heterogeneous population, the size of our population is similar to most of the previous works based on RL: according to the systematic review by Tejedor [6], only 3 out of 23 proposals that used *in silico* patients have over 30 patients. These proposals use 100-patient sets from the UVA/PADOVA simulator. Moreover, ten of the reviewed proposals use just one *in silico* patient. Unlike other methods, such as model-based classical control (MPC) [13, 14], which have an almost negligible computational cost and can be evaluated on thousands of virtual patients, methods based on RL typically use a reduced number of patients. There are two main practical reasons behind these low numbers. First, training RL agents is a highly time-consuming and resource-intensive task. Effective training of a single patient with a set of parameters and hyper-parameters typically takes seven to ten days with our mid-level computing facilities (Intel i9-10920X, 64 GB RAM, 2 Nvidia RTX 2080 GPUs). We are able to train 4 to 8 patients in parallel. Second, even if enough time and computing power is available, to train an RL agent, we have to rely on a proven training environment and a set of validated virtual patients. Generating additional patients is possible by sampling from the joint distribution of the model parameters, as described in [19], and variations of this generation method have been used by Di Ferdinando [13] and Borri [14]. However, the values of several parameters were not published, which makes it necessary to guess some of them. Pompa *et al.* have very recently compiled the required parameters for future implementations [80]. Nevertheless, we consider that the effort required to rigorously generate patients is beyond the scope of the current chapter.

Thus, for an initial search for a viable implementation strategy for RL, which is the goal, we consider that 30 patients split into age groups is a reasonable trade-off. As said, our choice is in line with most of the previous works on this topic and even surpasses most of them. Once a viable implementation strategy has been established, a more comprehensive training campaign can be carried out, including the generation of additional virtual patients.

### 3.1.2. Initial evaluation

We conducted a series of initial tests to determine the features of the environment that may have a greater influence on the agent learning, according to the initial choices described in the previous section. Simglucose comes with a population of 30 virtual patients: 10 adolescents, 10 adults, and 10 children, which statistically represent different cohorts of patients [41]. We have tested on one patient from each group the initial alternatives that are summarized below and in Table 3.1.

- Algorithms: **PPO** and **SAC**, with a standard configuration using dense DNN with two hidden layers of 256 units. In addition, we used an alternative recurrent architecture intended to capture temporal context, which uses as actor and critic networks a RNN with a 10 LSTM cells. We call this variant **PPO-RNN** and **SAC-RNN**. The performance of PPO-RNN and SAC-RNN was similar to the one shown in Table 3.1 and will not be reproduced.

**Table 3.1:** Summary of initial evaluation.All columns show average ± 95% confidence intervals.

| Subject | Alg/Obs/Reward | Length(min) | Hypoglycemic(%) | Hyperglycemic(%) | Euglycemic(%) |
|---|---|---|---|---|---|
| Child#1 | PPO/O1/R1 | 234 ± 50.921 | 0.0077 ± 0.003 | 0.44 ± 0.03 | 0.550 ± 0.040 |
| Child#1 | PPO/O2/R1 | 213.3 ± 34.836 | 0.004 ± 0.004 | 0.41 ± 0.06 | 0.58 ± 0.069 |
| Child#1 | PPO/O1/R2 | 42.9 ± 9.8369 | 0.074 ± 0.009 | 0± 0 | 0.925 ± 0.0093 |
| Child#1 | PPO/O2/R2 | 41.1 ± 6.7217 | 0.075 ± 0.007 | 0 ± 0 | 0.92 ± 0.0079 |
| Adolescent#1 | PPO/O1/R1 | 617.7 ± 155.75 | ± 0 | 0.48 ± 0.11 | 0.510 ± 0.11 |
| Adolescent#1 | PPO/O2/R1 | 536.7 ± 147.89 | ± 0 | 0.58 ± 0.09 | 0.415 ± 0.094 |
| Adolescent#1 | PPO/O1/R2 | 59.4 ± 1.8709 | 0.050 ± 0.001 | 0± 0 | 0.94 ± 0.0015 |
| Adolescent#1 | PPO/O2/R2 | 60 ± 0 | 0.05 ± 0 | 0 ± 0 | 0.95 ± 0 |
| Adult#1 | PPO/O1/R1 | 555.6 ± 157.16 | 0 ± 0 | 0.57 ± 0.14 | 0.42 ± 0.1 |
| Adult#1 | PPO/O2/R1 | 505.5 ± 121.09 | 0.0057 ± 0.001 | 0.41 ± 0.08 | 0.57 ± 0.08 |
| Adult#1 | PPO/O1/R2 | 63.6 ± 0.85843 | 0.047 ± 0.000619 | 0± 0 | 0.95 ± 0.0006 |
| Adult#1 | PPO/O2/R2 | 63.6 ± 0.85843 | 0.047 ± 0.000619 | 0± 0 | 0.95 ± 0.0006 |
| Child#1 | SAC/O1/R1 | 39 ± 0 | 0.0769 ± 9.9276e-18 | 0 ± 0 | 0.92308± 0 |
| Child#1 | SAC/O2/R1 | 63.6 ± 19.166 | 0.05248± 0.010184 | 0.030 ± 0.064 | 0.91773± 0.05 |
| Child#1 | SAC/O1/R2 | 36.3 ± 0.64 | 0.082± 0.001 | 0 ± 0 | 0.91731± 0.001 |
| Child#1 | SAC/O2/R2 | 36.9 ± 0.98 | 0.08± 0.002 | 0 ± 0 | 0.91859± 0.002 |
| Adolescent#1 | SAC/O1/R1 | 123.6 ± 12.13 | 0.02± 0.001 | 0 ± 0 | 0.97538± 0.001 |
| Adolescent#1 | SAC/O2/R1 | 97.2 ± 15.30 | 0.03± 0.0046488 | 0 ± 0 | 0.96773± 0.004 |
| Adolescent#1 | SAC/O1/R2 | 54 ± 0 | 0.05± 4.9638e-18 | 0 ± 0 | 0.94444± 0 |
| Adolescent#1 | SAC/O2/R2 | 64.5 ± 7.57 | 0.04± 0.0044488 | 0 ± 0 | 0.95248± 0.004 |
| Adult#1 | SAC/O1/R1 | 87.9 ± 14.83 | 0.03± 0.003685 | 0 ± 0 | 0.96466± 0.00 |
| Adult#1 | SAC/O2/R1 | 137.1 ± 29.87 | 0.02± 0.005159 | 0 ± 0 | 0.97598± 0.00 |
| Adult#1 | SAC/O1/R2 | 59.1 ± 0.98 | 0.05± 0.00086268 | 0 ± 0 | 0.94921± 0.0008 |
| Adult#1 | SAC/O2/R2 | 66.6 ± 2.68 | 0.04± 0.0018355 | 0 ± 0 | 0.95481± 0.001 |

- Observation space: we used as observation both **(O1)** the current CGM sample and a **(O2)** vector of the past 20 CGM samples, corresponding to the previous hour at 3-minute intervals.
- Reward functions: we used **(R1)** the one point per step reward of eq.3.7 and **(R2)** the negative of the Clarke BGRI with a termination penalty. In all the case we set the termination limits to BG<70 or BG>350.
- Meal schedule: the simulator selects a non-deterministic meal schedule particular to each patient according to the Harris-Benedict algorithm [8].

We used the PPO and SAC implementations from *stable-baselines3* [81] and PPO-RNN, SAC-RNN from *TensorFlow Agents* (TF-Agents [82]) and trained it on two Nvidia GeForce RTX 2080 GPUs and Intel Core i9-10920X CPU @ 3.50GHz 12 cores. All agents were trained for 1 million steps, keeping the best model (best average reward) and with a maximum episode length of 10,000 steps (a step represents 3-minute interval).

From the average length of the episode, it can be seen that the agents are not able to keep a safe control of BG levels beyond 10 hours. In general, PPO is able to achieve longer duration within a safe BG range, because it injects lower insulin levels and patients spend more time in a hyperglycemic state. In fact, most of the episodes with PPO end because of high BG levels. On the contrary, SAC tends to inject insulin aggressively and patients go rapidly to a hypoglycemic state, ending the episode. In both cases, using just the current CGM observation (O1) or the last one-hour interval of CGM observations (O2) has little influence, and using the simple reward (R1) works better than the negative of the Clarke BGRI.

### 3.1.3. Discussion and refinement of design

From the previous evaluation, we hypothesize that the three main characteristics of the simulation that affect the lack of training success are: (1) The action drives the BG level to lower values, but the rise of BG level depends on the environment dynamics and as consequence of the exogenous input actions, that is, the intake of CHO in the meals. (2) The observation (CGM) is a noisy sample of the BG level. (3) The effect of actions on the BG level is delayed. That is, applying an action does not immediately decrease the BG level. Even though those effects are expected from the initial analysis of the environment, and not particularly surprising, we think they deserve further discussion.

Regarding (1), the agent should learn the policy to deal with this fact, that is, that it does not need to deliver insulin when the BG level is low. In fact, our results show that the PPO agent is able to learn policies that anticipate the meal consumption and the subsequent rise of BG. However, they are not enough precise to control adequately the BG levels. This is probably because of the conservative BG range that leads to an early episode ending. Regarding (2), let us just notice that the reward is usually assigned according to the actual BG level, not the CGM sample, which seems to negatively affect learning since it makes termination penalties appear random. But even using only BG levels, instead of noisy CGM samples, did not improve learning. Regarding (3), both the large discount factor and the recurrent architecture should have improved learning. But the termination limits of the environment, the glucose dynamics and the randomness of meal schedules make it hard to learn: if the agent tries high insulin does, the patient goes very quickly to hypoglycemia, and the episode ends, and when the agent injects low doses, meals, which are not included in the observation, raise the BG level ending also the episode. Therefore, it seems that the combination of all these factors prevents the agent from properly learning. Both (2) and (3) stem from the fact that we are dealing with a POMDP and the recurrent architecture should improve learning, but several variations tried in our tests did not actually improve much. We could have tried changing the RNN architecture and other hyperparameters, but due to the large parameter space, we chose to focus on the delay of actions as follows.

**Observation frequency and insulin response**: We just configured the environment to decrement the frequency of the observations and actions, instead of using the usual 3 or 5-minute

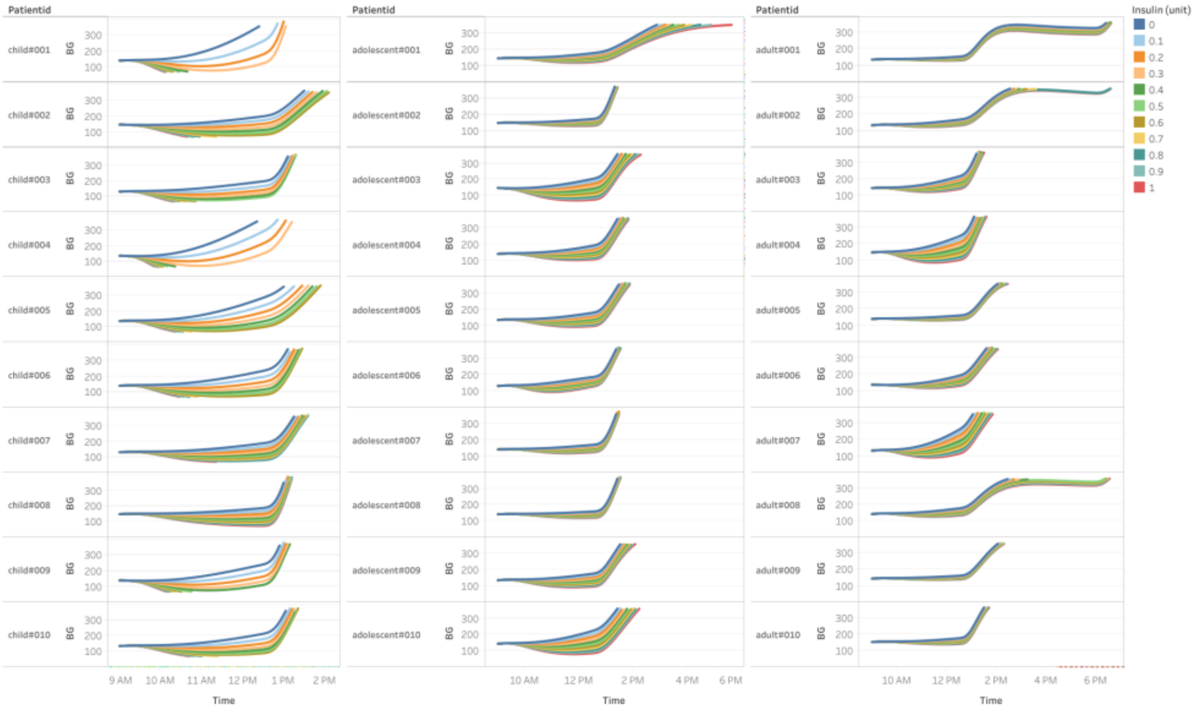


**Figure 3.2:** Insulin response time for each patient. The insulin dose depicted in the color legend on the right was injected at 9:00 AM and no meal is taken subsequently.

CGM sample resolution. Even though the samples are taken, since the SimGlucose environment is running and updating the state in *mini-steps* of one minute, they are passed less frequently to the agent, which then provides an action. The rationale is simple: to let the agent observe the actual effects of its actions, that is, the actual patient insulin response according to the glucose dynamics, instead of a seemingly random transition. Even though it is similar to using as state a history of the previous CGM data, as is done in [8], it improves the learning process and leads to more adequate insulin regimes in our results. Therefore, we have introduced an additional hyperparameter, *observation frequency*, which is actually already present in the simulator (CGM sample resolution) although usually left at the default device value (a CGM sample every three minutes) [8]. Let us notice that this new hyperparameter does not increase the complexity compared to using a history of previous samples [8], since in that case, the lengths of the history vectors are also additional hyperparameters to be tuned.

We selected the frequency for the observations by testing the insulin response time when injecting a given unit of insulin without taking any meals. This delay is different depending on the patient group, as expected, and it is shown in Fig.3.2: ten different amounts of insulin doses were used to estimate each subject insulin reaction, from 1 to 30 units. As can be seen, the adult response to insulin tends to be more stable and less pronounced and the reduction in BG starts to show around 45 minutes after injection. Adolescent reaction is slightly slower and more pronounced. Finally, children clearly react faster and more strongly to insulin. In fact, high insulin doses quickly drive some of the patients to hypoglycemia and episode termination. From these observations we chose reducing the frequency of observations as follows: for adults, observations are made every 45 minutes (corresponding to 15 three-minute environment steps), adolescent frequency is set to 30 minutes (10 steps) and children is set to 15 minutes (5 steps).

Then, we started over the training of the **SAC**, **SAC-RNN**, **PPO** and **PPO-RNN** agents. After running several experiments, only **PPO-RNN** could learn effective policies. Therefore, we have selected this algorithm and architecture for a full evaluation of performance in Section 3.2.

### 3.1.4. Summary of implementation strategy

We provide here a brief summary of our implementation strategy, before conducting an evaluation and comparison of alternatives in the next section.

From the discussion in the previous section we derive the following implementation strategy: (1) we reduce the *observation frequency* of the environment state (CGM) and, hence, rewards passed to the agent, depending on the subject (45 minutes, 1 hour and 15 minutes for adults, adolescents and children, respectively), (2) we set broad termination limits for the episodes, $BG \in [10, 1000]$, to let the agent explore more thoroughly the environment; and (3) we use the simple reward function of eq. (3.8) below, to force the agent to learn to keep the patients in euglycemia for as long as possible.

$$reward = \begin{cases} 1, & \text{if } BG \in [70, 180] \text{ mg/dL.} \\ 0, & \text{if } BG \in [10, 69] \text{ or } [181, 1000] \text{ mg/dL.} \\ -100, & \text{otherwise.} \end{cases} \tag{3.8}$$

Let us note that the reward is accumulated during all the simulation *ministeps* and then passed to the agent. For example, if we set the observation frequency to 10, the environment is going to simulate 10 ministeps before passing the sample to the agent, and, if BG level has been in the desired range all those ministeps, the accumulated reward passed will be 10.

## 3.2. Evaluation of the system through simulation

### 3.2.1. Experimental setup

**Training and evaluation**. Our goal is to keep the patient BG level in the selected range for as long as possible. We have trained a PPO-RNN agent for each of the patients with the implementation strategy summarized in the previous section and the hyperparameters listed in Table 3.2. During the training, there are periods of instability, where the average reward drops, as reported in other studies [8]. Rigorous convergence of the RL algorithms tested in this chapter, SAC and PPO,

**Table 3.2:** Hyperparameters for PPO-RNN

| **Hyperparameter name** | Value |
|---|---|
| actor_fc_layers | 200, 100 |
| value_fc_layers | 200, 100 |
| actor_lstm_size | 128, 128 |
| critic_lstm_size | 128, 128 |
| num_environment_steps | 25000000 |
| collect_episodes_per_iteration | 10 |
| num_parallel_environments | 30 |
| replay_buffer_capacity | 1001 |
| num_epochs | 25 |
| learning_rate | 1e-3 |
| num_eval_episodes | 20 |

including their variants with RNN, has not been proved analytically, in general, to the best of our knowledge. Only recently, the convergence of PPO to a local minimum of the associated losses has been proved [83]. In practice, the convergence of the algorithms is assumed when the learning curve does not improve over time. In our case, both PPO and SAC tend to show an oscillatory behavior in the learning curves so that the learning curve increases and then drops. PPO is able to recover from this behavior and we stop the training process when the learning curve has stabilized. The reason for these oscillations is likely the presence of unbounded exogenous stochastic inputs, that is, the meals or the noisy observations. We save the policy every 100 training steps and select the policy with a highest average reward as a trained agent. Once trained, the agents are evaluated 20 times with different seeds for all the patients, and statistics for episode length, fraction of time in glycemia states (eu, hypo and hyper) and other metrics are collected. For evaluation, we also set the environment termination limits to $BG = 10$ and $BG = 1000$, in order to test the fraction of time that the patients spend in the different states and make them comparable to similar proposals [8]. Let us remark that patients whose BG reaches levels below or above those limits are considered events that result in serious damage or death.

### 3.2.2. Baselines

We have compared our results with four baselines: a basal-bolus regime (BB), that simulates the usual self-managed treatment for patients with both T1D and T2D, basal-bolus with cooldown, a PID, and PID with insulin feedback baselines.

**Basal-bolus Baseline (BB).** A multiple daily injection therapy which involves using long-acting insulin with a dosage of basal:

$$basal = \frac{u2ss \times body\,weight(kg)}{6000(U/min)} \tag{3.9}$$

where $u2ss$ is the steady state insulin rate per kilogram $pmol/L \times kg$; and short or rapid-acting insulin (bolus) to regulate blood glucose concentration with bolus:

$$bolus = (CHO > 0) * (\frac{CHO}{CR} + \frac{BG_{current} - BG_{target}}{CF})/t \tag{3.10}$$

where CF is a correction factor, t is the time between samples and CR is a carbohydrate ratio [8].

To obtain a more stable regime, an alternative is to apply a cooldown signal to the basal-bolus insulin delivery policy (**BB-CD**) to ensure that each meal is corrected only once:

$$bolus = (CHO > 0) * (\frac{CHO}{CR} + cooldown * \frac{BG_{current} - BG_{target}}{CF})/t \tag{3.11}$$

where cooldown is 1, if the patient has not had meals in the past three hours and otherwise is 0. Let us note that this treatment requires the patient to be aware of the meal intake and so it is not a closed-loop control. The controls that require explicit knowledge of meal intake are usually called *controls with meal announcement* in the literature.

**PID Baseline (PID)**. A closed-loop control which uses a discrete PID controller aims to set the system output to a given target, $s_t$, by setting the control variable $a_k$ as a linear combination of three terms:

$$a_k = K_p P(s_k) + K_i I(s_k) + K_d D(s_k) \tag{3.12}$$

where $P(s_k) = s_k - s_t$, $I(s_k) = \sum_{i=0}^{k}(s_i - s_t)$ and $D(s_k) = |s_k - s_{k-1}|$.

We use the optimal values for the PID parameters, $K_p$, $K_d$, $K_i$, for each patient provided by Fox *et al.*[8]. In fact, insulin in blood suppresses the next insulin production, called insulin feedback. Thus, we introduced PID control with insulin feedback (**PID-IF**) based on [11],

$$a_d(k) = (1 + \gamma/K_{pi}) * a_k - \gamma * I_p(t) \tag{3.13}$$

where $\gamma$ is the degree of suppressed insulin delivery by the current plasma insulin, which is equal 0.5, $K_{pi}$ is the normalized insulin concentration in units. $K_{pi}$ is equal to 1, and $I_p(t)$ is the model of pharmacokinetics of insulin adapted from [12], given by

$$I_p(t) = \frac{I_B}{K_{pi}(\tau_2 - \tau_1)}(e^{-t/\tau_2} - e^{-t/\tau_1}) \tag{3.14}$$

where the parameter $I_B$ is the insulin injected in the previous action, and $\tau_1$ and $\tau_2$ are time constants (in minute) associated with the subcutaneous absorption of insulin equal to 55 and 70, respectively.

### 3.2.3. Episode length

**Table 3.3:** Fraction of completed 10-day evaluation reached for each method and group.

| Group-Method | Fraction of full episode (average % ± 0.95 CI) |
|---|---|
| Children-BB | 77.6 ± 4.7 |
| Children-BB-CD | 100 ± 0 |
| Children-PID | 100 ± 0 |
| Children-PID-IF | 100 ± 0 |
| Children-PPO-RNN | 100 ± 0 |
| Adolescent-BB | 88.2 ± 3.8 |
| Adolescent-BB-CD | 100 ± 0 |
| Adolescent-PID | 100 ± 0 |
| Adolescent-PID-IF | 100 ± 0 |
| Adolescent-PPO-RNN | 100 ± 0 |
| Adult-BB | 91.9 ± 3.3 |
| Adult-BB-CD | 100 ± 0 |
| Adult-PID | 100 ± 0 |
| Adult-PID-IF | 100 ± 0 |
| Adult-PPO-RNN | 100 ± 0 |

First, we examine the average episode length in evaluation. Table 3.3 shows the average fraction of episodes that were completed by each patient group. Note that an episode is terminated when the BG level goes out of the 10-1000 mg/dL range, which means that the patient has reached a BG level that may result in serious damage or death. A primary goal, therefore, of any control method is to avoid early episode termination. Almost all control methods are able to make all the patients finish the 10-day evaluation simulations for all the groups except for BB. The adult group is the easier to control and all patients. PPO-RNN, BB-CD, PID and PID-IF were able to finish the 10-day evaluation period for all the 20 simulations. On the contrary, the children are the most difficult group and BB can only reach on average 78% of full episodes, that is 7.8 days. The introduction of a cooldown (BB-CD) improves basal-bolus overall episode length. But being able to finish the evaluation period is not enough to determine the quality of the treatment: BG levels of patients must be kept in the desired range for as long as possible. In the following sections, we examine how the controls keep the state of the patient during that period.

### 3.2.4. Risk Index and Glycemic states

We compare the results of BB, BB-CD, PID, PID-IF and PPO-RNN controllers for the risk index and fraction of time spent in hyper, hypo and euglycemia. The aim is to determine how well controllers regulate the risk of hypoglycemia and hyperglycemia. First, Fig.3.3 shows with boxplots the distribution for all the methods evaluated. As can be seen, PPO-RNN makes all patients spend more time in an euglycemic state than the baselines, which is the actual goal of this mechanism. Both the median and 25 and 50 percentiles are above those of the other methods. In addition, PPO-RNN also outperforms the baselines globally in terms of the fraction of time spent in hyperglycemia and hypoglycemia. It is instructive to remark how the distributions are more informative in this case that single point estimates, such as the median or mean. For instance, even though the median for the hypoglycemic fraction is similar for PPO-RNN and PID, we can see that a remarkable number of patients spent an unacceptably large fraction of time in hypoglycemia with all PID variants.

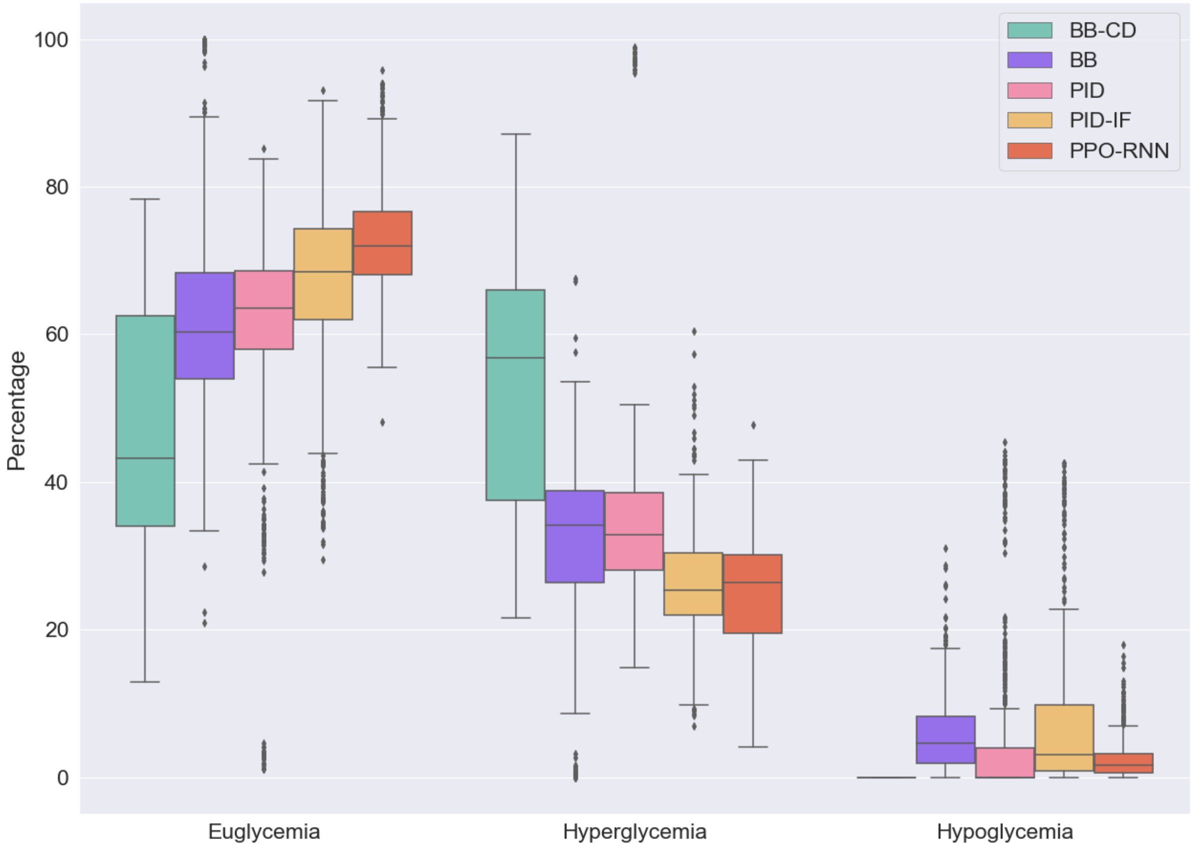


**Figure 3.3:** Comparative fraction of time spent in global glycemic states.

In Fig.3.3 we show that PPO-RNN also outperforms the baselines when the patient groups are examined separately. The results for children are especially noteworthy, since, as we have discussed, it is the most difficult group to train. For that group, we remark that: (1) PPO is able to perform especially well to avoid hypoglycemia, unlike PID, which fails clearly in this aspect; and (2) BB seems to provide reasonable results for children. However, for BB, the results of the previous section have to be taken into account, that is, that the control is only able to reach 78% of the full episode on average. In other words, the control may provide a good response on average for typical BG levels and meal intakes, but not be able to react to unusual variations, which drive the patient to a dangerous state. On the other hand, BB-CD is able to practically eliminate hypoglycemia, but at the cost of a much higher hyperglycemia for all groups.

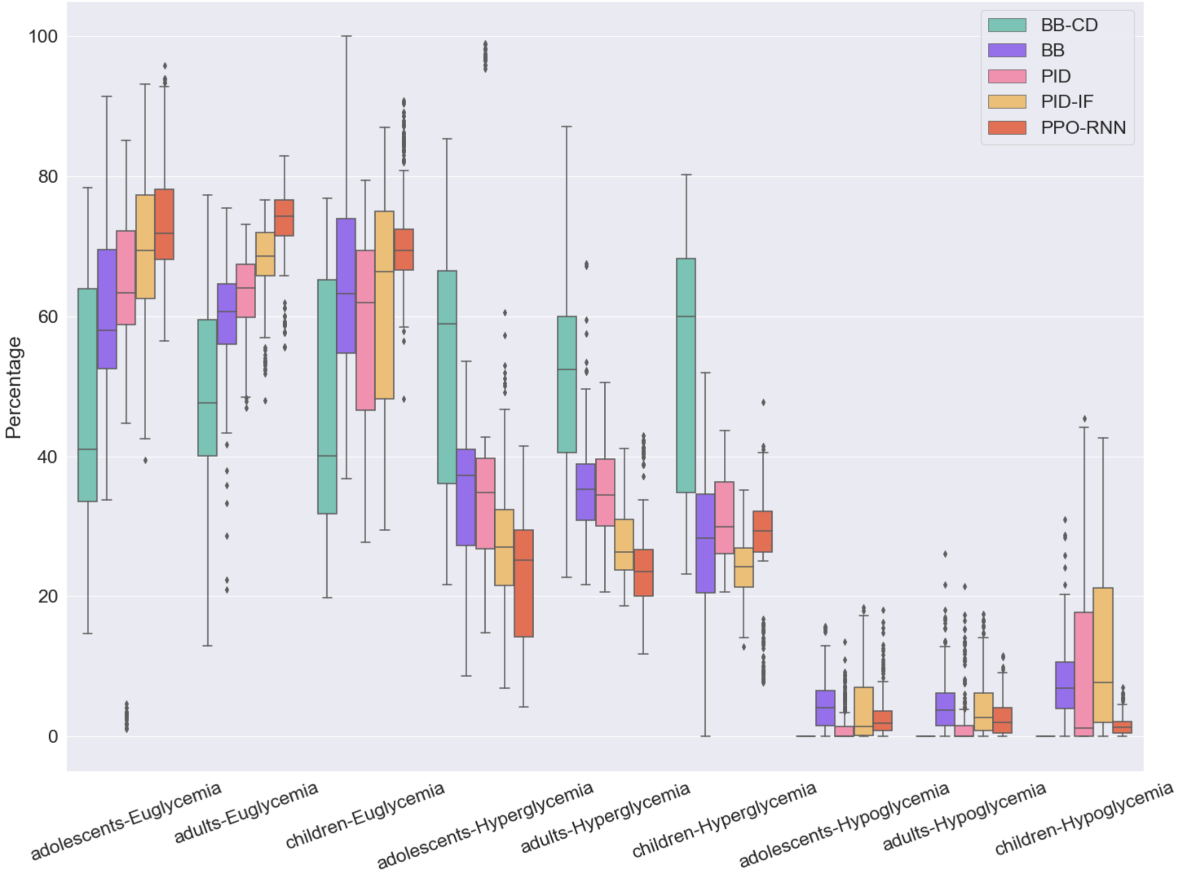


**Figure 3.4:** Comparative fraction of time spent in glycemic states by group.

Therefore, to better assess the results, it is necessary to look also at the risk index, which informs us whether the patient is at a safe level within the desirable range. For instance, a patient may spend a large fraction of the time in the euglycemic range but with BG levels very close to the hyper or hypoglycemia thresholds, which may make it vulnerable to unusual conditions, such as irregular meal intakes. Recall that RI penalizes more hypoglycemia, because even though both hypoglycemia or hyperglycemia can lead to fatal outcomes, the short-term effects of hypoglycemia can cause T1D patients to have an immediate crisis [10], as opposed to hyperglycemia, whose effects manifest in the long term. We show the average RI, HBGI and LBGI all over the evaluation period globally in Fig.3.5 and by groups in Fig.3.6. In both cases, PPO-RNN outperforms the other controls and keeps all the RI metrics within reasonable levels, unlike the baselines, especially PID and BB-CD, which show high RI metrics. These results show that PPO-RNN keeps the patients within safe limits most of the time, unlike the baselines, which do not success in smoothly control BG levels: Even though the patients may be euglycemic, they exhibit less safe BG levels. This is the reason why, combined with poor adaptability, some BB-controlled patients are not able to finish the 10-day episodes.

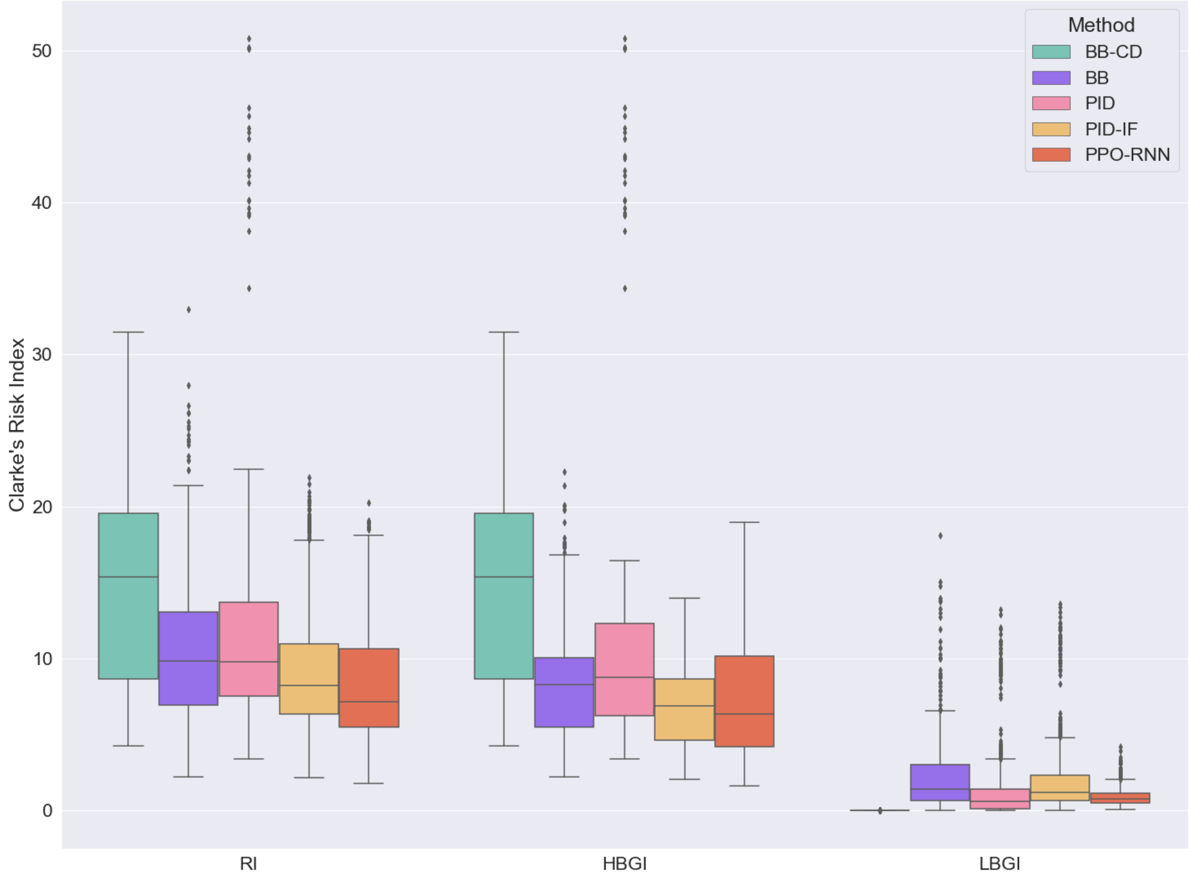


**Figure 3.5:** Comparative fraction of global risk index.

## 3.3. Discussion of results

Section 3.2 shows that our implementation strategy can effectively control BG levels and consistently outperforms the baselines. A first result to remark is that our tests showed that no patients controlled by PPO-RNN terminated earlier in any of the evaluation trials. Early terminations are called *catastrophic failures* by Fox [8], since they signal potentially fatal BG levels. Our work in this aspect shows slightly better results than Fox [8], which shows evaluation failures of around 0.1%, but their evaluation extended to 100 10-day replications per patient, whereas ours has been limited to 20 10-day replications. On the other hand, we define a failure when BG goes below

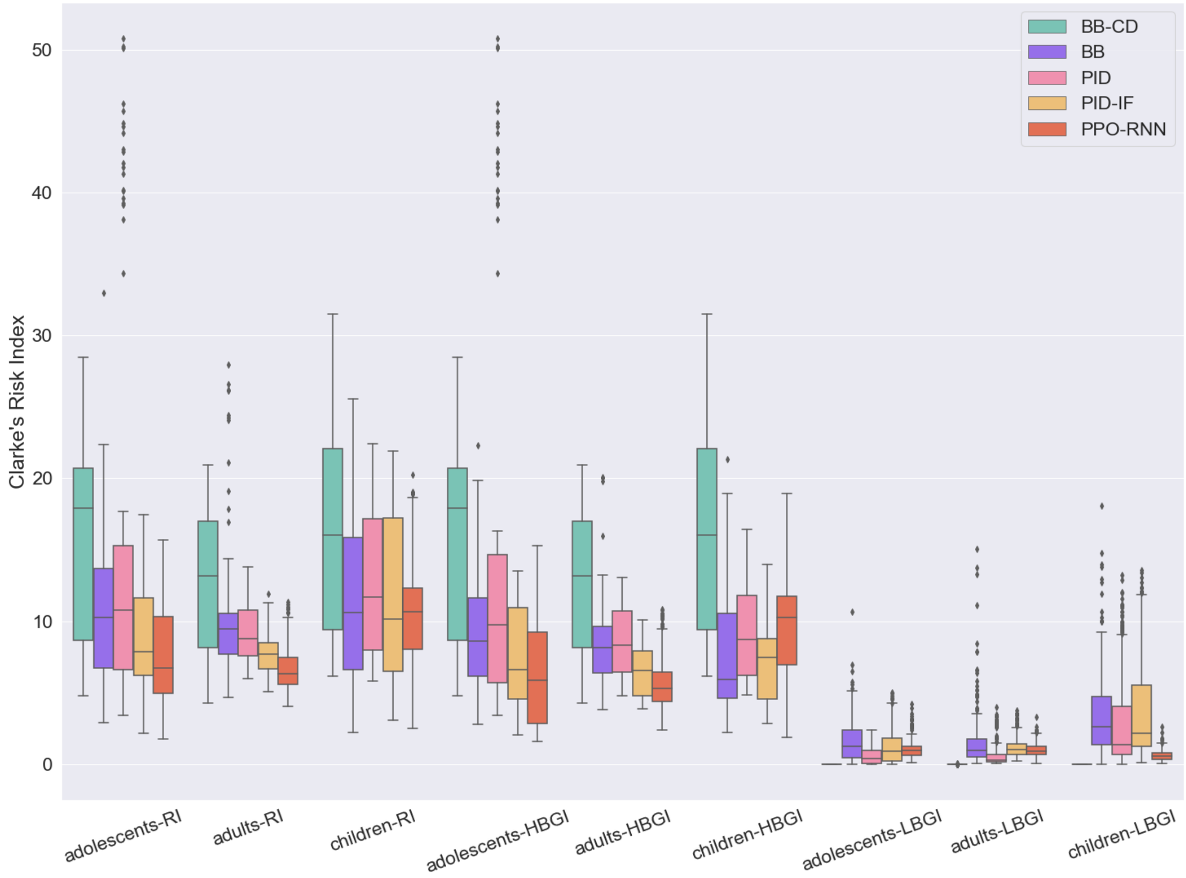


**Figure 3.6:** Comparative fraction of risk index by group.

10 mg/dL, whereas they apply an even lower threshold of 5 mg/dL. PPO-RNN clearly improves over Yamagata [20], which reports very high failure percentages, with some patients completely unable to finish any episode. Of course, the critical nature of BG control requires exhaustive evaluation of results. Closed-loop controllers must be evaluated by its capability to keep the BG within an acceptable range for years. As a future work we will evaluate for extended time periods, but, since our tests show that patients keep BG levels above 50 mg/dL and below 400 mg/dL in all the replications so far, we are confident PPO-RNN can avoid failures for longer episodes.

Attending to the time spent in euglycemia, our results are in line with the ones reported by Fox[8], but outperforms those of Lim [27] and Yamagata [20]. Regarding the former, the implementation strategy, as well as ours, is able to keep euglycemia over 73% of time globally, and, in our case, also for all groups. Our implementation strategy can be considered simpler than the Fox one because our observation space is unidimensional (last CGM sample), which makes training more efficient, whereas Fox uses 96 dimensions (previous 48 CGM and insulin data samples). The extended observation frequency for each patient group is an additional hyperparameter to be optimized in our case, but the choice of 48 previous samples is also a hyperparamter in their case. Lim and Yamagata papers show only marginal improvements over the baselines using different strategies, and in both cases only report 64% of time in euglycemia. Moreover, in all the discussed papers, the children group shows more difficulties to be appropriately controlled with a closed-loop controller. In fact, in the work by Lim [27] the children group is not evaluated at all, despite using a fairly sophisticated control involving DRL driven by PID as initial policy. Similarly, in [20] only three patients of each group were evaluated.

The evaluation discussed in this proposal and the aforementioned works show that the application of DRL to BG control is a challenging task. Our results show, however, that DRL has a great potential as a closed-loop controller. Proposals put forward so far discuss just a tiny fraction of the space of potential implementation strategies for DRL-based BG control. The fact that the employed ones at the moment are relatively simple but exhibit a performance better than PID and PID-IF, encourages further exploration of this approach. In our case, there is a great margin for improvement. As a future work, we plan to optimize the hyperparameters of our method and test variations in the RNN architecture. We think that changes in the reward function, narrowing the desired range to obtain reward, may help improve the euglycemia fraction. The introduction of an extended observation frequency in our case has been the key to make agents learn effectively. It is introduced as a new hyperparameter, which may be optimized, but we plan to try strategies to make agents learn it. Once the strategy has been optimized, we plan to conduct a thorough analysis of the way the agent applies insulin doses and compare it with those prescribed by clinical practitioners. The goal is to find whether trained policies employ unusual patterns of dosage that may help clinical practice. In any case, we acknowledge a slight limitation of our approach: the small size of the virtual population of patients, which cannot completely reflect a heterogeneous population. As discussed in previous sections, this is a common drawback of current RL proposals, due to demanding time and computation requirements of this method, and specially, the lack of validated patient sets. Since an exhaustive compilation of the required parameters have been recently published [80], we plan to generate and validate additional sets of virtual patients. Finally, as alternative implementation strategies, it should be worth trying algorithms designed specifically to deal with POMDP and input-dependent environments [29, 50].

As a final note, it is important to consider how to practically apply RL-based controls in real patients. Only a proof-of-concept approach can be considered, due to the difficulties for real application. That is, training an agent requires experimentation (insulin injection) on the subject to learn the optimal control, which is out of the question for real patients. It can only be tested on virtual patients and how to transfer it to real patients is a difficult matter [84]. However, our results may be used with more realistic approaches, such as *offline RL* [33]. With this novel method, the agent, usually a neural network with a transformer architecture, is trained on previously collected datasets, without direct experimentation on the subject. Those datasets may correspond to a series of BG levels and insulin doses collected from real patients, but they may also come from simulations on virtual patients, and both can be combined. Surprisingly, agents trained this way may show better performance than the original methods [34, 35], especially if the datasets contain high-reward regions of the state space. Therefore, the availability of methods to generate very diverse datasets for further use as input, combined with additional data sources, is important in offline RL. The agents proposed here can actually generate such high-rewards datasets.

# 4

# Optimization of PID parameters for blood glucose control

In this chapter, the focus is on the utilization of PID parameters and meal schedules in a virtual TIR patient environment, along with the introduction of a novel hyperparameter, the observation frequency (OF). The study begins with an assessment of conventional PID parameters as presented by Fox *et al.* [8], which operate under a 3-minute observation interval. In the previous chapter, we saw that changing the observation frequency can improve glucose control performance, so we want to investigate if this also applies to PID control. However, using different OF values may require new PID parameters that need to be found. To adapt to different observation frequencies, the optimization framework Optuna [85] is utilized to determine new PID parameters by adjusting the observation interval to insulin response based on age groups. The aim of this study is to investigate and evaluate personalized OF values to enhance the regulation of blood glucose levels in virtual patients with diabetes. We compare the OF adaptation with other commonly used conventional variations of PID, including the use of Harrison-Benedict meal generation [8] and insulin feedback [11]. To evaluate the impact of these values, experiments will be conducted and analyzed to determine the effectiveness of this approach in improving glucose control. The results and discussions of the experiments will be presented to further explore the impact of individualized OF values on glucose control and highlight any potential limitations.

## 4.1. Implementation strategy

### 4.1.1. Environment and experiment setup

To determine the optimal PID parameters ($K_p$, $K_i$, $K_d$) for controlling blood glucose levels in individuals with Type 1 Diabetes without taking into account meal announcements, we conduct experiments using the UVA/PADOVA T1D simulator, SimGlucose [41]. This virtual environment provides 30 patients, divided into three age groups: adults, adolescents, and children, with 10 patients per group. The CGM readings are taken every three minutes, with insulin infusion corresponding to the CGM readings. The BG level range is between 10 and 1000, and the simulations will be terminated if the BG level goes outside of this range due to catastrophic events.

### 4.1.2. Evaluation of ordinary PID for T1D BG regulation

First, we adopt the PID parameters from Fox *et al.* as implemented in the SimGlucose environment, which enables us to evaluate the glucose control of virtual patients with Type 1 Diabetes. The set point in this simulation is 112.517, which is based on the optimal point in the Clarke's blood glucose risk index [76]. This value serves as a target BG level for the PID controller to maintain throughout the simulation. In our study, we choose to use only insulin for injection, as the basal and bolus rates are treated similarly. According to Bergenstal [86], CGM systems can transmit glucose readings to a receiver, insulin pump, phone or watch at intervals ranging from 1 to 15 minutes. However, the specific frequency of CGM readings and insulin infusion intervals may vary depending on the device and the clinical setting. In our study, we have set the CGM reading and insulin infusion intervals to three minutes to provide a relatively high-resolution representation of BG levels and insulin delivery. This experimental design allows us to understand the current effectiveness of PID-based BG control and lay the groundwork for future advancements in this field.

### 4.1.3. Optimizing PID for BG regulation with Harrison-Benedict Meal Generation Algorithm

It should be noted that in the original SimGlucose, meals are generated by simple random probability for six meals, including breakfast, lunch, dinner, and three snacks. This approach can result in meal calculations that are highly fluctuating, not realistic and difficult to control. Thus, to solve this issue, we incorporate the Harrison-Benedict Meal Generation Algorithm, introduced in [8], into the SimGlucose environment. This algorithm calculates the estimated daily carbohydrate consumption for each individual based on their basal metabolic rate (BMR). The BMR is calculated based on factors such as sex, weight, height, and age. The estimated daily carbohydrates are divided among six potential meals: breakfast, lunch, dinner, and three snacks. The probability of occurrence and expected size of each meal is set to match the estimated BMR.

### 4.1.4. Optimizing PID for BG regulation with insulin feedback

To account for the fact that insulin in the blood can suppress subsequent insulin production (referred to as insulin feedback), we have introduced a control method called PID with insulin feedback (PID-IF) based on Huyett *et al.* [11]. This method takes into account the current plasma insulin concentration and modifies insulin delivery accordingly. The formula used is:

$$a_d(k) = (1 + \gamma / K_{pi}) * a_k - \gamma * I_p(t) \tag{4.1}$$

Where $\gamma$ represents the degree of insulin delivery suppression by the current plasma insulin (assumed to be 0.5), $K_{pi}$ is the normalized insulin concentration in units (set to 1), and $I_p(t)$ is a model of insulin's pharmacokinetics, adapted from Palerm et al. [12], given by:

$$I_p(t) = \frac{I_B}{K_{pi}(\tau_2 - \tau_1)}(e^{-t/\tau_2} - e^{-t/\tau_1}) \tag{4.2}$$

In this equation, $I_B$ represents the insulin injected in the previous action, and $\tau_1$ and $\tau_2$ are time constants (measured in minutes) associated with insulin's subcutaneous absorption, which are set to 55 and 70, respectively.

### 4.1.5. Selection of optimization algorithm for PID with observation frequency

In addition to insulin feedback, we introduce a new hyperparameter called observation frequency (OF), which determines the frequency of glucose observations and its impact on insulin response. The OF was selected based on human observations of the insulin response time of 10 different insulin doses for each patient group, resulting in OF values of 45 minutes for adults, 30 minutes for adolescents, and 15 minutes for children. The aim of the new hyperparameter is to improve glucose control and prevent hypoglycemia, while simultaneously reducing complexity compared to the approach without using it.

Upon comparing the outcomes of our evaluation to the performance of other techniques, it was discovered that Fox's parameter exhibits inadequate performance on OF, since they were optimized for 3-minute readings. To adapt to the new observation frequency, we find new PID parameters using the optimization algorithms available in the Optuna framework, such as Tree-structured Parzen Estimator (TPE) [87], Covariance matrix adaptation evolution strategy (CMA-ES) [88], and Gaussian Process (GP) [89]. In brief, we describe the following three algorithms:

**The Tree-structured Parzen Estimator (TPE)** [87] is a probabilistic machine learning algorithm that uses Bayesian optimization to determine the optimal parameters for a given problem. It is the most common optimization method in machine learning [90], particularly in hyperparameter tuning, as it offers a trade-off between exploration and exploitation by balancing the sampling of different regions of the search space.

**Covariance Matrix Adaptation Evolution Strategy (CMA-ES)** [88] is a derivative-free optimization algorithm that is particularly well-suited for non-linear and non-convex optimization problems. It uses information about the covariance matrix of the search space to guide the optimization process, and is known for its ability to handle noisy and multi-modal optimization problems.

**Gaussian Process (GP)** [89] is a machine learning method for regression and classification problems. It models the distribution of a target variable as a Gaussian process, allowing for predictions about the target variable at any given input location. In the context of optimization, GP can be used as an optimization algorithm to determine the optimal parameters for a given problem by modeling the relationship between the parameters and the objective function. GP offers a powerful framework for global optimization and is particularly useful in high-dimensional optimization problems.

We aim to optimize PID blood glucose control with OF for diabetic patients through a series of experiments. Starting with 1000 trials for each condition and algorithm with Adult#001, we choose the optimal algorithm with the highest euglycemia.

To select the optimal PID parameters, we conduct 1000 trials for each patient using the euglycemia percentage or Time in Range (TIR) metric, with BG levels between 70 to 180. We select the sets of PID parameters for each patient that achieve the highest euglycemia percentage and evaluate their performance in 20 additional trials, comparing them to other methods.

An optimization algorithm selection was conducted on Adult#001 over a five-day evaluation period to assess the efficacy of incorporating insulin feedback (IF) in controlling blood glucose levels. Two sets of $K_p$, $K_i$, and $K_d$ ranges were used, as indicated in Table 4.1. The metric of euglycemia was employed to evaluate the performance of each algorithm. The results indicated that TPE was able to find the best set of hyperparameters when applied to PID-IF with an euglycemia score of 0.714, followed by GP. Meanwhile, CMA-ES terminated prematurely in the first set of PID range and was not able to find a good set of hyperparameters in the second set. Based on these findings, TPE was selected as the method for determining the optimal PID values for each individual patient.

After selecting the TPE optimization algorithm, we utilized the Optuna framework to find optimal PID parameters for each patient. The original values of the $K_p, K_i$, and $K_d$ from Fox ($F_v$) were used as an upper boundary, $F_v \times 5$, and a lower boundary, $F_v \times 0.5$, for PID-IF. The selected algorithm is incorporated with age group OF values and the found parameters are shown in Table 4.2

**Table 4.1:** Summary of optimization algorithms.

| Algorithm | Kp range | Ki-range | Kd-range | IF | Eug | Hyper | Hypo |
|---|---|---|---|---|---|---|---|
| CMA-ES | -1.00E-02, -5.00E-03 | -1.00E-06, -1.00E-07 | -1.00E-01, -1.00E-02 | Yes | stop before reaching 5 days | | |
| TPE | | | | | 0.6981 | 0.2706 | 0.0313 |
| GP | | | | | 0.6664 | 0.3028 | 0.0309 |
| CMA-ES | | | | No | stop before reaching 5 days | | |
| TPE | | | | | 0.7008 | 0.2777 | 0.0215 |
| GP | | | | | 0.7004 | **0.2464** | 0.0532 |
| CMA-ES | -5.00E-03, -1.00E-04 | -1.00E-07, -1.00E-08 | -1.00E-02, -1.00E-03 | Yes | 0.5970 | 0.4030 | 0.0000 |
| **TPE** | | | | | **0.7138** | 0.2782 | **0.0081** |
| GP | | | | | 0.6959 | 0.2929 | 0.0112 |
| CMA-ES | | | | No | 0.4119 | 0.5881 | 0.0000 |
| TPE | | | | | 0.6485 | 0.3430 | 0.0085 |
| GP | | | | | 0.6377 | 0.3623 | 0.0000 |

### 4.1.6. Optimizing PID for BG regulation with personalized observation frequency

Thereafter, we conducted 1000 trials with the same PID parameter range as the previous step, but with the addition of a new parameter, an individualized OF for each patient in the range of 0 to 60 minutes. The evaluation metric for this step was the euglycemia percentage, and we selected the highest value for each patient. The results are shown in Table 4.3. Subsequently, we evaluated each patient with 20 evaluations, following the same method as the previous steps.

## 4.2. Evaluation of the system by simulation

We performed an evaluation of each chosen parameter by running 20 episodes of insulin delivery control. Each episode simulates 10 days of glucose control using the implemented insulin delivery controller, and the metrics of interest include episode length, percentage of euglycemia, hyperglycemia, hypoglycemia, and Clarke's risk index. The results of these evaluations are analyzed and presented to demonstrate the impact of insulin delivery control on glucose control and to identify areas for improvement. These will serve as our baseline for further exploration and optimization of blood glucose control.

### 4.2.1. Episode length

Table 4.4 presents the results of various methods used to control blood glucose levels in virtual patients with diabetes. The methods include PID, which utilizes Fox's parameters, PID-Har, which incorporates Harrison-Benedict meal generation and OF, PID-IF, which adds insulin feedback

**Table 4.2:** Optimal PID parameters with age group OF obtained by Optuna.

| **Patient** | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|
| adolescent#001 | -0.000291775 | -1.42915E-07 | -0.01999 |
| adolescent#002 | -0.000428201 | -1.43021E-07 | -0.00987 |
| adolescent#003 | -0.000187463 | -6.29647E-08 | -0.00785 |
| adolescent#004 | -0.000188523 | -1.12114E-07 | -0.00912 |
| adolescent#005 | -5.23529E-05 | -1.76362E-07 | -0.01109 |
| adolescent#006 | -8.65727E-10 | -2.96707E-11 | -0.01167 |
| adolescent#007 | -1.03457E-07 | -8.77117E-08 | -0.00846 |
| adolescent#008 | -3.34156E-10 | -8.98967E-12 | -0.00927 |
| adolescent#009 | -0.000118396 | -1.73358E-07 | -0.00774 |
| adolescent#010 | -2.237E-10 | -5.3542E-12 | -0.01215 |
| adult#001 | -0.000255779 | -8.80847E-08 | -0.01967 |
| adult#002 | -0.000762343 | -1.35421E-07 | -0.01966 |
| adult#003 | -4.93202E-10 | -1.32181E-07 | -0.01304 |
| adult#004 | -0.000187846 | -1.10494E-07 | -0.00892 |
| adult#005 | -0.000401528 | -1.12032E-07 | -0.01999 |
| adult#006 | -0.001015064 | -1.02666E-06 | -0.02417 |
| adult#007 | -0.002457841 | -9.76956E-06 | -0.0179 |
| adult#008 | -0.000164119 | -1.23146E-07 | -0.01839 |
| adult#009 | -0.0001885 | -1.64768E-07 | -0.01997 |
| adult#010 | -0.000165964 | -3.62289E-08 | -0.01791 |
| child#001 | -4.32616E-05 | -4.99315E-07 | -0.0012 |
| child#002 | -2.43848E-05 | -1.19047E-08 | -0.0063 |
| child#003 | -0.000114261 | -2.2317E-08 | -0.0019 |
| child#004 | -0.000122317 | -9.84608E-07 | -0.00171 |
| child#005 | -0.000144505 | -2.35487E-08 | -0.01025 |
| child#006 | -8.50475E-05 | -4.07014E-07 | -0.0017 |
| child#007 | -6.38112E-05 | -7.54145E-08 | -0.00464 |
| child#008 | -6.03971E-05 | -1.14231E-07 | -0.00226 |
| child#009 | -6.68974E-05 | -1.83219E-07 | -0.002 |
| child#010 | -8.80842E-06 | -5.85201E-08 | -0.00395 |

to PID-Har, and PID-OF, which extends PID-IF by individualizing OF values for each patient. These methods were applied to three patient groups: adolescents, adults, and children. The results indicate that for the children group, the standard PID method had an episode length of 95.76% (9.6 days) with a 95% confidence interval of 0.47, suggesting that only some episodes reached

**Table 4.3:** PID parameters and personalized observation frequency in minutes for each patient.

| Patient | Kp | Ki | Kd | OF |
|---|---|---|---|---|
| child#001 | -0.00015 | -1.50E-06 | -0.00084 | 27 |
| child#002 | -8.69E-06 | -1.10E-06 | -0.00588 | 18 |
| child#003 | -0.00027 | -2.81E-07 | -0.00151 | 27 |
| child#004 | -0.0001 | -8.21E-07 | -0.00181 | 9 |
| child#005 | -0.0009 | -6.29E-07 | -0.00905 | 15 |
| child#006 | -0.00034 | -3.06E-06 | -0.00095 | 36 |
| child#007 | -0.00033 | -1.73E-06 | -0.00281 | 21 |
| child#008 | -0.00025 | -1.15E-06 | -0.00153 | 27 |
| child#009 | -0.00021 | -2.13E-06 | -0.00091 | 18 |
| child#010 | -4.51E-05 | -2.85E-07 | -0.00231 | 9 |
| adolescent#001 | -0.00068 | -1.77E-06 | -0.01932 | 12 |
| adolescent#002 | -0.00085 | -1.13E-06 | -0.00712 | 33 |
| adolescent#003 | -0.00045 | -2.38E-06 | -0.00312 | 18 |
| adolescent#004 | -0.00088 | -2.60E-06 | -0.00583 | 27 |
| adolescent#005 | -0.00013 | -1.85E-06 | -0.00924 | 15 |
| adolescent#006 | -1.48E-09 | -6.18E-10 | -0.01188 | 9 |
| adolescent#007 | -4.06E-06 | -1.95E-06 | -0.00573 | 18 |
| adolescent#008 | -3.37E-09 | -1.40E-10 | -0.01018 | 21 |
| adolescent#009 | -5.97E-05 | -2.87E-06 | -0.00602 | 15 |
| adolescent#010 | -3.61E-09 | -1.03E-10 | -0.01184 | 9 |
| adult#001 | -0.0034 | -7.01E-07 | -0.01492 | 60 |
| adult#002 | -0.00117 | -2.71E-06 | -0.02286 | 33 |
| adult#003 | -1.22E-09 | -3.93E-06 | -0.00993 | 18 |
| adult#004 | -0.00038 | -2.40E-06 | -0.00355 | 15 |
| adult#005 | -0.00059 | -2.23E-06 | -0.02001 | 30 |
| adult#006 | -0.00132 | -4.03E-06 | -0.01327 | 18 |
| adult#007 | -0.00012 | -1.37E-05 | -0.00858 | 21 |
| adult#008 | -0.00155 | -3.28E-07 | -0.01366 | 45 |
| adult#009 | -0.00076 | -3.63E-06 | -0.01746 | 24 |
| adult#010 | -2.97E-05 | -3.36E-06 | -0.01301 | 18 |

**Table 4.4:** Comparison of the percentage of episode length by method and group.

| Method | Group | Avg. EP length | Confidence interval |
|---|---|---|---|
| PID | adolescents | 100 | - |
| | adults | 100 | - |
| | children | 95.76 | ± 0.47 |
| PID-Har | adolescents | 100 | - |
| | adults | 100 | - |
| | children | 100 | - |
| PID-IF | adolescents | 100 | - |
| | adults | 100 | - |
| | children | 100 | - |
| PID-OF | adolescents | 100 | - |
| | adults | 100 | - |
| | children | 100 | - |

10 days in the evaluation. On the other hand, the mean episode length was 100% (10 days in the evaluation) for all other methods and groups. This suggests that all PID control methods incorporating Harrison-Benedict meal generation were the most effective in controlling blood glucose levels for the children group.

### 4.2.2. Time in range and risk index

We ran simulations on all available patients over 20 rounds, with each round lasting for 10 days. From Fig. 4.1 and 4.2, it can be observed that each method has demonstrated improvement. The incorporation of Harrison-Benedict meal management, followed by insulin feedback (IF) and observation frequency (OF), resulted in an upward trend in the TIR as indicated by the chart on euglycemia. In the case of personalized OF in the PID-OF method, although its performance was comparable to that of PID-IF, the TIR and fluctuations in hypoglycemia showed a decrease in comparison to the other methods.

The risk index, as demonstrated in Fig. 4.3 and 4.4, also reflects the trend of TIR, indicating that the integration of supportive methods leads to a reduction in the risk index. Specifically, the PID-OF method demonstrates a notable decrease in the fluctuations of hypoglycemia in all age groups, leading to the conclusion that IF and OF play a significant role in the regulation of blood sugar levels through insulin administration. Although the performance of PID-IF and PID-OF is similar, the personalized OF in PID-OF further helps in reducing the fluctuations in the risk index to a greater extent.

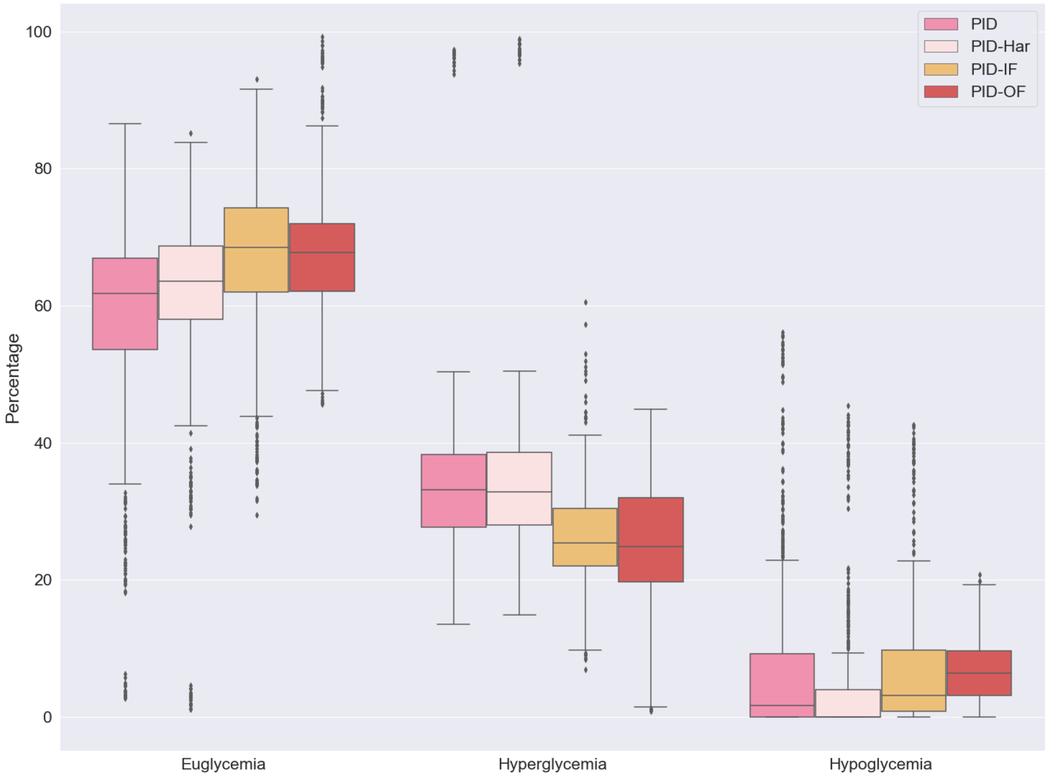


**Figure 4.1:** Comparative fraction of time spent in global glycemic states.

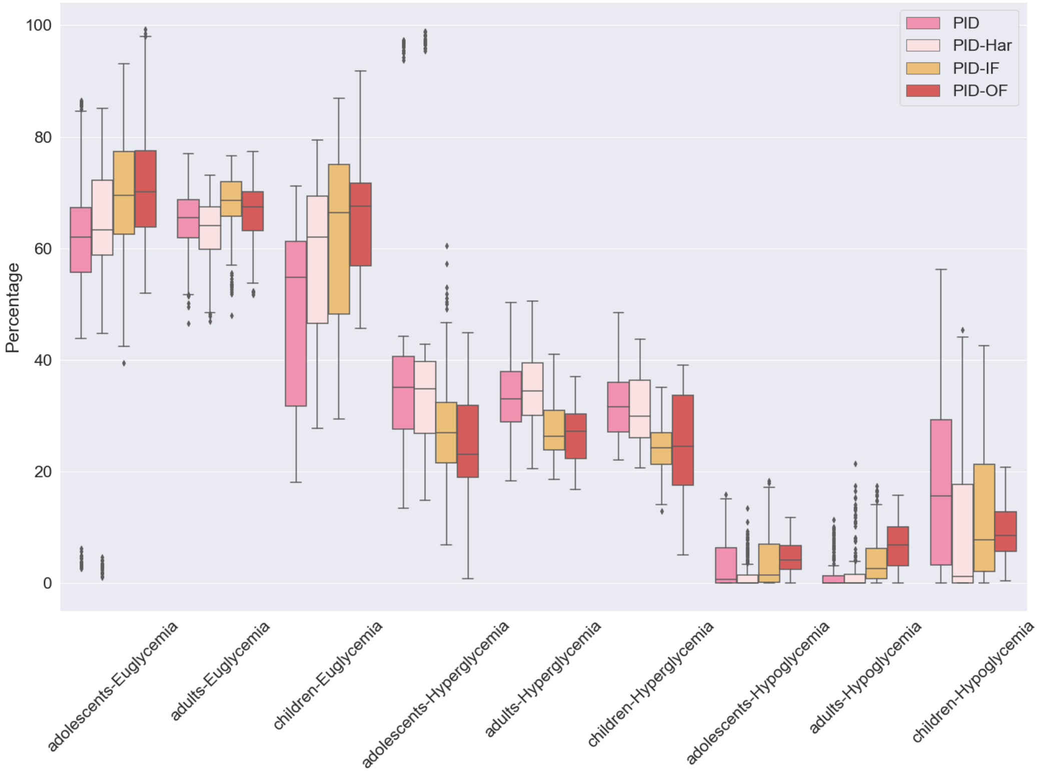


**Figure 4.2:** Comparative fraction of time spent in glycemic states by group.

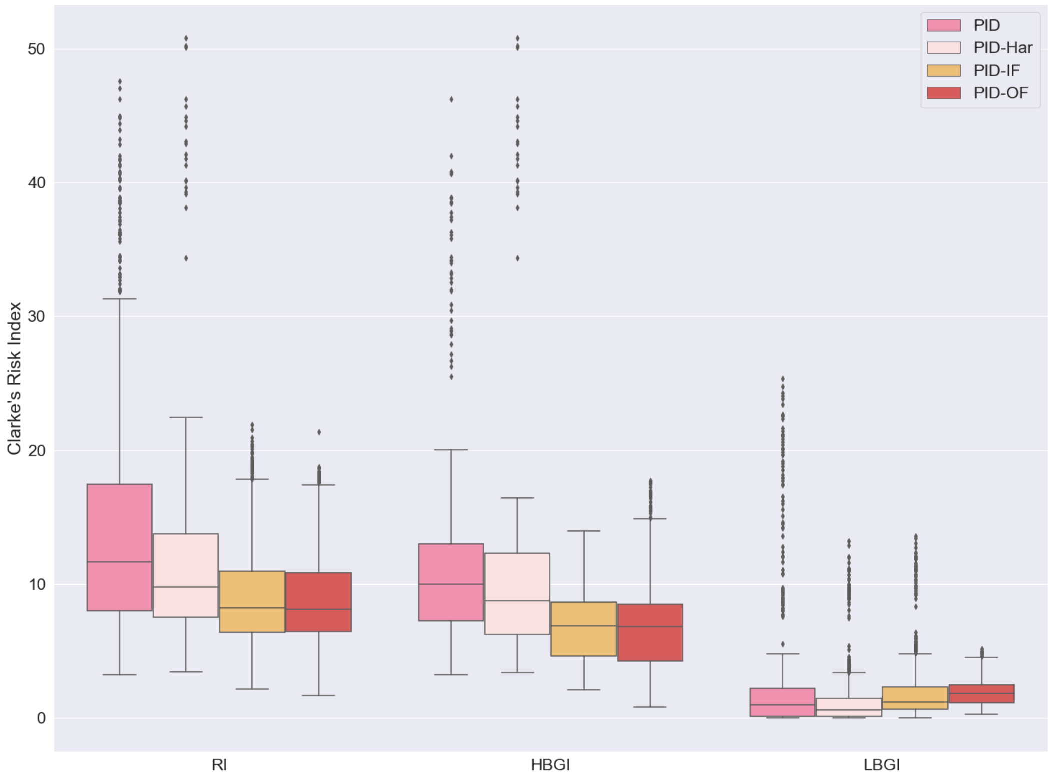


**Figure 4.3:** Comparative fraction of global risk index.

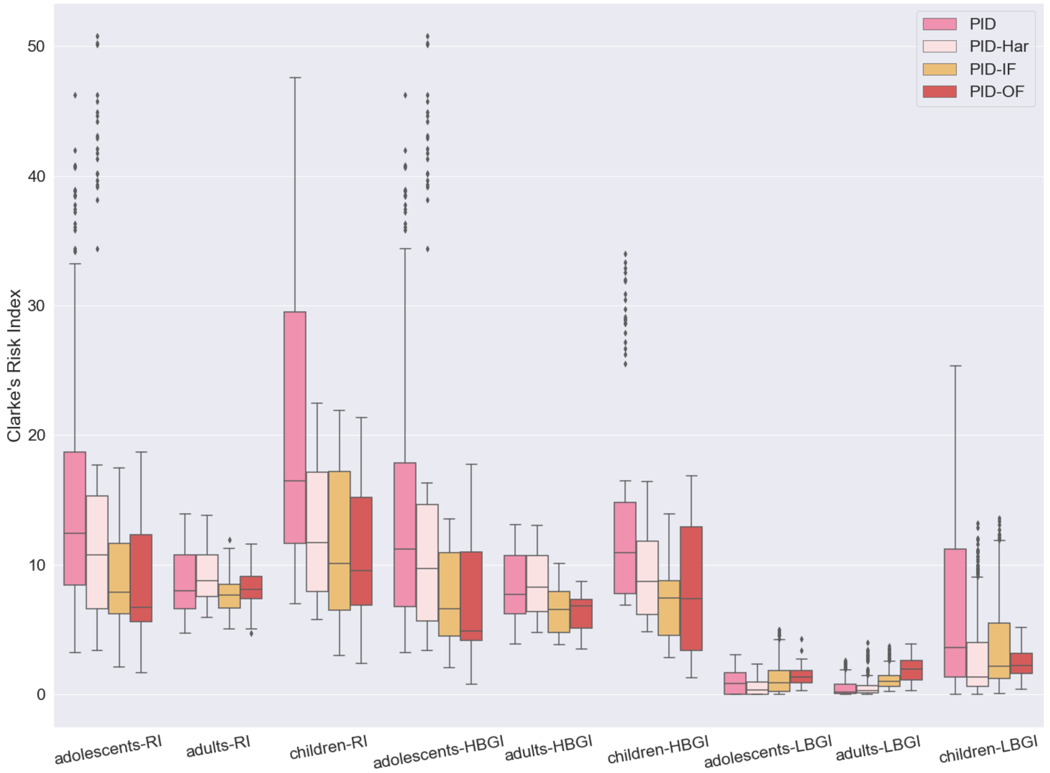


**Figure 4.4:** Comparative fraction of risk index by group.

### 4.2.3. PID with personalized OF over a 24-hour period for each patient

From Fig. 4.5, 4.6, and 4.7, it is observed that the average BG levels are depicted with PID-OF for each patient over a period of 24 hours. Three reference stripes with distinct colors have been used to differentiate between the risk levels of BG. The green stripe represents the TIR between 70-180 mg/dL, while the orange stripe represents BG levels between 181-250 mg/dL. The red stripe, on the other hand, represents BG levels below 69 mg/dL or above 250 mg/dL. The best performers in terms of BG control are adolescent#001 and child#005, as they maintain their BG levels within the TIR for the entire day. On the other hand, several patients from all age groups are found to have spent some time outside of the TIR, in either the orange or red zones. This observation highlights that PID remains only as a relatively effective method for BG control in all age groups.

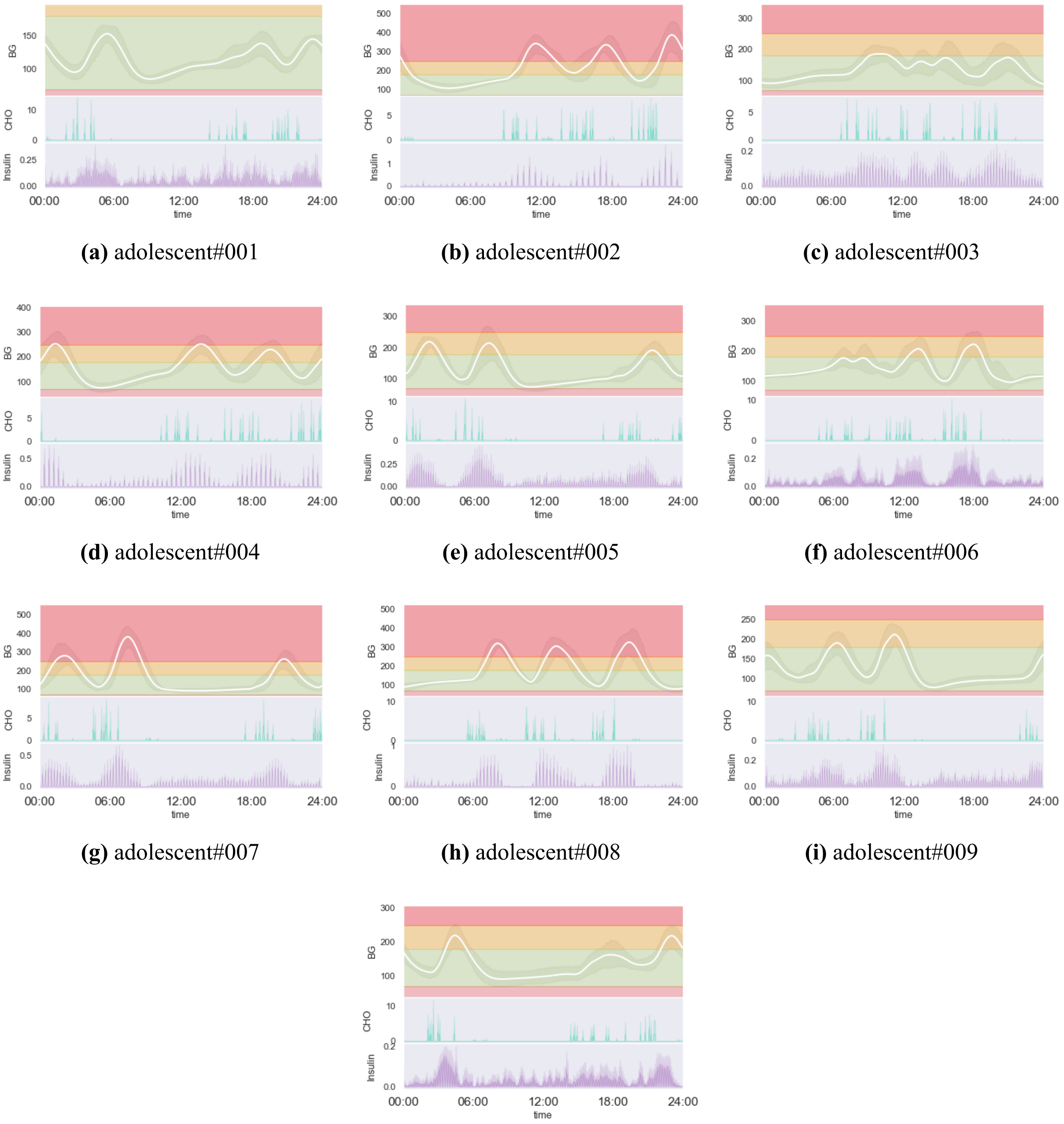
## 4.3. Discussion of results

The results presented in Table 4.4 and the accompanying figures indicate that a conventional PID approach can effectively regulate blood glucose levels to above 60% of TIR but it fails in children, who experience these events more frequently, making it difficult to maintain optimal control for more than eight days. We can conclude that incorporating Harrison-Benedict meal generation is effective for all groups and specially for children. The incorporation of insulin feedback (IF) and observation frequency (OF) in the PID-OF method leads to a reduction in the fluctuations of hypoglycemia and an upward trend in the target insulin range (TIR), as indicated by the euglycemia chart. The best performers in terms of BG control were adolescent#001 and child#005, who maintained their BG levels within the TIR for the entire day.

Our proposed PID-based blood glucose control system performs similarly to the PID system presented by Fox *et al.* [8] and Emerson *et al.* [18]. The target insulin range (TIR) values are comparable to their results, and in terms of catastrophic events, PID-OF outperforms their work, as we have not experienced any such events in PID-OF.

For a more comprehensive comparison, we reviewed recent studies on the commercial CGM and insulin pump system from Medtronic, namely the Minimed 640G and 670G, which utilize PID with insulin feedback [91] and have a reading and insulin-delivery interval of 5 minutes [92]. The results of Minimed 640G in adults show a TIR of about 59.5% with 4% hypoglycemia [93], while the Minimed 670G achieves a TIR of 67% with 2.8% hypoglycemia for adolescents, a TIR of 74% with 3.4% hypoglycemia for adults[94], and a TIR of 65% with 3% hypoglycemia for children [95]. Our results show that PID-OF keeps the median at similar or higher levels for all groups and can improve the TIR for some patients. Notably, this approach only requires a simple adjustment that is not difficult to implement.

These findings demonstrate the effectiveness of PID-IF and PID-OF methods for BG control;



**(a)** adolescent#001

**(b)** adolescent#002

**(c)** adolescent#003

**(d)** adolescent#004

**(e)** adolescent#005

**(f)** adolescent#006

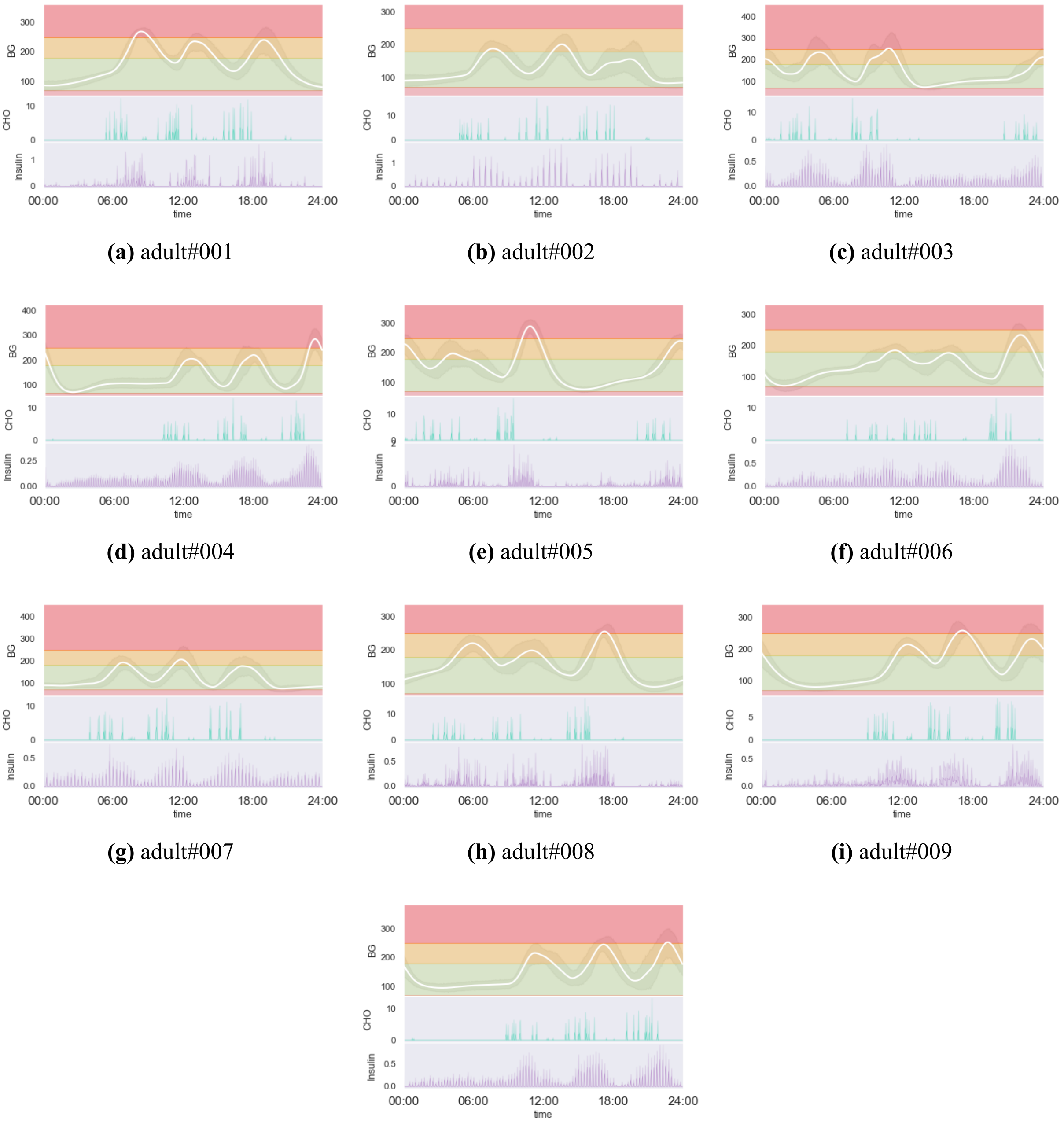
**(g)** adolescent#007

**(h)** adolescent#008

**(i)** adolescent#009

**(j)** adolescent#010

**Figure 4.5:** Average blood glucose level of adolescents over a day by using PID-OF method.



**(a)** adult#001

**(b)** adult#002

**(c)** adult#003

**(d)** adult#004

**(e)** adult#005

**(f)** adult#006

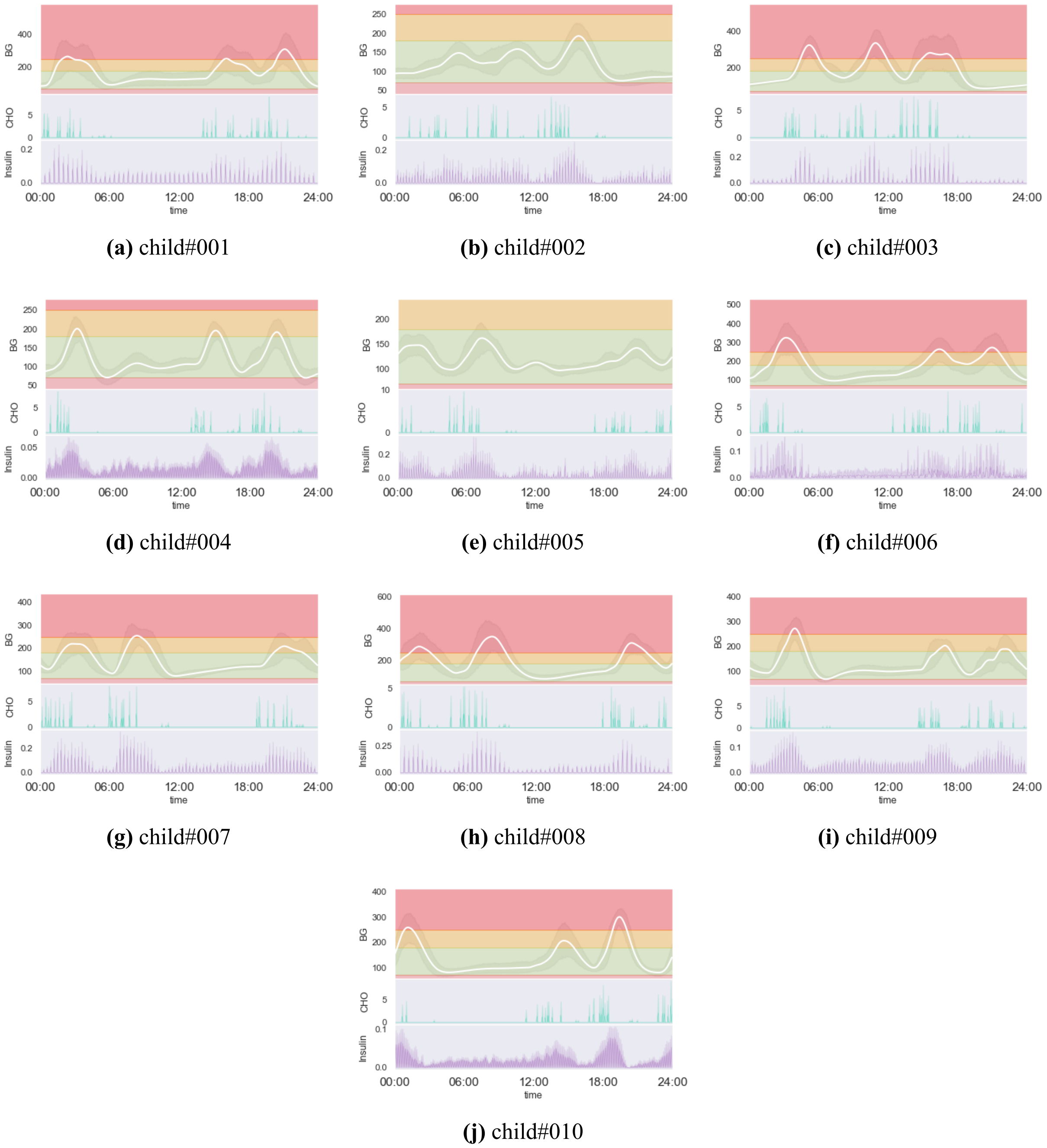
**(g)** adult#007

**(h)** adult#008

**(i)** adult#009

**(j)** adult#010

**Figure 4.6:** Average blood glucose level of adults over a day by using PID-OF method.



**(a)** child#001

**(b)** child#002

**(c)** child#003

**(d)** child#004

**(e)** child#005

**(f)** child#006

**(g)** child#007

**(h)** child#008

**(i)** child#009

**(j)** child#010

**Figure 4.7:** Average blood glucose level of children over a day by using PID-OF method.

however, it should be noted that most diabetics aim for a TIR of at least 70 percent of readings [18]. While PID is effective, it may not always achieve this level of control, highlighting the need to explore reinforcement learning methods. Given these findings, we use PID methods as our baselines in the next chapter as we continue to explore new and innovative methods for controlling blood glucose levels in patients with diabetes.

# 5

# Examining offline reinforcement learning for blood glucose control in T1D patients

In this chapter, we describe our evaluation of offline RL as a method for automatic BG control. We evaluate two offlineRL algorithms, *Decision Transformer* [34] (DT) and *Trajectory Transformer* [35] (TT). Each of the algorithms have been trained with two different sets of datasets, one generated by our previous online RL BG controller, PPO-RNN and PID-IF in Chapter 3. We also use those methods as baselines for comparison. In the remaining of the chapter, each combination is referred to as **Decis-PPO**, **Traj-PPO**, **Decis-PID-IF** and **Traj-PID-IF**, respectively. In addition, a dataset that mixes trajectories from both methods (**PPO-RNN** and **PID-IF**) is also used to evaluate both algorithms. As metrics used to determine whether the glycemic control algorithm works appropriately, we use the time percentage in euglycemia or Time in Range (TIR). In both cases they refer to the time spent in the target glycemic level range between 70 and 180 mg/dL. Lower (hypoglycemia) and higher ranges (hyperglycemia) may cause short-term and long-term complications in T1D. Most diabetics should aim for a TIR of at least 70 percent of readings [18].

We first describe the baselines and the experimental setup and then discuss our evaluation results. Our general goal is to determine whether offline RL is a feasible method for automated BG control and how the quality and size of the datasets influence the learning process.

# 5.1. Implementation strategy and methodology

## 5.1.1. Baselines

**Proximal Policy Optimization (PPO-RNN)**. In a previous Chapter 3, we proposed and evaluated a RL control based on the PPO algorithm [31]. One key finding of our previous work was that we were able to successfully train the agents if we selected a proper observation frequency for each type of patient, different from the default 3-minute CGM samples. That is, instead of using the default frequency of the CGM sensor, observations were made every 45, 30 and 15 minutes for adults, adolescents and children respectively. In addition, a simple reward function, shown in eq. (3.8), was used.

With this implementation strategy, we showed that the PPO agent outperforms other control methods and is able to keep over 73% of time in euglycemia across all groups.

**Proportional Integrative Derivative with Insulin feedback (PID-IF)**. In the previous Chapter 3, we also tested a PID control that aims to keep the BG level at a target point of 112.517 mg/dl, which is the zero-risk point in Clake's Risk Index. Note that PID-IF includes insulin feedback [11, 12]. Insulin feedback is an adjustment of insulin delivery that adapts to metabolism changes due to life activities and has been shown to improve the performance of PID controls.

In our previous chapter, we implemented the PID-IF control for the default observation frequency of three minutes for all patients. However, in this chapter we want to combine PPO-RNN trajectories with PID-IF for training the offline RL agents. Since the PPO-RNN agents use different observation frequencies for each group age, as discussed previously, we have to adapt the PID parameters, proportional, derivative and integral constants, $K_p$, $K_d$, and $K_i$, for that particular frequencies. We use Optuna, an automatic hyperparameter optimization framework [85], to find the optimal $K_p$, $K_d$, and $K_i$ values for each patient . The optimal PID parameters for each patient are provided in Table 4.2.

In summary, in this work both baselines, PPO-RNN and PID-IF, use the same observation frequency; 15, 30, and 45 minutes for children, adolescents, and adults, respectively. Finally, meals were randomly generated by the Harris-Benedict algorithm [8] and used along in data generation for training and evaluation.

## 5.1.2. Experimental setup

We used the open-source implementations of TT and DT, available at [34, 35]. For training and evaluation, we used the SimGlucose: python framework based on the UVA/PADOVA simulator, with 30 virtual patients divided into three groups: adults, adolescents and children, with 10 subjects each [41]. The parameters of patients were obtained from the academic edition of the commercial UVA/PADOVA simulator version 2008, according to the developer [79]. This simulator is based on the OpenAI Gym standard [44], which is compatible with RL algorithms and easy to adapt to various kinds of research. It also provides different types of CGM sensors, insulin pumps, and a random meal scheduler with noise. SimGlucose has been previously used in similar studies [6, 8, 42, 43]. We trained DT and TT with the datasets generated by our baselines previously described.

**Data gathering.** Initially, we generated three groups of datasets for training the offline RL agents.

Each dataset contains five features: observation, action, reward, terminal, and timeout. An observation is the current CGM state; an action is an amount of delivered insulin, and the reward is genereated according the reward function in eq. (3.8), described in [32]. A terminal is True when the patient's BG is under 10 or over 1,000 mg/dL, which is considered a *catastrohpic failure* and timeout is True when a patient *survived* for 10 days, that is, there was no catastrophic failure in the 10 days. In the first stage, we used the datasets generated from baselines - PPO and PID-IF. The size of each dataset is one million samples per patient, so we generated 30 million samples in total. The second stage considers a combination of PPO and PID-IF datasets, since we hypothesize that if we combine data from multiple sources, the agents may learn better. Thus, we sorted the datasets by the highest rewards and then mixed the datasets as follows: the first one with 80% samples from PPO and 20% from PID-IF 20%, and a second one with 50% of PPO and 50% of PID-IF. A new mixed dataset for each patient was generated. Finally, to test the influence of the dataset size in the learning process, in the final stage, we generated new datasets from the sorted baselines ones, by reducing the number of samples to one hundred thousand and ten thousand. In total, there are three groups of datasets for each patient: two baseline datasets, two combined datasets, and two reduced datasets.

**Training.** We trained the offline RL agents for each patient and dataset with the original hyperparameters from its code repositories [34, 35].

**Evaluation.** We evaluated all the offline RL agents and dataset combinations, as well as the baselines, using 20 simulation replications with different seeds, per patient. Each replication is run for 10-days of simulation time, so each episode is 10-days long. The observation frequency is 45 minutes, 30 minutes, and 15 minutes, for adults, adolescents and children, respectively. The termination due to catastrophic failure (BG level under 10 or above 1,000 mg/dL) is identical to the one used in the training process. TIR or euglycemia fraction of time as well as hyperglycemia, hypoglycemia fractions and Clarke's risk index [76] are the metrics used for evaluation and comparison between DT and TT with different datasets.

## 5.2. Evaluation of the system through simulation

Our first test is to determine whether offline RL agents are able to avoid catastrophic failures. To this purpose, we simulate all patients with the different models for ten days and look at the average episode length. Our result in Fig.5.1 shows that offline RL Trajectory and Decision Transformers cannot outperform PID-IF and online RL PPO-RNN, and cannot reach ten days as the baselines, which means that BG level reaches a value outside the 10-1000 mg/dL. **Traj-PPO** achieves the longest average episode length. It reaches an average episode length of over eight days of simulated time in every age group. There are notable differences for each group and method, without a clear trend. In the following sections we look at the fraction of time spent at each state during the episode and discuss reasons for this behavior.

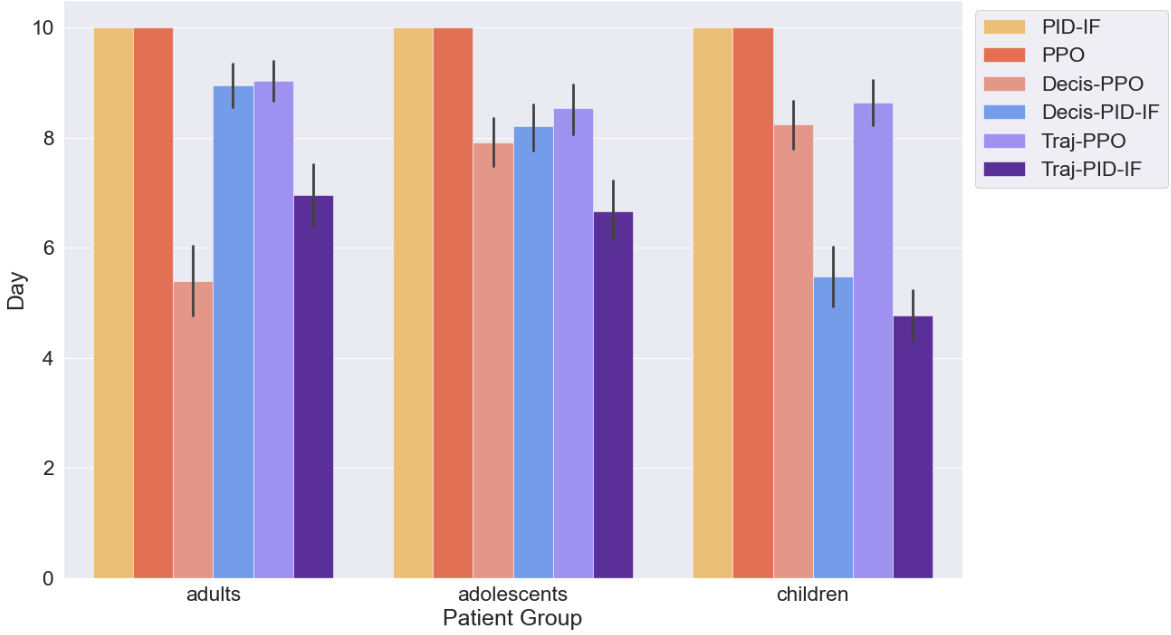


**Figure 5.1:** Fraction of completed 10-day evaluation reached for each method and group.

### 5.2.1. Risk index and glycemic states

We now compare the glycemic state, that is, the fraction of time spent in each BG range. In Fig. 5.2, we show all methods for all age groups. **Traj-PPO** achieves the highest median euglycemia of offline RL methods. Its median and 75 percentile slightly outperform the PID-IF baseline. On the contrary, when trained with PID-IF trajectories, **Traj-PID-IF**, it exhibits a poor performance. The performance of the Decision Transformer is bad with all the datasets tested. The results show clearly that offline RL cannot learn properly how to control with PID-IF trajectories. In fact, **Decis-PID-IF** has the highest hyperglycemia fraction, while **Traj-PID-IF** has the highest hypoglycemia fraction. We can see in Fig.5.3 the glycemic state by age group. **Traj-PPO** shows good performance across all age groups and even its median hyperglycemia in all groups is better than the original online PPO. However, its hypoglycemia median and 75 percentile are high and have a broad range, meaning that **Traj-PPO** implies a high low blood glucose risk, a serious concern in modern AP products. **Decis-PPO**, in its turn, shows unacceptable high ranges for both hypoglycemia in adults and hyperglycemia in adolescents and children.

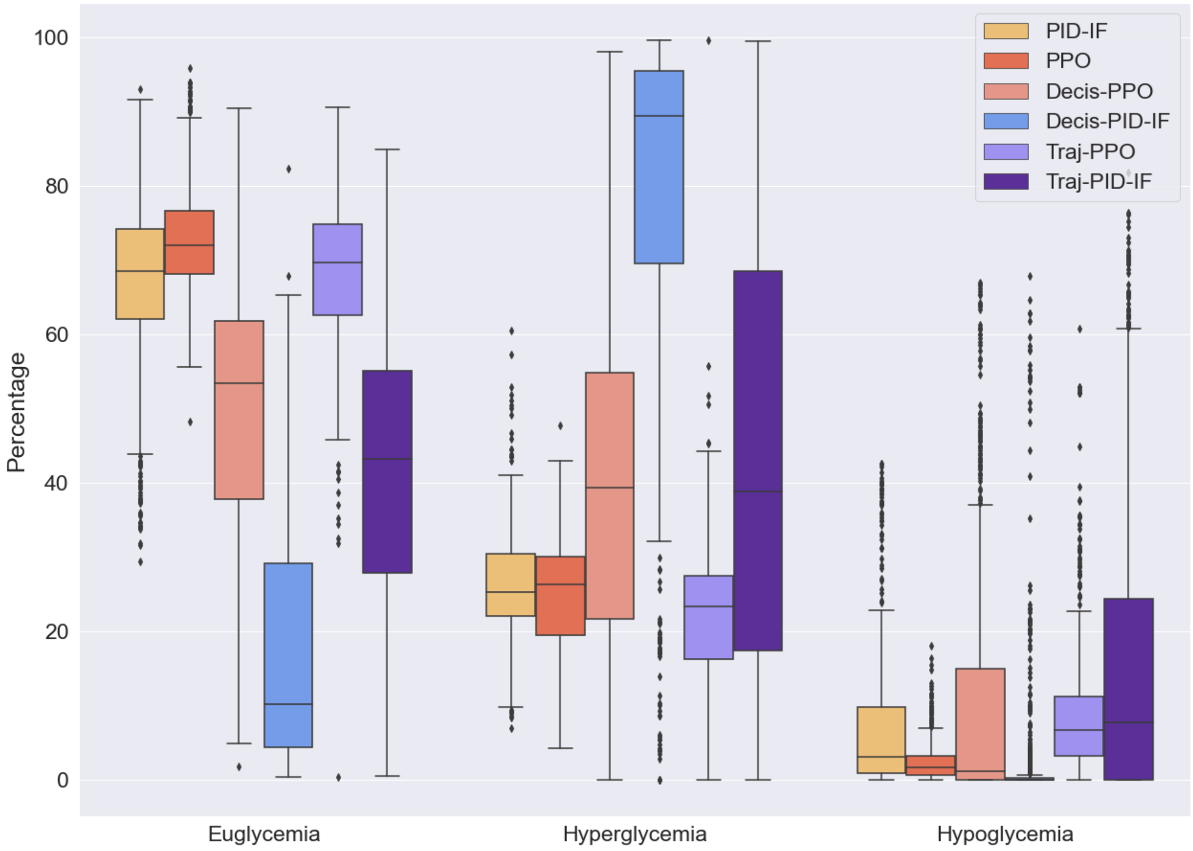


**Figure 5.2:** Comparative fraction of time spent in global glycemic state.

Actually, the risk index, evaluated in Fig. 5.4 and Fig.5.5, provides a more summarized view of the relative danger of hyper and hypoglycemic states, and shows that the riskiest method when attending to hyperglycemia is **Decis-PID-IF**, while hypoglycemia is more frequent in adults, adolescents, and children when using **Decis-PPO**, **Traj-PPO**, and **Traj-PID-IF**, respectively.

As as summary from this section we can conclude that **Traj-PPO** provides a level of performance similar to online PPO and PID-IF, but it has serious issues with hypoglycemia, that is, tends to inject too much insulin. In the following sections we come back to this matter.

### 5.2.2. Combination of PPO and PID-IF datasets

We compare the **Decis-PPO**, **Decis-PID-IF**, **Traj-PPO** and **Traj-PID-IF** with the combined datasets of PPO and PID-IF with two different ratios: eight to two (PP82) and five to five (PP55). In Table 5.1, we show the variation in percentage of the average episode length. We can see that the use of mixed datasets does not improve TT. On the contrary, it worsens its performance for all glycemic states. For DT, the mixed dataset slightly increases its performance for children and adolescents compared to **Decis-PID-IF**, and very clearly for adults. In Fig.5.6 and Fig.5.7 the global euglycemia in all methods is about the same level at 40%. However, the DT with both

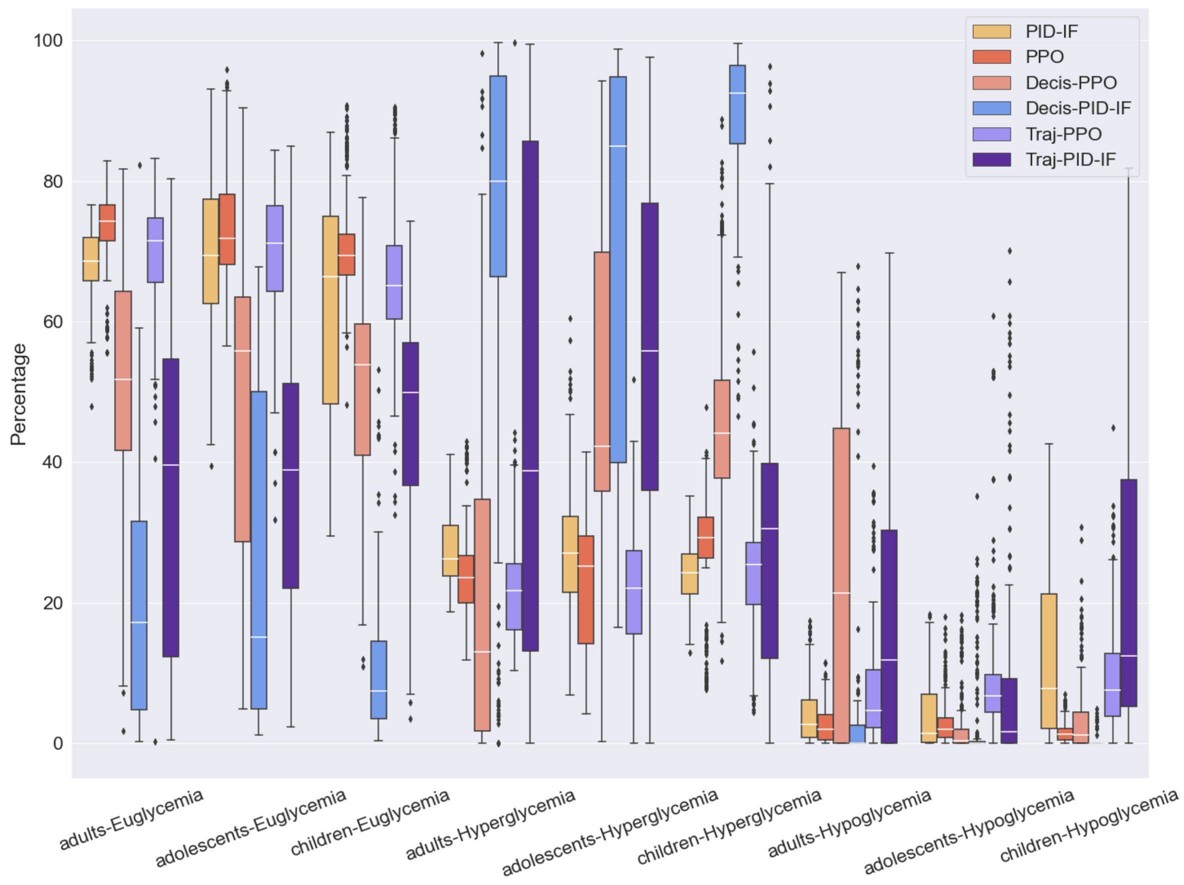


**Figure 5.3:** Comparative fraction of time spent in glycemic state by group.

**Table 5.1:** Increase/reduction of completed 10-day episodes of mixed datasets reaced for each method and group.

| Method | Group | PP55/PID | PP82/PID | PP55/PPO | PP82/PPO |
|---|---|---|---|---|---|
| **Trajectory** | children | -42.35% | -45.09% | -68.17% | -69.69% |
| | adolescents | -48.19% | -44.23% | -59.60% | -56.52% |
| | adults | -29.03% | -2.33% | -45.31% | -24.74% |
| **Decision** | children | 20.13% | 12.71% | -20.28% | -25.20% |
| | adolescents | 1.92% | 3.64% | 5.68% | 7.47% |
| | adults | 4.01% | 2.55% | 72.90% | 70.49% |

datasets performed well in avoiding hypoglycemia. TT has the same high and low glycemic risks. In terms of RI, from Fig.5.8 and Fig.5.9, we can see all DT and TT cases with mixed datasets range in 20-40 and they are outperformed only by the previous **Decis-PID-IF**.

In Table 5.2 we show the average daily dose of insulin injected by each method. As can be seen, there is a direct correlation, as expected, between the daily dose and the time spent at each glycemic state shown in Fig. 5.2. Moreover, in Table 5.3 we show, in percentage, whether the

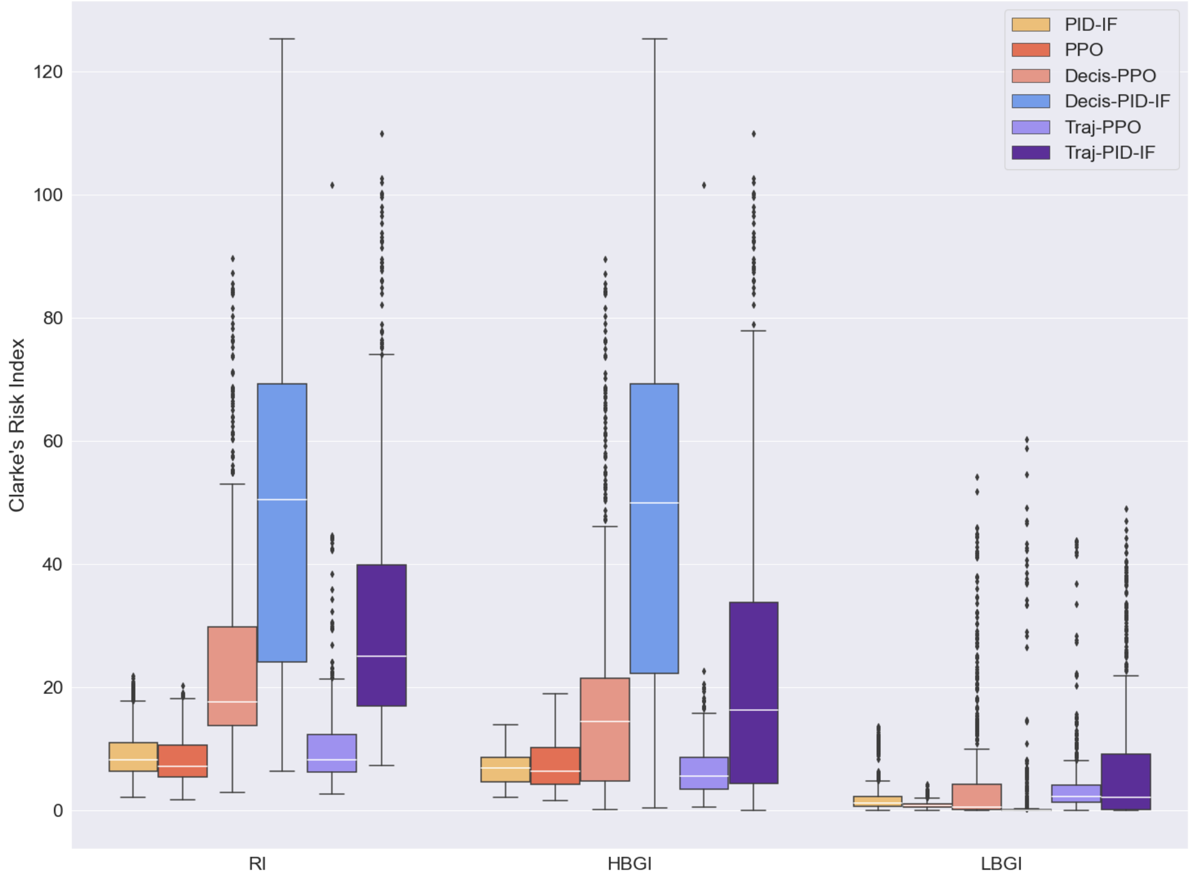


**Figure 5.4:** Comparative fraction of global risk index.

**Table 5.2:** Average daily insulin dose.

| **Method** | **Average daily dose** |
|---|---|
| **Decis-PID-IF** | 6.9 |
| **Decis-PP-55** | 7.7 |
| **Decis-PP-82** | 8.5 |
| **Decis-PPO** | 9.5 |
| **PID-IF** | 10.7 |
| **PPO** | 10.9 |
| **Traj-PID-IF** | 9.7 |
| **Traj-PP-55** | 16.3 |
| **Traj-PP-82** | 16.0 |
| **Traj-PPO** | 13.3 |

catastrophic events of each method are due to hyperglycemia or hypoglycemia.

From these data, we see that the average insulin dose of **Traj-PPO** is higher than that of PPO and that all the catastrophic events of **Traj-PPO** are due to hypoglycemia, while in **Decis-PPO**

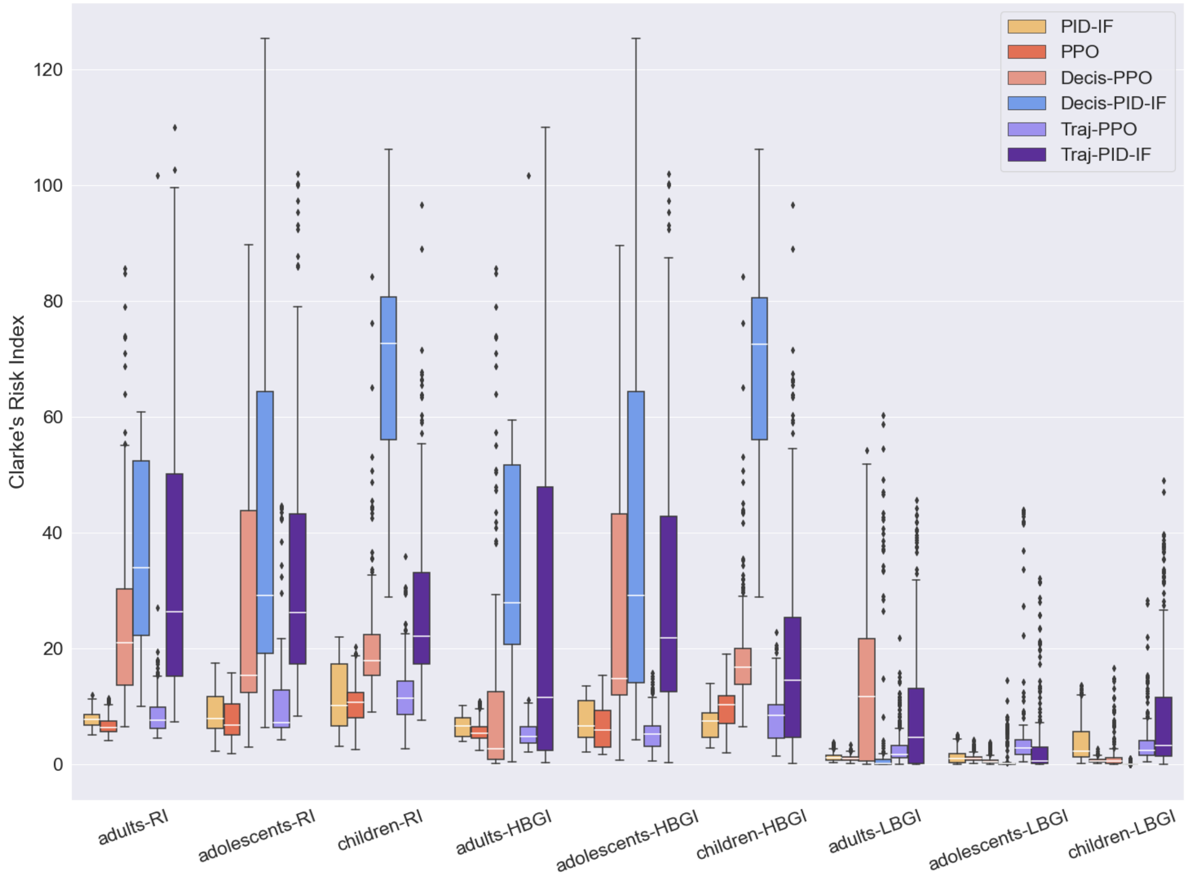


**Figure 5.5:** Comparative fraction of risk index by group.

**Table 5.3:** Type of catastrophic events by methods (boldface highlights the higher risk of each algorithm).

| Method | Hyper% | Hypo% |
|---|---|---|
| **Decis-PID-IF** | **80.10%** | 19.90% |
| **Decis-PP-55** | **84.18%** | 15.82% |
| **Decis-PP-82** | **82.70%** | 17.30% |
| **Decis-PPO** | **51.68%** | 48.32% |
| **Traj-PID-IF** | **58.11%** | 41.89% |
| **Traj-PP-55** | 25.11% | **74.89%** |
| **Traj-PP-82** | 35.76% | **64.24%** |
| **Traj-PPO** | 0.00% | **100.00%** |

they are practically balanced. When mixing the datasets, the proportion of catastrophic events due to hyperglycemia increases for all the methods.

With these tests our aim is to test if the offline agents may improve their performance when trained with a ”more” distributed dataset, that is, with a dataset with a potentially wider range of states and actions. Our results show that transformers cannot generalize adequately. We conclude that more care has to be put in selecting the trajectories for the datasets. For instance, when ordering the trajectories we just look at the highest rewards, but the average BG level of those trajectories is not taken into account. **Traj-PPO** only has catastrophic events due to hypoglycemia because it tends to keep patients on a low BG level. Due to our reward function, such kind of trajectories may have a reward which is high but equal to other trajectories that keep the patient on a higher BG level, which would be better. Such considerations have to be taken into account when creating the training datasets.

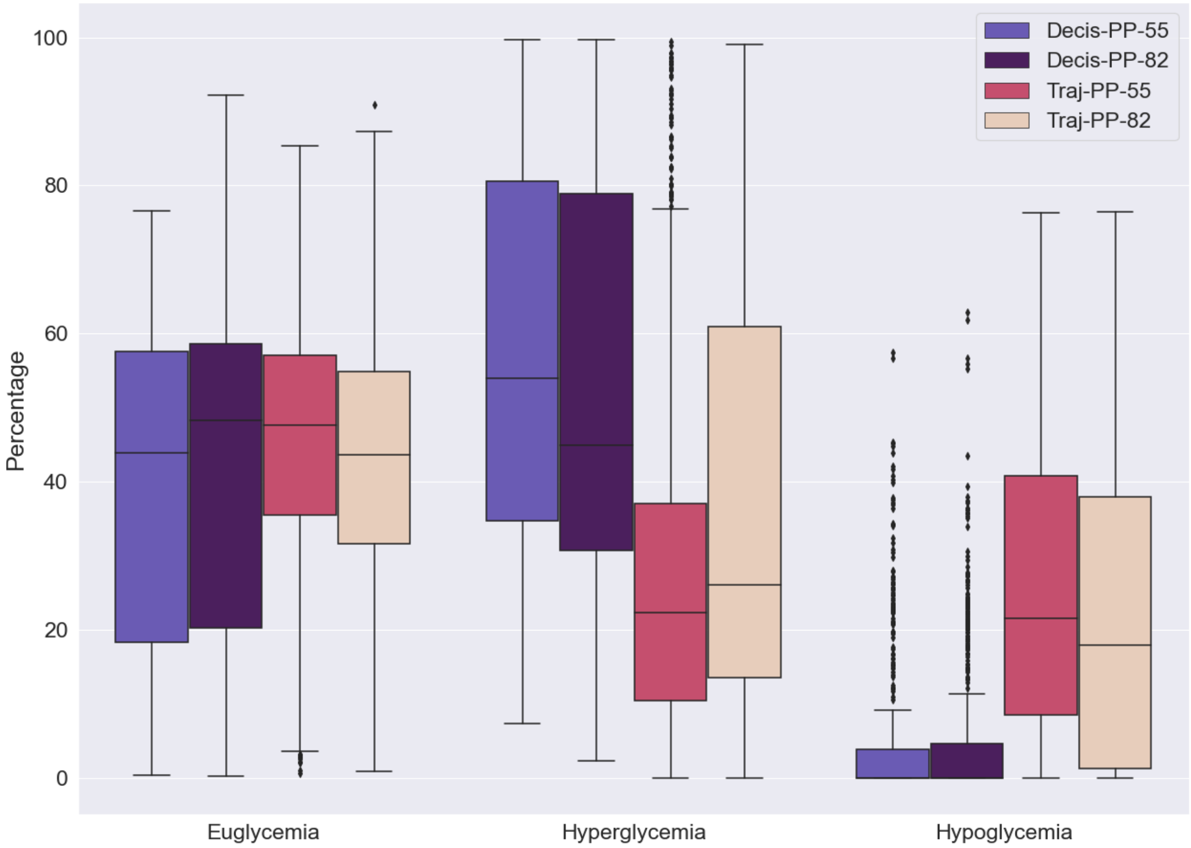


**Figure 5.6:** Comparative fraction of time spent in global glycemic state of mixed datasets

### 5.2.3. Dataset size

The dataset size is important because one cannot realistically expect to collect samples from patients for years and so we want to test how much we can reduce the dataset to get good enough results. Interestingly, from Table 5.4, both DT and TT with 10k size have longer episode lengths than 1M size on average. This is due to the fact that we sorted and use only the best trajectories. And the average euglycemia percentage is almost the same level as the 1M dataset. The difference in euglycemia for TT is 0.47% and 1.8% for DT. While hyperglycemia between 10k and 1M

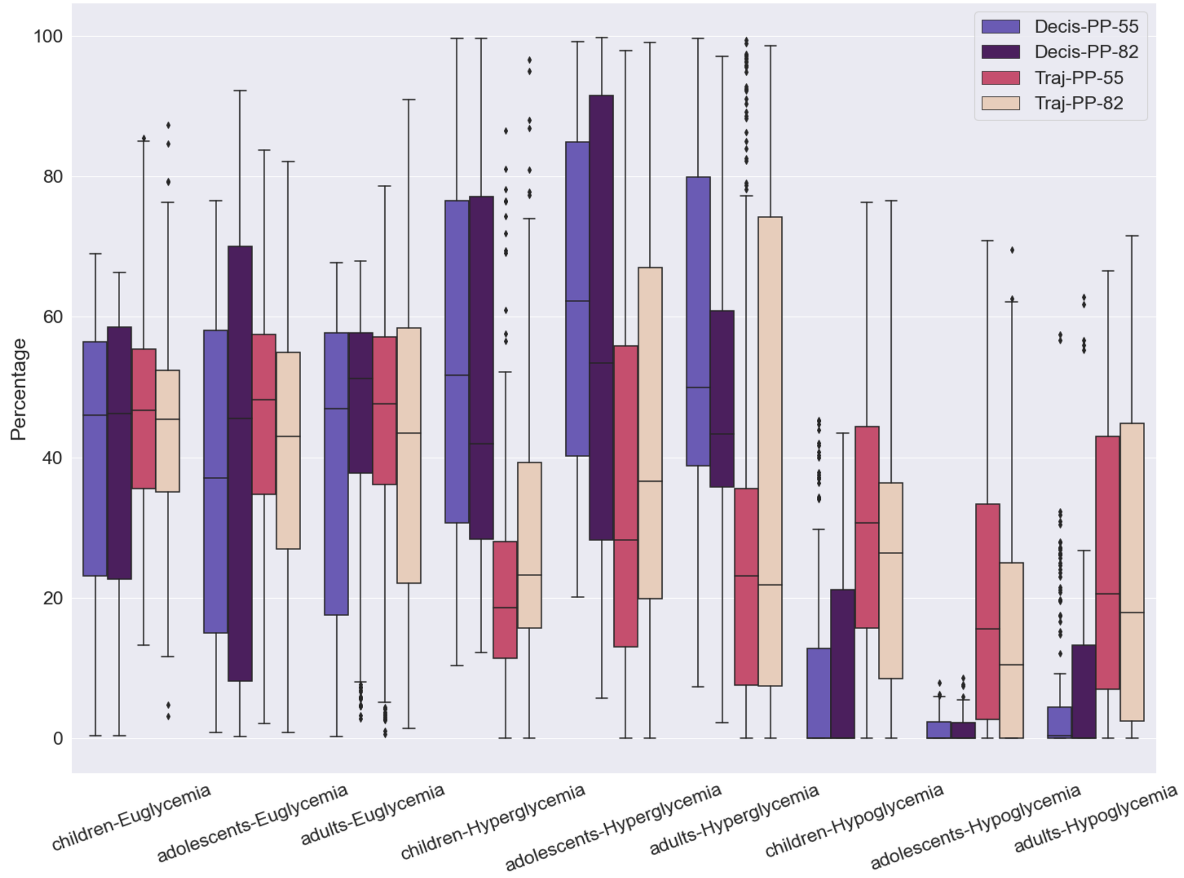


**Figure 5.7:** Comparative fraction of time spent in glycemic state of mixed datasets by age group.

datasets in TT decreases, in DT it increases by almost 10%. As a result, TT globally improves performance with 10k and has better RI than 1M, but DT with 10k slightly decreases an already poor performance. Clearly, DT and TT are less effective when the amount of data was reduced to 10k. Both methods had a decrease of more than 10% in TIR and a significant increase in RI.

**Table 5.4:** Evaluation of influence of dataset size (boldface highlights the best performance of each algorithm).

| Method | Trajectory transformers | | | Decision transformers | | |
|---|---|---|---|---|---|---|
| Dataset size | 10k | 100k | 1M | 10k | 100k | 1M |
| Episode Length | 81.50 ± 2.55 | **96.03 ± 1.28** | 87.37 ± 2.31 | 74.92 ± 2.90 | **86.69 ± 2.23** | 71.85 ± 3.05 |
| Euglycemia | 52.14 ± 1.36 | 67.80 ± 0.91 | **68.27 ± 0.84** | 40.79 ± 1.73 | 48.68 ± 1.72 | **50.48 ± 1.45** |
| Hyperglycemia | 36.00 ± 1.59 | 24.29 ± 0.91 | **22.47 ± 0.77** | 54.34 ± 1.88 | 48.84 ± 1.82 | **38.75 ± 2.04** |
| Hypoglycemia | 11.84 ± 0.70 | **7.91 ± 0.56** | 9.26 ± 0.77 | 2.85 ± 0.40 | **2.47 ± 0.31** | 10.76 ± 1.42 |
| Risk index | 20.24 ± 1.27 | **9.75 ± 0.50** | 10.41 ± 0.60 | 31.71 ± 1.97 | 24.39 ± 1.67 | **24.08 ± 1.45** |

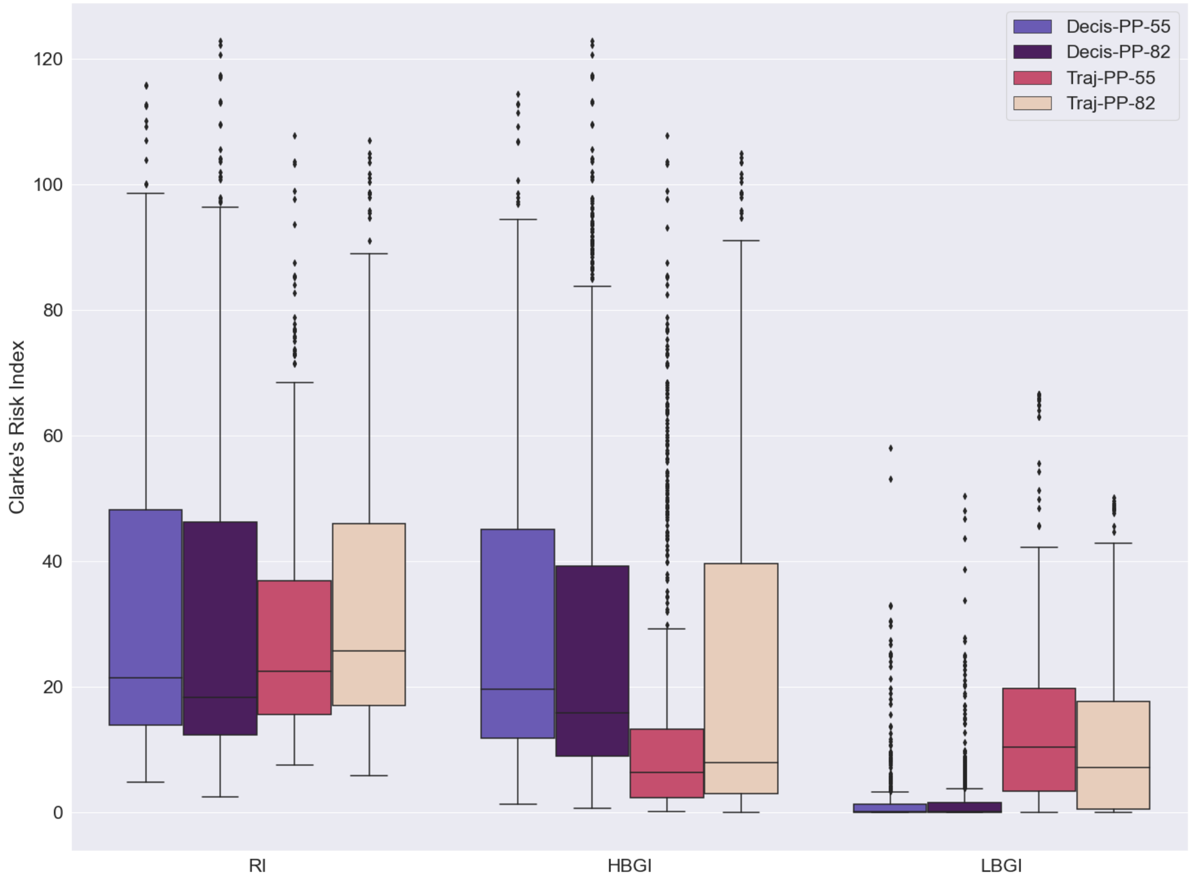


**Figure 5.8:** Comparative fraction of global risk indexes of mixed dataset.

## 5.3. Discussion of results

In the previous Chapter 3, and similar works [8], PID and PPO agents performed considerably well for BG control in the T1D simulator, so our hypothesis was that offline RL with these datasets should have comparable performance. Our results show that at least **Traj-PPO** has a performance similar to that of online PPO in most of the metrics, which is promising, since the main goal of this work is to determine whether offline RL can be a realistic alternative for data-driven BG control, before attempting clinical trials with real patient data. Our results also agree quantitatively with the work of [18], which shows a similar level of performance, although tested with fewer patients and different algorithms. Our evaluation also shows that training offline RL is not straightforward: neither all the algorithms tested nor the datasets used were equally effective in learning. It suggests that a better understanding of the influence of different data aspects and careful planning and design of the data-gathering is still necessary before collecting real-patient data for further tests, which is a complex and time-consuming task.

More research is needed to correct some of the observed deficiencies of offline RL methods. Most importantly, to prevent the inability to achieve full episode length without catastrophic

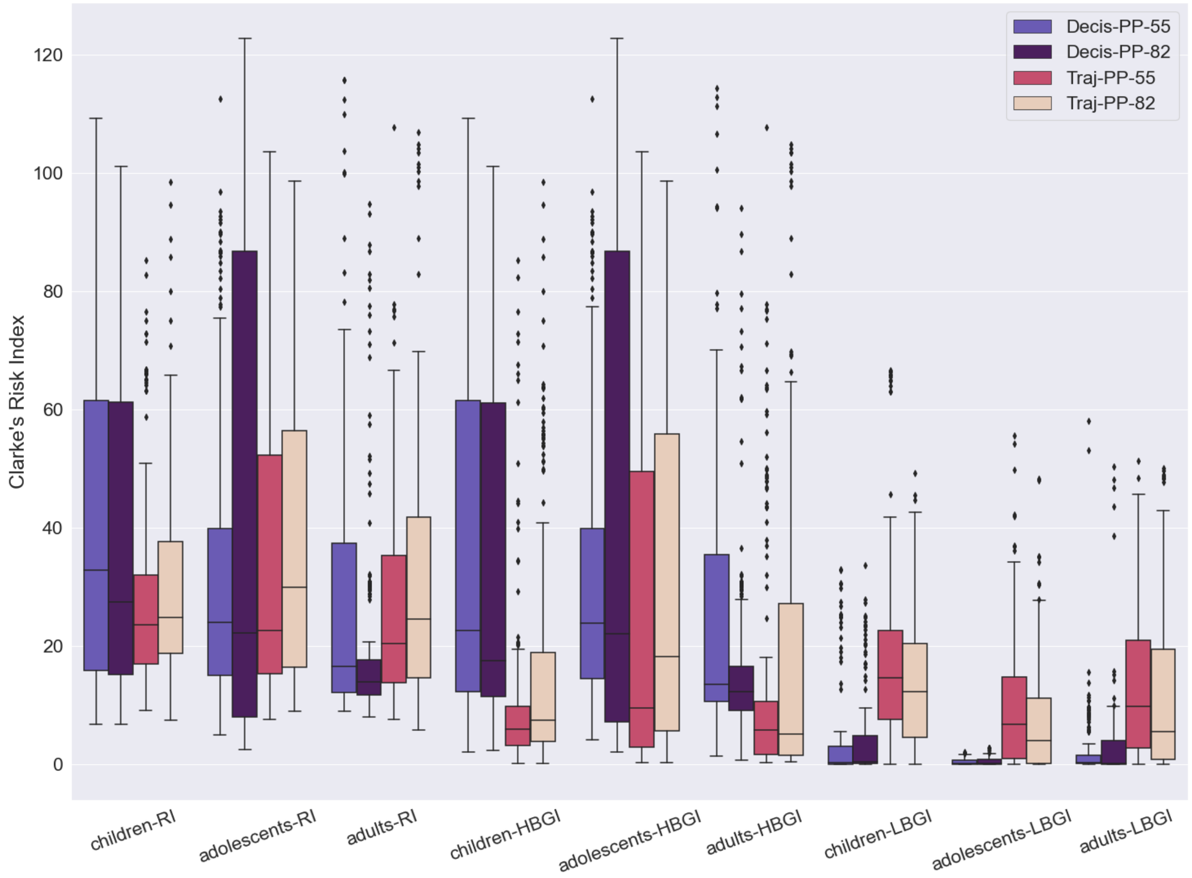


**Figure 5.9:** Comparative fraction of risk index of mixed dataset by group.

failures. Unlike the baselines, the average episode length cannot reach the full 10 days, even though the best one, **Traj-PPO**, reaches almost nine days globally.

In Table. 5.3, the catastrophic event of **Traj-PPO** is 100% due to hypoglycemia, while no catastrophic hyperglycemia occurred. Thus, additional research is needed to ensure that **Traj-PPO** is able to avoid hypoglycemia and thus able to achieve the full episode length and higher TIR. A direct next step is to further improve the quality of the training dataset to avoid hypoglycemic trajectories, as discussed below.

From our results, it is also clear that DT is not able to deliver good performance in this task, showing unacceptable high hyperglycemia levels in some groups. A simple reason may be that we have not optimized the DT hyperparameters, in particular, the minibatch sequence length, to which DT is sensitive for several tasks.

But there may be the need for deeper adaptations, such as pretraining or architectural changes, which have been shown to improve the basic DT performance [96, 29, 97, 98]. We leave the improvement of DT behavior as future work. Training with the PID-IF dataset did not yield satisfactory results for any of the algorithms. It seems that PID-IF generates too many out-of-distribution samples, that is, actions that move the state to not previously seen states which degrade the performance [33].

We sorted data by reward and length of the episode, then combined sorted PID and PPO datasets to determine if we can improve the learning process of the offline RL. Unfortunately, just a crude mixing, even with sorted trajectories, is not enough to improve the performance. It was partially effective with DT, slightly improving an already quite bad performance. It suggests that it may have potential but our results also imply that it is actually the quality of the datasets what actually brings the improvements.

In fact, the importance of having good trajectories is obvious: if the dataset size is reduced but only the best trajectories are kept, the performance can be even improved. The average TIR in the 10k-sample dataset is at a value similar to the 1M-sample datasets. The episode length is increased because of sorting trajectories and keeping the best ones, which can be seen in the results obtained from combining datasets and dataset size reduction. However, offline RL algorithms can not learn from datasets when the size is down to 10k samples. We have found a good trade-off with a dataset size of 10k samples, which also agrees with the work of [18] and [72]. But we may further improve the results by filtering appropriately the datasets, that is keeping the best ones, and removing the trajectories with undesirable characteristics. For instance, removing the trajectories that result in high hypoglycemic and hyperglycemic fractions, even if they have a good accumulated reward. This can be done by shrinking the target TIR, for example, to be in the range of 90-100 mg/dL. Alternatively, we can redesign the reward function to punish more hypoglycemia and high hyperglycemia.

Although the offline RL with Transformers architecture does not outperform clearly the baselines, the main advantage of offline RL is that it does not require interaction with the environment, as compared to online RL, which needs to interact with the patient to collect data for training. Offline RL emerges therefore as a safer and promising alternative for RL, being a practical application of automated and customized glycemic control.

# 6
# Conclusion and future lines

## 6.1. Conclusion

In this thesis, we propose a closed-loop BG level control based on DRL. We discuss the particular characteristics of a realistic simulator of the glucoregulatory system as a training environment for DRL agents and the complexity of their training in this environment. Effective training of such agents can be achieved by very different design choices for the learning process, which we call the implementation strategy.

We describe the initial evaluation of several alternatives conducted on a T1D simulator by UVA/PADOVA and, based on the results, propose a particular DRL implementation strategy based on reducing the frequency of the observations and rewards passed to the agent, and using a simple reward function. PPO-RNN agents are trained with that strategy for three groups of patients, evaluated and compared with PID, PID-IF, BB, BB-CD baselines. Our system is able to outperform common PID and BB strategies in overall terms, attending to healthy glycemic states and the risk index. A critical discussion of the results and a comparison with several recent works is provided, indicating that our system outperforms current solutions at a lower computational cost. Euglycemia is maintained in 73% of the time, and no early termination events (BG out of range) are reported. Hence, our results show DRL as a promising methodology for implementing closed-loop BG control.

In Chapter 4, we investigated the impact of incorporating individualized observation frequency

(OF) into the PID control algorithm for blood glucose control in type 1 diabetes. We found that optimizing the OF can significantly improve the performance of the PID controller for some patients, while it can maintain similar or higher median blood glucose levels for all patients. Our results also showed that tuning the OF is a simple and effective method to enhance the performance of the PID controller, which is widely used due to its simplicity and robustness.

Additionally, in Chapter 5, we have carried out a thorough evaluation of two recent offline RL algorithms for automated BG control of T1D patients. We have evaluated the influence on training and performance of the method that generated the datasets, as well as the influence of the type of trajectories used (single-method or mixed trajectories), the quality of the trajectories and the size of the datasets, and compared it with typically used baselines: PID and online RL methods.

Our results show that a Trajectory offline RL trained with a previous optimal PPO agent data performs at the level of the baselines, which supports that offline RL can be a realistic alternative for data-driven BG control with the advantage of not requiring real interaction with patients.

## 6.2. Future lines

The application of RL to control BG in type 1 diabetes involves a scenario with many tuning options and implementation strategies. In this context, the following potential improvements are identified, particularly for online RL:

- Hyperparameter optimization: In order to further improve the performance of the proposed method, it would be beneficial to conduct additional experiments that test various configurations of the RNN architecture and fine-tune the hyperparameters. By systematically exploring these design choices, we can gain a better understanding of how different settings affect the method's ability to control blood glucose levels and identify the most effective configurations.
- Reward function refinement: One area where the method could be refined is in the specification of the reward function. Specifically, we could revise the function to more strongly incentivize euglycemia and encourage tighter control over the desired range of BG levels. By experimenting with different reward functions, we can identify a function that strikes the right balance between these competing objectives and leads to better overall performance.
- Observation frequency optimization: The frequency of observations is a crucial factor in the success of the method, as it determines how often the system receives feedback on its performance and updates its control policy accordingly. Therefore, it would be useful to investigate different observation frequencies to determine the optimal trade-off between accuracy and efficiency. By varying the observation frequency and measuring its impact on performance, we can identify the frequency that yields the optimal results in practice.

- Alternative implementation strategies: Although the proposed method shows potential, it is possible that alternative implementation strategies could lead to even better results. For instance, we could explore different approaches to modeling the POMDP and input-dependent environments, such as using different types of machine learning algorithms or applying different regularization techniques. By experimenting with different strategies and comparing their performance to the proposed method, we can identify areas for improvement and potentially develop more effective solutions.

There is significant potential for improvement in the application of offline RL to BG control. The next phase of research in this area should aim to further optimize existing methods in order to achieve normal levels of BG control. The next lines are found of interest in this framework:

- Optimizing hyperparameters of current methods: It can improve the performance of offline RL algorithms for BG control. This involves finding the best combination of model parameters that result in the most effective control of BG levels. However, this process can be time-consuming and requires careful experimentation to ensure that the optimal hyperparameters are identified.
- Customized training datasets: Utilizing patient-specific CGM data to create a customized training dataset for offline RL presents several questions that need to be addressed. Currently, the best performing model is generated from a simulated environment and an optimal agent that was previously trained in the same environment. However, to create a real-patient dataset, CGMs and insulin doses must be collected, which may not be optimal in the first place.
- Mixed trajectories for training datasets: Another approach tested involves generating training datasets by combining trajectories from different sources such as real patient data and an optimized agent from a simulated environment customized to the patient group. However, the results have not been satisfactory, and further research is required to determine the best approach.
- Consideration of dual-hormone and triple-hormone artificial pancreas systems: Evaluate the performance of RL methods in a dual-hormone artificial pancreas system in which insulin is used in conjunction with glucagon in the BG control task [99]. Also, investigate the use of a novel dual-hormone artificial pancreas system that combines insulin and pramlintide [100]. Even, considering the extension of the system to a triple-hormone artificial pancreas in which insulin, glucagon, and pramlintide are used in the BG regulation process.
- Testing on virtual patients and limitations clarification: The feasibility of direct experimentation on real patients is limited by ethical and practical considerations, making proof-of-concept testing on virtual patients the only option [84]. However, transferring the results to real patients remains a challenge. Before conducting clinical trials with healthcare profes-

sionals, it is crucial to clarify any limitations and issues in the current methods to ensure meaningful and useful results.

Furthermore, in order to improve the representation of the heterogeneous population of patients with Type 1 diabetes for both online and offline RL, it would be valuable to generate additional sets of virtual patients and validate them against clinical data. By simulating a wide range of patient characteristics and medical histories, we can create a more diverse set of training examples that better reflect the variability of real-world patients. This, in turn, can lead to a more robust and effective control method.

# References

[1] *International Diabetes Federation Diabetes Atlas, 9th edition*. International Diabetes Federation, 2019. URL: `https://www.diabetesatlas.org` (visited on 02/03/2021).

[2] B. Wayne Bequette, Faye Cameron, Bruce A. Buckingham, David M. Maahs, and John Lum. “Overnight Hypoglycemia and Hyperglycemia Mitigation for Individuals with Type 1 Diabetes: How Risks Can Be Reduced”. In: *IEEE Control Systems Magazine* 38.1 (2018), pp. 125–134. URL: `https://doi.org/10.1109/MCS.2017.2767119`.

[3] Shadi Khodakaramzadeh, Yazdan Batmani, and Nader Meskin. “Automatic blood glucose control for type 1 diabetes: A trade-off between postprandial hyperglycemia and hypoglycemia”. In: *Biomedical Signal Processing and Control* 54 (Sept. 2019), p. 101603. URL: `https://doi.org/10.1016/j.bspc.2019.101603`.

[4] Amar Singh. *A basal-bolus injection regimen involves taking a number of injections through the day.* Oct. 2022. URL: `https://www.diabetes.co.uk/insulin/basal-bolus.html`.

[5] Eleni Bekiari et al. “Artificial pancreas treatment for outpatients with type 1 diabetes: systematic review and meta-analysis”. In: *BMJ (Clinical research ed.)* (2018). URL: `https://doi.org/10.17863/CAM.23542`.

[6] Miguel Tejedor, Ashenafi Zebene Woldaregay, and Fred Godtliebsen. “Reinforcement learning application in diabetes blood glucose control: A systematic review”. In: *Artificial Intelligence in Medicine* 104 (Apr. 2020), p. 101836. URL: `https://doi.org/10.1016/j.artmed.2020.101836`.

[7] Melanie K Bothe et al. “The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas”. In: *Expert Review of Medical Devices* 10.5 (Sept. 2013), pp. 661–673. URL: `https://doi.org/10.1586/17434440.2013.827515`.

[8] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. “Deep reinforcement learning for closed-loop blood glucose control”. In: *Machine Learning for Healthcare Conference*. PMLR. 2020, pp. 508–536.

[9] Sara Trevitt, Sue Simpson, and Annette Wood. “Artificial Pancreas Device Systems for the Closed-Loop Control of Type 1 Diabetes”. In: *Journal of Diabetes Science and Technology* 10.3 (Nov. 2016), pp. 714–723. URL: `https://doi.org/10.1177/1932296815617968`.

[10] Melissa J. Schoelwer and Mark D. DeBoer. “Artificial Pancreas Technology Offers Hope for Childhood Diabetes”. In: *Current Nutrition Reports* 10.1 (Jan. 2021), pp. 47–57. URL: `https://doi.org/10.1007/s13668-020-00347-9`.

[11] Lauren M. Huyett, Eyal Dassau, Howard C. Zisser, and Francis J. Doyle. “Design and Evaluation of a Robust PID Controller for a Fully Implantable Artificial Pancreas”. In: *Industrial & Engineering Chemistry Research* 54.42 (June 2015), pp. 10311–10321. URL: `https://doi.org/10.1021/acs.iecr.5b01237`.

[12] Cesar C. Palerm. “Physiologic insulin delivery with insulin feedback: A control systems perspective”. In: *Computer Methods and Programs in Biomedicine* 102.2 (May 2011), pp. 130–137. URL: `https://doi.org/10.1016/j.cmpb.2010.06.007`.

[13] M. Di Ferdinando, P. Pepe, S. Di Gennaro, and P Palumbo. “Sampled-Data Static Output Feedback Control of the Glucose-Insulin System”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 3626–3631. URL: `https://doi.org/10.1016/j.ifacol.2020.12.2044`.

[14] Alessandro Borri, Giordano Pola, Pierdomenico Pepe, Maria Domenica Di Benedetto, and Pasquale Palumbo. “Symbolic Control Design of an Artificial Pancreas for Type-2 Diabetes”. In: *IEEE Transactions on Control Systems Technology* (2021), pp. 1–16. URL: `https://doi.org/10.1109/tcst.2021.3135320`.

[15] Revital Nimri et al. “Feasibility Study of a Hybrid Closed-Loop System with Automated Insulin Correction Boluses”. In: *Diabetes Technology & Therapeutics* 23.4 (Apr. 2021), pp. 268–276. URL: `https://doi.org/10.1089/dia.2020.0448`.

[16] Eric Renard, Jerome Place, Martin Cantwell, Hugues Chevassus, and Cesar C Palerm. “Closed-loop insulin delivery using a subcutaneous glucose sensor and intraperitoneal insulin delivery: feasibility study testing a new model for the artificial pancreas”. In: *Diabetes care* 33.1 (2010), pp. 121–127.

[17] Alexander James Barnes and Richard W. Jones. “PID-based Glucose Control using Intra-Peritoneal Insulin Infusion: An In Silico Study”. In: *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. 2019, pp. 1057–1062. URL: `https://doi.org/10.1109/ICIEA.2019.8833728`.

[18] Harry Emerson, Matt Guy, and Ryan McConville. *Offline Reinforcement Learning for Safer Blood Glucose Control in People with Type 1 Diabetes*. 2022. URL: `https://arxiv.org/abs/2204.03376`.

[19] Lalo Magni et al. “Model Predictive Control of Type 1 Diabetes: An in Silico Trial”. In: *Journal of Diabetes Science and Technology* 1.6 (Nov. 2007), pp. 804–812. URL: `https://doi.org/10.1177/193229680700100603`.

[20] Taku Yamagata et al. “Model-Based Reinforcement Learning for Type 1 Diabetes Blood Glucose Control”. In: *Singular Problems for Healthcare Workshop at ECAI 2020 ; Conference date: 29-08-2020 Through 08-09-2020*. June 2020, pp. 1–14. URL: `http://www.smarttechresearch.com/SP4HC2020/`.

[21] Cari Berget, Samantha Lange, Laurel Messer, and Gregory P Forlenza. “A clinical review of the t:slim X2 insulin pump”. In: *Expert Opinion on Drug Delivery* 17.12 (2020), pp. 1675–1687.

[22] MS Ibbini and MA Masadeh. “A fuzzy logic based closed-loop control system for blood glucose level regulation in diabetics”. In: *Journal of Medical Engineering & Technology* 29.2 (Jan. 2005), pp. 64–69. URL: `https://doi.org/10.1080/03091900410001709088`.

[23] Torben Biester et al. “The automated pancreas: A review of technologies and clinical practice”. In: *Diabetes, Obesity and Metabolism* 24.S1 (Nov. 2021), pp. 43–57. URL: `https://doi.org/10.1111/dom.14576`.

[24] Phuong D. Ngo, Susan Wei, Anna Holubová, Jan Muzik, and Fred Godtliebsen. “Control of Blood Glucose for Type-1 Diabetes by Using Reinforcement Learning with Feedforward Algorithm”. In: *Computational and Mathematical Methods in Medicine* 2018 (Dec. 2018), pp. 1–8. URL: `https://doi.org/10.1155/2018/4091497`.

[25] Gavin Robertson, Eldon D. Lehmann, William Sandham, and David Hamilton. “Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-Concept Pilot Study”. In: *Journal of Electrical and Computer Engineering* 2011 (2011), pp. 1–11. URL: `https://doi.org/10.1155/2011/681786`.

[26] Ian Fox and Jenna Wiens. *Reinforcement learning for blood glucose control: Challenges and opportunities*. 2019. URL: `https://openreview.net/forum?id=ByexVzSAs4`.

[27] Min Hyuk Lim, Woo Hyung Lee, Byoungjun Jeon, and Sungwan Kim. “A Blood Glucose Control Framework Based on Reinforcement Learning With Safety and Interpretability: In Silico Validation”. In: *IEEE Access* 9 (2021), pp. 105756–105775. URL: `https://doi.org/10.1109/access.2021.3100007`.

[28] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[29] Lingheng Meng, Rob Gorbet, and Dana Kulic. “Memory-based Deep Reinforcement Learning for POMDPs”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sept. 2021. URL: `https://doi.org/10.1109/iros51168.2021.9636140`.

[30] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1861–1870.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[32] Phuwadol Viroonluecha, Esteban Egea-Lopez, and Jose Santa. “Evaluation of blood glucose level control in type 1 diabetic patients using deep reinforcement learning”. In: *PLOS ONE* 17.9 (Sept. 2022). Ed. by Pasquale Palumbo, e0274608. URL: `https://doi.org/10.1371/journal.pone.0274608`.

[33] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. *Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems*. 2020. URL: `https://arxiv.org/abs/2005.01643`.

[34] Michael Janner, Qiyang Li, and Sergey Levine. “Offline Reinforcement Learning as One Big Sequence Modeling Problem”. In: *Advances in Neural Information Processing Systems*. 2021, pp. 1–14.

[35] Lili Chen et al. “Decision Transformer: Reinforcement Learning via Sequence Modeling”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021, pp. 1–14. URL: `https://openreview.net/forum?id=a7APmM4B9d`.

[36] Navid Resalat, Joseph El Youssef, Nichole Tyler, Jessica Castle, and Peter G. Jacobs. “A statistical virtual patient population for the glucoregulatory system in type 1 diabetes with integrated exercise model”. In: *PLOS ONE* 14.7 (July 2019). Ed. by Pasquale Palumbo, e0217301. URL: `https://doi.org/10.1371/journal.pone.0217301`.

[37] Muhammad Asad, Usman Qamar, and Muhammad Abbas. “Blood Glucose Level Prediction of Diabetic Type 1 Patients Using Nonlinear Autoregressive Neural Networks”. In: *Journal of Healthcare Engineering* 2021 (Feb. 2021). Ed. by Saverio Maietta, pp. 1–7. URL: `https://doi.org/10.1155/2021/6611091`.

[38] Noah Salas, Brock Ferguson, and Jacob Zweig. *Reinforcement learning for personalized medication dosing*.

[39] Roberto Visentin et al. “The UVA/Padova Type 1 Diabetes Simulator Goes From Single Meal to Single Day”. In: *Journal of Diabetes Science and Technology* 12.2 (Feb. 2018), pp. 273–281. URL: `https://doi.org/10.1177/1932296818757747`.

[40] Ashenafi Zebene Woldaregay et al. “Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes”. In: *Artificial Intelligence in Medicine* 98 (July 2019), pp. 109–134. URL: `https://doi.org/10.1016/j.artmed.2019.07.007`.

[41] Jinyu Xie. *Simglucose v0. 2.1*. 2018. URL: `https://github.com/jxx123/simglucose`.

[42] Sofia Goel, Sudhansh Sharma, and RC Tripathi. “Predicting Diabetes using CNN for Various Activation Functions: A Comparative Study”. In: *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE, Dec. 2021, pp. 665–669. URL: `https://doi.org/10.1109/smart52563.2021.9676280`.

[43] Audrey Huang, Liu Leqi, Zachary Lipton, and Kamyar Azizzadenesheli. “Off-policy risk assessment in contextual bandits”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23714–23726.

[44] Greg Brockman et al. “OpenAI gym”. In: *arXiv preprint arXiv:1606.01540* (2016).

[45] Tarık Kırkgöz et al. “Efficacy of the Novel Degludec/Aspart Insulin Co-formulation in Children and Adolescents with Type 1 Diabetes: A Real-life Experience with One Year of IDegAsp Therapy in Poorly Controlled and Non-compliant Patients”. In: *Journal of Clinical Research in Pediatric Endocrinology* 14.1 (Mar. 2022), pp. 10–16. URL: `https://doi.org/10.4274/jcrpe.galenos.2021.2021.0113`.

[46] Md. Kamrul Hasan, Md. Ashraful Alam, Dola Das, Eklas Hossain, and Mahmudul Hasan. “Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers”. In: *IEEE Access* 8 (2020), pp. 76516–76531. URL: `https://doi.org/10.1109/access.2020.2989857`.

[47] Himanshu Gupta, Hirdesh Varshney, Tarun Kumar Sharma, Nikhil Pachauri, and Om Prakash Verma. “Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction”. In: *Complex & Intelligent Systems* 8.4 (May 2021), pp. 3073–3087. URL: `https://doi.org/10.1007/s40747-021-00398-7`.

[48] Elaheh Afsaneh, Amin Sharifdini, Hadi Ghazzaghi, and Mohadeseh Zarei Ghobadi. “Recent applications of machine learning and deep learning models in the prediction, diagnosis,

and management of diabetes: a comprehensive review”. In: *Diabetology & Metabolic Syndrome* 14.1 (Dec. 2022). URL: `https://doi.org/10.1186/s13098-022-00969-9`.

[49] Benjamin Eysenbach and Sergey Levine. “If MaxEnt RL is the answer, what is the question?” In: *International Conference on Learning Representations*. 2020, pp. 1–26.

[50] Hongzi Mao, Shaileshh Bojja Venkatakrishnan, Malte Schwarzkopf, and Mohammad Alizadeh. “Variance reduction for reinforcement learning in input-driven environments”. In: *arXiv preprint arXiv:1807.02264* (2018).

[51] Scott Fujimoto, Herke Hoof, and David Meger. “Addressing function approximation error in actor-critic methods”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1587–1596.

[52] Weiwei Zhao et al. “Research on the Multiagent Joint Proximal Policy Optimization Algorithm Controlling Cooperative Fixed-Wing UAV Obstacle Avoidance”. In: *Sensors* 20.16 (Aug. 2020), p. 4546. URL: `https://doi.org/10.3390/s20164546`.

[53] Sunan Cui, Huan-Hsin Tseng, Julia Pakela, Randall K. Ten Haken, and Issam El Naqa. “Introduction to machine and deep learning for medical physicists”. In: *Medical Physics* 47.5 (May 2020). URL: `https://doi.org/10.1002/mp.14140`.

[54] Clara Meister, Tim Vieira, and Ryan Cotterell. “Best-First Beam Search”. In: *Transactions of the Association for Computational Linguistics* 8 (Dec. 2020), pp. 795–809. URL: `https://doi.org/10.1162/tacl_a_00346`.

[55] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[56] Huma Naz and Sachin Ahuja. “Deep learning approach for diabetes prediction using PIMA Indian dataset”. In: *Journal of Diabetes & Metabolic Disorders* 19.1 (Apr. 2020), pp. 391–403. URL: `https://doi.org/10.1007/s40200-020-00520-5`.

[57] Kezhi Li, John Daniels, Chengyuan Liu, Pau Herrero, and Pantelis Georgiou. “Convolutional Recurrent Neural Networks for Glucose Prediction”. In: *IEEE Journal of Biomedical and Health Informatics* 24.2 (Feb. 2020), pp. 603–613. URL: `https://doi.org/10.1109/jbhi.2019.2908488`.

[58] Md Fazle Rabby, Yazhou Tu, Md Imran Hossen, Insup Lee, Anthony S. Maida, and Xiali Hei. “Stacked LSTM based deep recurrent neural network with kalman smoothing for blood glucose prediction”. In: *BMC Medical Informatics and Decision Making* 21.1 (Mar. 2021). URL: `https://doi.org/10.1186/s12911-021-01462-5`.

[59] Wenbo Wang, Meng Tong, and Min Yu. “Blood Glucose Prediction With VMD and LSTM Optimized by Improved Particle Swarm Optimization”. In: *IEEE Access* 8 (2020), pp. 217908–217916. URL: https://doi.org/10.1109/access.2020.3041355.

[60] Taiyu Zhu, Lei Kuang, Kezhi Li, Junming Zeng, Pau Herrero, and Pantelis Georgiou. “Blood Glucose Prediction in Type 1 Diabetes Using Deep Learning on the Edge”. In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2021, pp. 1–5. URL: https://doi.org/10.1109/iscas51556.2021.9401083.

[61] Xiang Lu and Ruizhuo Song. “A Hybrid Deep Learning Model for the Blood Glucose Prediction”. In: *2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE, Aug. 2022, pp. 1037–1043. URL: https://doi.org/10.1109/ddcls55054.2022.9858348.

[62] Ammar Jalalimanesh, Hamidreza Shahabi Haghighi, Abbas Ahmadi, and Madjid Soltani. “Simulation-based optimization of radiotherapy: Agent-based modeling and reinforcement learning”. In: *Mathematics and Computers in Simulation* 133 (2017), pp. 235–248.

[63] Brenden K. Petersen et al. “Deep Reinforcement Learning and Simulation as a Path Toward Precision Medicine”. In: *Journal of Computational Biology* 26.6 (June 2019), pp. 597–604. URL: https://doi.org/10.1089/cmb.2018.0168.

[64] Zihao Wang et al. “Reinforcement Learning-Based Insulin Injection Time And Dosages Optimization”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2021, pp. 1–8. URL: https://doi.org/10.1109/ijcnn52387.2021.9533957.

[65] Adnan Jafar, Anas El Fathi, and Ahmad Haidar. “Long-term use of the hybrid artificial pancreas by adjusting carbohydrate ratios and programmed basal rate: A reinforcement learning approach”. In: *Computer Methods and Programs in Biomedicine* 200 (Mar. 2021), p. 105936. URL: https://doi.org/10.1016/j.cmpb.2021.105936.

[66] Marı́a Cecilia Serafini, Nicolás Rosales, and Fabricio Garelli. “Long-Term Adaptation of Closed-Loop Glucose Regulation Via Reinforcement Learning Tools”. In: *IFAC-PapersOnLine* 55.7 (2022), pp. 649–654. URL: https://doi.org/10.1016/j.ifacol.2022.07.517.

[67] Sayyar Ahmad, Aleix Beneyto, Ivan Contreras, and Josep Vehi. “Bolus Insulin calculation without meal information. A reinforcement learning approach”. In: *Artificial Intelligence in Medicine* 134 (Dec. 2022), p. 102436. URL: https://doi.org/10.1016/j.artmed.2022.102436.

[68] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. “Basal Glucose Control in Type 1 Diabetes Using Deep Reinforcement Learning: An In Silico Validation”. In: *IEEE Journal of Biomedical and Health Informatics* 25.4 (Apr. 2021), pp. 1223–1232. URL: https://doi.org/10.1109/jbhi.2020.3014556.

[69] Taiyu Zhu, Kezhi Li, Lei Kuang, Pau Herrero, and Pantelis Georgiou. “An Insulin Bolus Advisor for Type 1 Diabetes Using Deep Reinforcement Learning”. In: *Sensors* 20.18 (Sept. 2020), p. 5058. URL: https://doi.org/10.3390/s20185058.

[70] Alan Mackey and Eoghan Furey. “Artificial Pancreas Control for Diabetes using TD3 Deep Reinforcement Learning”. In: *2022 33rd Irish Signals and Systems Conference (ISSC)*. IEEE, June 2022, pp. 1–6. URL: https://doi.org/10.1109/issc55427.2022.9826219.

[71] Mariano De Paula, Luis Omar Ávila, and Ernesto C. Martínez. “Controlling blood glucose variability under uncertainty using reinforcement learning and Gaussian processes”. In: *Applied Soft Computing* 35 (Oct. 2015), pp. 310–332. URL: https://doi.org/10.1016/j.asoc.2015.06.041.

[72] Ian Fox. “Machine Learning for Physiological Time Series: Representing and Controlling Blood Glucose for Diabetes Management”. PhD dissertation. University of Michigan, July 2020.

[73] Boris P. Kovatchev, Marc Breton, Chiara Dalla Man, and Claudio Cobelli. “In Silico-Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes”. In: *Journal of Diabetes Science and Technology* 3.1 (Jan. 2009), pp. 44–55. URL: https://doi.org/10.1177/193229680900300106.

[74] Matthew Hausknecht and Peter Stone. “Deep recurrent Q-learning for partially observable MDPs”. In: *arXiv preprint arXiv:1507.06527* (2015).

[75] US Food, Drug Administration, et al. *FDA approves automated insulin delivery and monitoring system for use in younger pediatric patients*. 2018.

[76] William Clarke and Boris Kovatchev. “Statistical Tools to Analyze Continuous Glucose Monitor Data”. In: *Diabetes Technology & Therapeutics* 11.S1 (June 2009), S–45–S–54. URL: https://doi.org/10.1089/dia.2008.0138.

[77] Tianqi Wei and Barbara Webb. “A Bio-inspired Reinforcement Learning Rule to Optimise Dynamical Neural Networks for Robot Control”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018, pp. 556–561. URL: https://doi.org/10.1109/iros.2018.8594017.

[78] Tadej Battelino et al. “Clinical Targets for Continuous Glucose Monitoring Data Interpretation: Recommendations From the International Consensus on Time in Range”. In: *Diabetes Care* 42.8 (June 2019), pp. 1593–1603. URL: https://doi.org/10.2337/dci19-0028.

[79] Jinyu Xie. *How did you obtain the parameters in vpatient_params.CSV?* Mar. 2022. URL: https://github.com/jxx123/simglucose/issues/26.

[80] Marcello Pompa, Simona Panunzi, Alessandro Borri, and Andrea De Gaetano. “A comparison among three maximal mathematical models of the glucose-insulin system”. In: *PLOS ONE* 16.9 (Sept. 2021). Ed. by Lidia Castagneto-Gissey, e0257789. URL: https://doi.org/10.1371/journal.pone.0257789.

[81] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. *Stable Baselines3*. 2019. URL: https://github.com/DLR-RM/stable-baselines3.

[82] Sergio Guadarrama et al. *TF-Agents: A library for reinforcement learning in tensorflow*. 2018. URL: https://www.tensorflow.org/agents.

[83] Markus Holzleitner, Lukas Gruber, Jose Arjona-Medina, Johannes Brandstetter, and Sepp Hochreiter. “Convergence Proof for Actor-Critic Methods Applied to PPO and RUDDER”. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII*. Springer Berlin Heidelberg, 2021, pp. 105–130. URL: https://doi.org/10.1007/978-3-662-63519-3_5.

[84] Wenshuai Zhao, Jorge Pena Queralta, and Tomi Westerlund. “Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey”. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Dec. 2020, pp. 737–744. URL: https://doi.org/10.1109/ssci47803.2020.9308468.

[85] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 2623–2631.

[86] Richard Bergenstal. “Understanding Continuous Glucose Monitoring Data”. In: *ADA Clinical Compendia* (Aug. 2018), pp. 20–23. URL: https://doi.org/10.2337/db20181-20.

[87] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. “Algorithms for hyper-parameter optimization”. In: *Advances in neural information processing systems* 24 (2011).

[88] Nikolaus Hansen. “The CMA evolution strategy: a comparing review”. In: *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms* (2006), pp. 75–102.

[89] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*. Vol. 1. Springer, 2006.

[90] Guangzhi Rong et al. “Comparison of Tree-Structured Parzen Estimator Optimization in Three Typical Neural Network Models for Landslide Susceptibility Assessment”. In: *Remote Sensing* 13.22 (Nov. 2021), p. 4694. URL: `https://doi.org/10.3390/rs13224694`.

[91] Charlotte K. Boughton and Roman Hovorka. “New closed-loop insulin systems”. In: *Diabetologia* 64.5 (Feb. 2021), pp. 1007–1015. URL: `https://doi.org/10.1007/s00125-021-05391-w`.

[92] Francine R. Kaufman and Ohad Cohen. *A Guide to Personal Continuous Glucose Monitoring*. English. Medtronic. 36 pp.

[93] Emanuele Bosi et al. “Efficacy and safety of suspend-before-low insulin pump technology in hypoglycaemia-prone adults with type 1 diabetes (SMILE): an open-label randomised controlled trial”. In: *The Lancet Diabetes & Endocrinology* 7.6 (June 2019), pp. 462–472. URL: `https://doi.org/10.1016/s2213-8587(19)30150-0`.

[94] Satish K. Garg et al. “Glucose Outcomes with the In-Home Use of a Hybrid Closed-Loop Insulin Delivery System in Adolescents and Adults with Type 1 Diabetes”. In: *Diabetes Technology & Therapeutics* 19.3 (Mar. 2017), pp. 155–163. URL: `https://doi.org/10.1089/dia.2016.0421`.

[95] Gregory P. Forlenza et al. “Safety Evaluation of the MiniMed 670G System in Children 7–13 Years of Age with Type 1 Diabetes”. In: *Diabetes Technology & Therapeutics* 21.1 (Jan. 2019), pp. 11–19. URL: `https://doi.org/10.1089/dia.2018.0264`.

[96] Sachin G Konan, Esmaeil Seraj, and Matthew Gombolay. “Contrastive Decision Transformers”. In: *6th Annual Conference on Robot Learning*. June 2022, pp. 1–11.

[97] Mengdi Xu et al. “Prompting decision transformer for few-shot policy generalization”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 24631–24645.

[98] Ashkan Kazemi et al. “Adaptable Claim Rewriting with Offline Reinforcement Learning for Effective Misinformation Discovery”. In: *arXiv preprint arXiv:2210.07467* (2022).

[99] T. M. Peters and A. Haidar. “Dual-hormone artificial pancreas: benefits and limitations compared with single-hormone systems”. In: *Diabetic Medicine* 35.4 (Feb. 2018), pp. 450–459. URL: `https://doi.org/10.1111/dme.13581`.

[100] Ahmad Haidar et al. “A Novel Dual-Hormone Insulin-and-Pramlintide Artificial Pancreas for Type 1 Diabetes: A Randomized Controlled Crossover Trial”. In: *Diabetes Care* 43.3 (Jan. 2020), pp. 597–606. URL: `https://doi.org/10.2337/dc19-1922`.

# A

# Supplementary material

The source codes, evaluation and training results, and trained agent policies presented in Chapter 3, can be accessed through our public repository located at https://github.com/girtel/AIML4Diabetes and https://osf.io/gj783. These resources are made available to support reproducibility and further research in the field.

Additionally, the datasets used in the study described in Chapter 5 including the baseline data and training datasets, are accessible in CSV format on the open science framework repository at https://osf.io/zurvk. These datasets have been shared to support reproducibility and encourage further investigation in this area.

# B

# List of publications

- Phuwadol Viroonluecha, Esteban Egea-Lopez, and Jose Santa, “Evaluation of blood glucose level control in type 1 diabetic patients using deep reinforcement learning,” *PLOS ONE*, vol. 17, no. 9. Public Library of Science (PLoS), p. e0274608, Sep. 13, 2022. URL: http://dx.doi.org/10.1371/journal.pone.0274608