A Generalised Closed-Loop Reinforcement Learning Artificial Pancreas

Thomas Milton, Robert Lieck

***Abstract*— Type 1 diabetes is a chronic metabolic disease that prevents the production of insulin to regulate blood glucose. As such, diabetic patients are required to manually inject insulin which is subject to human error. Therefore, we use reinforcement learning to produce a model to automatically inject insulin to regulate blood glucose with a higher degree of control. Diabetic patients are also required to give meal announcements to current state-of-the-art control systems to mitigate large blood glucose spikes due to carbohydrate ingestion. Meal announcements are also still subject to human error. Therefore, we produce a closed-loop model that does not require human interaction. Prior work has focused on training patient-specific models, as this is believed to be necessary due to the high inter-patient variability. However, training such models in-vivo is challenging. Therefore, we produce a generalised model that can perform effectively on unseen patients without additional training. In conclusion, our closed-loop generalised reinforcement learning model controls blood glucose to a high degree of accuracy whilst not requiring manual input. Our unique contribution is generalisation which should increase the ease and accessibility of switching to a closed-loop artificial pancreas system in a real-world setting. Specifically, we use the deep reinforcement learning algorithm Soft Actor-Critic with various enhancements and the FDA-approved UVA/Padova simulator with corrected meal scenarios. We compare our results with a non-generalised baseline PID controller (lower bound) and a non-closed-loop ground truth meal oracle (upper bound). To ensure reproducibility of results and to provide a test bench for future research on this topic, we provide clean and well-documented code, including our models and an improved standardised environment with various meal scenarios.**

***Index Terms*— Artificial intelligence, artificial pancreas, diabetes, reinforcement learning**

I. Introduction

Type 1 Diabetes (T1D) is a chronic metabolic disease that prevents the production of insulin to regulate Blood Glucose (BG). We give relevant background information on T1D and justify its management as an important area of research in section I-A. T1D is traditionally managed with manual BG readings and manual insulin injections which is particularly onerous and subject to human error as described in section I-B. As such, we introduce a more sophisticated approach to T1D management in section I-C, called an Artificial Pancreas (AP), to automatically measure BG and automatically give insulin doses. The controller component of an AP calculates the appropriate insulin dosage and is traditionally a classical control algorithm such as Proportional Integral Derivative (PID). We introduce these classical APs in section I-C and further discuss them in section II-A. We come to the conclusion that Reinforcement Learning (RL) – introduced in section I-D with a more formal definition in III-A – is a better alternative for T1D management.

The main challenge for T1D management is meal ingestion since the Carbohydrate (CHO) content of meals causes a large spike in BG. Classical APs require a meal announcement from the user to allow the controller to preempt the BG spike. This means that classical APs are not Closed-Loop (CL) – they require human interaction which is burdensome and dangerous. We produce a CL RL AP that is able to safely control BG without human intervention. This is discussed further in I-C.

https://github.com/Tom-Milton/Dissertation-RLAP

Prior works that produce CL RL APs train patient-specific models due to the large inter-patient variability, however, training these models both in-silico and in-vivo is challenging in a real-world setting. We show that a generalised RL AP can perform effectively on unseen patients without prior training. This is also discussed further in I-C.

In conclusion, we produce a generalised CL RL AP that can control BG to a high degree of accuracy without manual intervention and can be easily switched to in a real-world setting. We conclude that our generalisation approach is a unique contribution to the field in section II-B by evaluating the current literature.

Specifically, we use the RL algorithm Soft Actor Critic (SAC) which we formally define and justify in section III-B. We provide an in depth review of prior works that use the same algorithm as us in section II-C. Specifically, we use the python port simglucose of the FDA-approved UVA/Padova T1D simulator as our environment. We discuss the development of T1D simulators and justify the use of simglucose in section III-C. Note that we overhaul the meal scenario generation of this simulator as described in section III-D.

In the remainder of our methodology, we give detailed descriptions and justifications of our implementational design choices. This includes our state space III-E, action space III-F, and reward function III-G. We conclude our methodology by defining our training parameters in section III-H and introducing our final models in section III-I. These models include our fully generalised CL model, our non-CL ground truth meal oracle (upper bound), and our non-generalised PID baseline (lower bound).

In our results section IV, we give results for our three models mentioned above and compare with the state-of-the-art. This includes average metrics for Time in Range (TIR) and Risk Index (RI) which can be found in Table I. More detailed (patient-specific) graphs can be found in Appendix I alongside progress plots of our parameters during training.

In our evaluation sections V-A to V-F we discuss the performance of our three models mentioned above and compare against the state-of-the-art. In section V-G, we also argue a case for a level of generalisation between that of full generalisation and patient-specific generalisation – namely class-specific generalisation. Finally, we conclude with a discussion of future work in section V-H

A. Diabetes

Diabetes Mellitus (DM) is a chronic metabolic disease that affects how the body responds to Blood Glucose (BG), characterised by prolonged high BG levels. DM is categorised into two chronic conditions: Type 1 Diabetes (T1D) and Type 2 Diabetes (T2D); and two reversible conditions: gestational diabetes and prediabetes. T1D is a genetic condition that causes the body to attack the cells in the patient's pancreas whereas T2D develops often as a result of lifestyle choices such as poor diet and physical inactivity.

Insulin is a hormone produced by the pancreas that lowers BG levels by allowing BG to enter the body's cells, primarily for cellular respiration. T1D prevents the production of insulin whereas T2D prevents the body from producing enough insulin or prevents the cells of the body from responding properly to the insulin that is produced. Lastly, gestational diabetes develops during pregnancy and

usually disappears after giving birth whereas prediabetes is classified by having higher than normal BG levels, but not high enough to be diagnosed as T2D.

Whilst more than 95% of global diabetes cases are T2D [1], it is both curable and more easily managed through medication, exercise, and diet. The latter two forms of DM are not necessarily permanent, although some cases of both can still develop into T2D. In contrast, T1D is both chronic and incurable so that management is the only viable solution. We therefore focus our study on this form of DM. Moreover, simulators for in-silico experimentation are primarily available for T1D only, as discussed in Section III-C.

In 2014, there were 422 million global cases of adult DM with a global annual cost of $825 billion towards treating and managing DM [2]. Prolonged DM can result in damage to the heart, blood vessels, eyes, kidneys, and nerves, resulting in heart attacks, strokes, blindness, kidney failure, and lower limb amputation. In 2019, DM was the ninth leading cause of global deaths with an estimated 1.5 million deaths [1]. Therefore, for these reasons, DM management is an important area of current research.

B. Diabetes Management

Since patients with T1D do not produce insulin, BG can rise uncontrollably resulting in hyperglycemia if it becomes too high. Insulin is then injected to lower BG, however, too much insulin can result in hypoglycemia if the BG becomes too low. Whilst hypoglycemia is more dangerous [3], prolonged hyperglycemia can also result in diabetic ketoacidosis as the body runs out of insulin and starts producing ketones. Therefore, the optimal control of BG close to the non-DM average and without sharp spikes in BG is essential. Diabetic patients aim to keep their BG between 70 mg/dL and 180 mg/dL for as long as possible – this is referred to as the Time in Range (TIR) and is an important metric to consider for T1D control.

Since insulin was first discovered in 1921, it has been used to treat T1D with a basal-bolus insulin regime that aims to mimic the pancreas' normal insulin production – long-acting basal insulin controls fasting BG whilst rapid-acting bolus insulin controls sharp BG spikes after meals. Intensive Insulin Therapy (IIT) is the most common form of T1D management using either Multiple Daily Injections (MDI) or Continuous Subcutaneous Insulin Infusion (CSII), combined with BG monitoring using either Self Monitoring of Blood Glucose (SMBG) or a Continuous Glucose Monitor (CGM) [4].

In practice, MDI requires manual injections of long- and rapid-acting insulin, whilst CSII reduces burden on the user by using an insulin pump to continuously deliver basal insulin as the user delivers prandial and corrective boluses. In practice, SMBG requires manually retrieving BG levels via finger-tip testing, whilst CGMs reduce burden on the user by using a sensor to automatically provide BG values up to every 5 minutes.

However, IIT requires the user to have an in-depth understanding of the highly complex glucoregulatory system. As such, many user errors result in hypo- and hyperglycemia and an increased chance of their associated risks. It has been shown that the more automative approach of T1D management, by using CSII with a CGM, has a favourable effect on Glycated Haemoglobin (HbA1c) levels over MDI with SMBG [5]. HbA1c reflects the average BG levels for the last two to three months and is an important metric to consider for T1D control. Therefore, further increasing T1D control autonomy is likely to increase performance.

C. Artificial Pancreas

An Artificial Pancreas (AP) is a more sophisticated approach to T1D management than IIT and involves three components: a CGM to measure interstitial BG levels, an insulin pump to deliver insulin, and a control algorithm that converts inputs from the CGM to insulin delivery instruction for the insulin pump [6]. The AP framework is shown in fig. 1. This system is more sophisticated than those discussed above as it allows the CGM to directly interact with the CSII pump via the controller. As such, the controller is arguably the most important component of the AP so is an important area of current research.

It is also worth briefly noting that there is a more relaxed form of an AP, known as a Decision Support System (DSS), that gives treatment recommendations rather than direct control for less invasive and safer DM management [7]. This could also be an interesting alternative area of research.

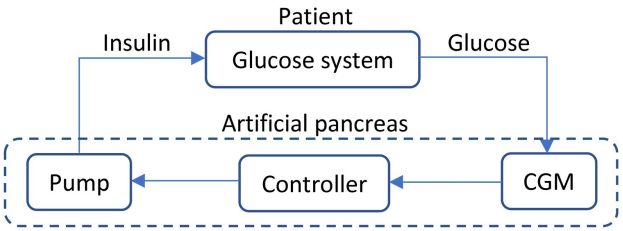


Fig. 1: Artificial Pancreas Framework [8]

The three most common control algorithms currently used for the controller are Proportional Integral Derivative (PID) Control, Model Predictive Control (MPC), and Fuzzy Logic Control (FLC) [4], however these all suffer from the inability to quickly adapt to postprandial BG spikes [9]. Consequently, the currently available AP solutions are merely hybrid Closed-Loop (CL) systems as they still require user input regarding meals to help control large fluctuations in postprandial BG. A further discussion of these classical control algorithms is given in section II-A.

Meal announcement is highly subject to human error – a previous study found that the Carbohydrate (CHO) content estimate of meals by T1D patients was underestimated by 20.9±9.7% of the total actual CHO content [10]. Estimation errors cause the controller to calculate inappropriate insulin dosages which may result in dangerous fluctuations in BG. Therefore, we aim to implement a fully CL AP as it reduces the burden on the user by removing the need for meal announcements and reduces the risks associated with human error in CHO estimation. In fact, CL APs have been shown to perform better than all other approaches [11].

It is believed that patient-specific APs are necessary due to the high inter-patient variability [12, 13]. Training a patient-specific model can be safely performed in-silico but these requires a detailed metabolic description of the T1D patient which may be subject to some inaccuracies and is likely expensive. Alternatively, training in-vivo may be dangerous since the model directly interacts with the patient's metabolic system. Therefore, we implement a fully generalised AP such that it can perform well on unseen patients without prior training. This will make switching to an AP system safer and more accessible.

In conclusion, we produce a generalised CL RL AP that can control BG to a high degree of accuracy without manual intervention and can be easily switched to in a real-world setting.

D. Reinforcement Learning

Reinforcement Learning (RL) is a sub-field of Artificial Intelligence (AI) that automates the decision making process through an agent takes actions in a given environment, based on its current state, in order to maximise its cumulative reward. The RL framework is shown in fig. 2. The mapping of a certain state to a certain action is

known as the policy, and the aim of RL is to find an optimal policy that maximises the cumulative reward over time. The optimal policy learnt by this system can then be used to optimally control the user's BG levels. We give a more formal definition of RL in section III-A

For the T1D RL control task, this system is given as follows:

- the **agent** represents the controller
- the **action** represents the insulin dosage as provided by Continuous Subcutaneous Insulin Infusion (CSII)
- the **environment** represents the users glucoregulatory system, in this case a simulator
- the **state** represents the BG level as measured by the Continuous Glucose Monitor (CGM)
- the **reward** represents some notion of the discrepancy between the ideal and actual BG levels

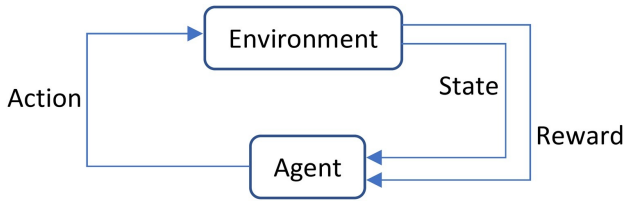


Fig. 2: Reinforcement Learning Framework [8]

RL is suited for situations involving sequences of decisions along a timeline where actions depend on the current state whilst the actions' effects may manifest at later points in time than when induced (time delay). These conditions are all true in the AP system so RL would provide an apt solution to T1D control [14].

There is also a large inter- and intraindividual variability in the pharmacokinetics and pharmacodynamics between diabetic patients. Since the RL agent continuously adapts to its environment, it can produce a personalised control algorithm that more traditional control algorithms such as PID, MPC, and FLC are less capable of [14]. Although for our generalised task, this feature is not exploited. Nevertheless, our fully generalised model could still be used as an entry point for training a patient-specific fine-tuned control algorithm via Transfer Learning (TL).

Other subdomains of AI such as Machine Learning (ML) have also been applied to T1D control. However, supervised ML control algorithms require large amounts of labeled training data, whilst model-based control algorithms are required to model the glucoregulatory system which is extremely complex. Since RL is unsupervised and can be model-free, we are able to bypass these laborious and complex processes that would otherwise limit the capabilities of our algorithm.

II. Related Work

We discuss classical control algorithms used for T1D management in section II-A – including Proportional Integral Derivative (PID) Control, Model Predictive Control (MPC), and Fuzzy Logic Control (FLC) – and conclude that RL is a superior alternative. We assess all relevant RL APs in section II-B, with the aid of a recent literature review, to conclude that our generalised approach is unique. Given the breadth and depth of the previously proposed RL APs, we discuss only some of the general implementational details such as algorithm selection and state/action space formulation in section II-B. Of these papers, we provide a further investigation into those that use the same RL algorithm as us (Soft Actor Critic) in section II-C, concluding that their results are not comparable with ours. Finally, in section II-D, we give an overview of the future of T1D via Multiple-Input Artificial Pancreas System (MAPS).

A. Classical Artificial Pancreases

We first discuss non-AI-based APs used for T1D control – some of which have even been approved for commercial use by the Food and Drug Administration (FDA) [15]. We do not evaluate individual approaches as they are not within our focus on RL APs. However, we do give a brief introduction to the most common general AP control algorithms – namely PID, MPC, and FLC – as this provides a good understanding of the current T1D control approaches and the benefits of a RL approach. Moreover, these approaches are also often used as baselines to evaluate an AP's performance – in particular, we use a PID baseline.

Proportional Integral Derivative (PID) controllers adjust the insulin delivery rate according to three factors: proportional – the difference in the current BG from the ideal level; integral – the area under the curve between the observed and target BG; derivative – the rate of change of the BG. An intuitive view of this algorithm is that the proportional, integral, and derivative terms reflect the present, past, and future BG trends respectively. These three components are balanced through loop tuning where the tuning constants are specific to the application.

Model Predictive Control (MPC) first defines a mathematical model of the glucoregulatory system which can include any errors and time delays associated with an AP. It is combined with constraints on the dependent and independent variables to ensure feasibility and safety. This model is then used to calculate insulin infusion rates by minimising the difference between the actual BG level and the model's predicted BG level.

Fuzzy Logic Control (FLC) first uses a fuzzier to transform the input BG variable into linguistic variables. These are then operated on by the controller with 'if-then' logic rules where the 'if' part is termed the antecedent and the 'then' part is termed the consequent. A defuzzier then converts this fuzzy output back into an appropriate insulin infusion rate.

Both PID and FLC are reactive algorithms as they are only able to respond to current BG levels; in contrast, MPC is a proactive algorithm as it can anticipate BG trends from its predefined model [4]. This gives MPC the ability to accommodate for insulin absorption delays allowing it to somewhat mitigate meal intake BG spikes with preprandial boluses. However, MPC requires a sophisticated model of the glucoregulatory system which is highly complex. A RL approach would provide better proactive capabilities than MPC without the necessity for any model.

FLC is well suited for the medical field since its linguistic-based rules can be easily derived from human expert knowledge. However, this could equally be a disadvantage as it is entirely dependent on humans – specifically human error and the constraints of human knowledge. Interestingly, fuzzy logic can be applied on top of other control algorithms, including RL [16].

B. Reinforcement Learning Artificial Pancreases

A thorough review of the state-of-the-art RL approaches for DM control, published between 1990 and 2019, was performed by Tejedor et al. [8]. We use this as a basis for our research whilst considering the remaining RL approaches published since – of which there are numerous. We investigate these papers for references to generalisation. Directly from the review paper by Tejedor et al. we see that the majority of these papers do not involve a generalised approach as they are limited to too few patients. From a closer review of the remaining papers we see that they choose not to investigate a generalised model since it is believed that the inter-patient variability is too large for any one control algorithm to provide sufficient performance across multiple patients [12, 13].

The closest to a generalised approach is provided by Zhu et. al. [17, 18] who create a generalised model by training on the average virtual patient (this is given by the average of all the metabolic parameters of the UVA/Padova virtual patient cohort). Firstly, this is only used as an initial model for patient-specific fine-tuning via Transfer Learning (TL) so is not the main focus of their paper. Secondly, this generalised model is tested on the entire cohort of virtual patients which is not a fair assessment since these are not strictly unseen patients as they have been partially included in the average virtual patient used for training. Therefore, we may conclude that, to the best of our knowledge, our generalised RL approach is unique.

The specific approaches of each RL AP are too diverse and complex to discuss in detail, however we may discuss some of the more general implementation details. The most common RL AP approaches are model-free (79.3% identified by Tejedor et al.) with the most common algorithm being an Actor-Critic (AC) method (37% identified by Tejedor et al.). We follow suit and choose to use Soft Actor Critic (SAC), a model-free AC algorithm, as it has already shown high performance in state-of-the-art approaches for T1D control [19, 20, 21, 22]. We formally define SAC and give further justification for its use in section III-B.

The nature of the state space and action spaces is fairly unanimous with 73% of the state spaces and 67% of the action spaces in the reviewed papers being continuous – note that SAC can only be used with continuous action spaces. What is more important is the defining parameters of the state and action spaces.

For the state space, 43% of implementations include only BG whilst 30% use BG and insulin levels. We also choose to include insulin (among other things) into the state space of our model as it reflects Insulin on Board (IOB) – the amount of insulin already in the patient's body. We also find that only 6.67% include Carbohydrate (CHO) in the state space. This is expected since adding CHO would make the controller no longer Closed-Loop (CL) which would bypass the main benefit of using RL since there are already many reliable non-AI-based hybrid-CL alternatives commercially available. We add CHO to the state space only for our upper bound oracle model. We fully define and justify our state space choices in section III-E

For the action space, 93% of the parameters are solely comprised of the insulin dose – we follow suit. However, it would be interesting to investigate external actions such as exercise or ingesting food to counteract high or low BG respectively – although this would make the AP no longer CL. We fully define and justify our action space choices in section III-F.

C. Soft Actor-Critic Artificial Pancreases

We also check all identified papers for mention of our algorithm Soft Actor Critic (SAC) as they may be useful for comparison of methodologies and results. We discover four papers [19, 20, 21, 22] that we shall discuss in turn – the first of which also has publicly available code [23]. As a reminder, none of these papers produce a generalised model, which is the focus of our research. It is important to note also that there is no direct comparability between our results and the results presented by these papers since all approaches use a slightly altered version of the simglucose environment as discussed in section III-D.

Fox et al. [19] produce a Soft Actor Critic (SAC) Closed-Loop (CL) Reinforcement Learning (RL) Artificial Pancreas (AP). We use their approach as guidance for a lot of implementational choices since they provide their code. The following two papers also utilise a lot of the implementational details provided by this approach. As the only paper of the above cited papers that provides relevant numerical results for their model, we may compare our results with Fox et al. who achieve an average of 72.7% time in euglycemia, 0.7% time in hypoglycemia, 26.2% time in hyperglycemia, and a 6.5 Magni Risk Index across all virtual patients. Further comparison of implementation details is given throughout our paper with a comparison of results in section V-B.

Felice et al. [20] and Viroonluecha et al. [21] both also provide a SAC CL RL AP as well as an alternative Deep Deterministic Policy Gradient (DDPG) and Proximal Policy Optimisation (PPO) approach respectively. Felice et al. conclude that their SAC approach performs best but do not provide any performance results – this supports our choice of RL algorithm. Viroonluecha et al. conclude that only their PPO approach could learn effective policies so only provide results for this model – whilst also not providing exact numerical values. As such we do not compare the performance of our model with either of these two approaches.

Lim et al. [22] alternatively focus on a safe and interpretable CL RL AP by combining SAC with a guiding PID policy during the early stages of training and a modulating adaptive safe actor. They also use random forest regression and dual attention network prediction models to forecast future glucose levels for safety measures whilst their attention and attribution scores can be used for interpretability. Since this is a completely different task, we do not compare our model with their results.

D. Multiple Input Artificial Pancreases

An important area of future research is Multiple-Input Artificial Pancreas System (MAPS), where additional inputs such as ketones, lactate, adrenaline, heart rate, skin resistance, and/or galvanic skin response are added to an AP to try and account for the effects of exercise, sleep, stress, and/or illness on the glucoregulatory system [24]. These parameters could simply be added to the state space, however, current T1D simulators do not provide this additional information.

Practically, this data can be easily gathered through either non-invasive methods such as wearable devices, or invasive methods such as sensors – these invasive methods could even be incorporated into the sensors currently used for Continuous Glucose Monitors (CGMs) and Continuous Subcutaneous Insulin Infusion (CSII). This data can then be used to augment the controller in three ways: switching controller modes, direct input to the controller, or the development of specific AP modules [24].

Dual hormone APs are similar to MAPS where hormones such as glucagon or amylin are added to the action space to better reflect the biological pancreas. There currently exists dual hormone APs even within the RL AP subdomain [17], however, we do not include additional hormones in our model given that only one form of insulin is provided by the simulator and to ensure our research is focused solely on a generalised model.

III. Methodology

We begin by formally introducing Reinforcement Learning (RL) and Soft Actor Critic (SAC) in section III-A and section III-B respectively. This includes our enhancements to SAC which comprise of automatic entropy tuning, a learning rate scheduler, a Prioritised Experience Replay (PER) buffer, and Emphasising Recent Experiences (ERE). In section III-C, we give a brief description of the development of Diabetes Mellitus (DM) simulators and a justification of our choice in using the python port simglucose of the FDA-approved UVA/Padova T1D simulator. We make a few small adjustments to the simulator to fix minor bugs as well as providing an overhaul of the meal scenarios as discussed in section III-D – the documentation of our changes can be found in our provided code. We

then give detailed justification regarding our implementational design choices with respect to the state space, action space, reward function, and training in sections III-E to III-H. We conclude this section by introducing our final models for analysis in section III-I.

A. Reinforcement Learning Formal Definition

RL aims to learn a reward-optimising policy for an agent in a given environment. The environment is framed as a Markov Decision Process (MDP) consisting of the 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the action space, $\mathcal{P}(s_{t+1}|s_t, a_t)$ is the transition function, $\mathcal{R}(s_t, a_t)$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor.

For the T1D control setting, we frame the problem as a Partially-Observable Markov Decision Process (POMDP) as we do not have access to the true environment states given the CGM sensor noise and unobserved meals. This consists of the 7-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$ where $\mathcal{S}$ becomes the set of true states and Ω is the set of observed noisy states with respect to the observation function $\mathcal{O}(o_t|s_{t+1}, a_t)$.

At each discrete timestep t, with the environment in state $s_t \in \mathcal{S}$, the agent selects an action $a_t \in \mathcal{A}$ with respect to its policy $\pi(a_t|o_t)$, causing the environment to transition to state s_{t+1} with respect to the transition function. The agent simultaneously receives an observation $o_t \in \Omega$ and a reward r_t from the observation and reward functions respectively.

To learn the optimal policy, the agent aims to maximise the expected infinite-horizon discounted return $J(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)]$. When $\gamma \to 0$ the agent prioritises immediate rewards and when $\gamma \to 1$ the agent prioritises long term future rewards. From hereon in, we formulate everything for a MDP instead of a POMDP as this is how SAC is formulated and the difference is trivial in the theoretical case.

B. Soft Actor-Critic Formal Definition

Soft Actor Critic (SAC) [25] is a state-of-the-art off-policy model-free Actor-Critic (AC) deep Reinforcement Learning (RL) algorithm which optimises a stochastic policy by maximising a trade-off between the expected reward and the expected entropy. Entropy is a measure of randomness within the policy that has a strong connection with the exploration-exploitation dilemma – increasing the entropy results in more exploration which can prevent the agent converging on a local optimum. Maximum entropy policies have also been shown to perform well in POMDPs [26] like the one present in our T1D control framework.

1) Overview: SAC augments the infinite-horizon discounted return to include the expected entropy. Here the temperature parameter $\alpha > 0$ is the trade-off coefficient between the entropy of the policy $\mathcal{H}$ and the reward $\mathcal{R}$.

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (\mathcal{R}(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)))\right] \tag{1}$$

SAC uses three networks: a soft V-network $V_\psi(s_t)$, a soft Q-network $Q_\theta(s_t, a_t)$, and a policy network $\pi_\phi(a_t|s_t)$. The parameters of these networks are ψ, θ, and ϕ respectively which can be adjusted to alter the behaviour of the agent. We now summarise the mathematical formulation of these three networks.

2) State-Value Function: The state-value function is used in RL to calculate the expected return when starting from state s and always acting according to policy π. The V-network V_ψ estimates these state-values and can be trained by minimising the expected squared residual error

$$J_V(\psi) = \mathbb{E}\left[\frac{1}{2}(V_\psi(s_t) - \mathbb{E}[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t)])^2\right]$$

where the gradient can be approximated as (2)

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(s_t)(V_\psi(s_t) - Q_\theta(s_t, a_t) + \log \pi_\phi(a_t|s_t)).$$

3) Action-Value Function: The action-value function (critic) is used in RL to give the expected return when starting from state s, taking some action a, and thereafter acting according to policy π. The Q-network Q_θ estimates these action-values and can be trained by minimising the expected mean squared Bellman error

$$J_Q(\theta) = \mathbb{E}\left[\frac{1}{2}(Q_\theta(s_t, a_t) - R(s_t, a_t) - \gamma \mathbb{E}[V_{\bar{\psi}}(s_{t+1})])^2\right]$$

where the gradient can be approximated as (3)

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(a_t, s_t)(Q_\theta(s_t, a_t) - R(s_t, a_t) - \gamma V_{\bar{\psi}}(s_{t+1})).$$

The critic can often overestimate the action-value causing the policy to break by exploiting this error – this is known as overestimation bias. To prevent it, SAC learns two critics $(Q_{\theta_1}, Q_{\theta_2})$ and updates both via $\hat{\nabla}_\theta J_Q(\theta)$ using whichever target action-value is smaller – this is known as clipped double-Q learning. If $Q_{\theta_2} > Q_{\theta_1}$ then no bias has been induced; if $Q_{\theta_2} < Q_{\theta_1}$ then overestimation bias has occurred and the action-value is reduced to prevent it.

4) Policy Network: Finally, the policy (actor) π_ϕ is trained by minimising the expected KL-divergence. A reparametrisation trick using the neural network transformation $a_t = f_\phi(\epsilon_t; s_t)$ is used to ensure sampling from the policy is differentiable to allow for backpropogation. This gives the objective

$$J_\pi(\phi) = \mathbb{E}[\log \pi_\phi(f_\phi(\epsilon_t; s_t)|s_t) - Q_\theta(s_t, f_\phi(\epsilon_t; s_t))]$$

where the gradient can be approximated as

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(a_t|s_t) + (\nabla_{a_t} \log \pi_\phi(a_t|s_t) - \nabla_{a_t} Q(s_t, a_t)) \nabla_\phi f_\phi(\epsilon_t; s_t). \tag{4}$$

5) Enhancements: We use a more recent variation of SAC [27] which makes the state-value function redundant and provides automatic tuning of the temperature parameter α with the objective function $J(\alpha) = \mathbb{E}[-\alpha \log \pi_t(a_t|s_t) - \alpha\bar{\mathcal{H}}]$. Note, the objective functions for Q and π also need to be updated with respect to the temperature parameter α.

We also introduce a learning rate scheduler to prevent the agent jumping out of any local optima during Transfer Learning (TL). This is set to linearly increase the learning rate for the actor and critic to 3e-4 over the first 10,000 timesteps.

We base our models on code [28] which contains three models: a standard SAC model, a model with a Prioritised Experience Replay (PER) buffer, and a model with PER whilst Emphasising Recent Experiences (ERE). We find that the best model is SAC with PER and ERE which supports the findings in the original ERE paper [29]. As such, we use SAC PER ERE for our entire paper.

Prioritised Experience Replay (PER) replaces the standard uniform sampling in a classical replay buffer with prioritised sampling with respect to the transitions Temporal Difference (TD) error. Intuitively, this means that transitions with the most error are more likely to be more information rich so are more likely to be sampled. This increases the sample efficiency during training, but also causes the algorithm to find a better optimal control policy.

Emphasising Recent Experiences (ERE) also affects the sampling from the replay buffer by encouraging recently observed data to be sampled more frequently without forgetting older transitions. This also increases the sample efficiency and causes the model to find a better optimal policy since more recent transitions during training are more information rich.

C. Environment – Simulators

The FDA-approved UVA/Padova simulator S2008 [30] was one of the first simulators available for modelling glucose-insulin metabolism, comprised of 30 virtual patients (10 adults, adolescents, and children). It also models important inaccuracies in the AP system from CGM and insulin pump errors such as time lag and calibration bias.

However, S2008 suffered from hypoglycemia underestimation so was succeeded by S2013 [31] which suffered from hypoglycemia overestimation so was further improved by S2017 [32]. Neither S2013 nor S2017 have been developed for RL so we use the 2018 open-source python implementation of S2008 named simglucose for our experimentation [33].

Therefore, we are subject to the errors associated with S2008 and are unable to train our model on the more sophisticated simulators S2013 or S2017. These include general improvements and new features such as glucagon, intradermal insulin, inhaled insulin, and dawn phenomenon models. The dawn phenomenon model is particularly important for DM management as it reflects abnormally high BG between 2 a.m. and 8 a.m. when the patient is often asleep.

The UVA/Padova simulators are not the only T1D simulators available. There also exists simulators for Multiple-Input Artificial Pancreas System (MAPS) [34, 35] which allow for the future development of more sophisticated APs to better control T1D. However, these simulators are yet to be formatted for the RL framework. There also exists a complementary Padova simulator for T2D [36] which allows for the development of DM management research outside of the T1D subdomain.

We make a few small adjustments to the simulator to fix minor bugs as well as providing an overhaul of the meal scenarios as discussed in section III-D – the documentation of our changes can be found in our provided code.

D. Meal Scenarios

The main problem with the simglucose simulator is that the default meal scenario provided does not scale meal sizes with respect to the virtual patient. This is particularly important for the more sensitive child cohort of virtual patients as, given the current meal schedule, they eat the same amount as adults. This may explain why some prior works that use the simglucose simulator do not choose to train or evaluate their models on the child cohort of virtual patients [17, 18, 37].

The meal schedule is critical since it is the only factor that makes this research area challenging – on a no-meal scenario it is trivial to achieve near perfect performance. As such, a meal scenario that gives more daily carbs will always produce a model that performs worse than a meal scenario that provides less daily carbs.

As such, we provide correct standardised meal scenarios by replacing the default meal scenario with two alternatives provided by Fox et al. [19] – named random scaled and Harris Benedict. Random scaled simply scales the meal amounts with respect to the patients body weight. The alternative scenario uses the Harris Benedict equation to calculate the patients Basal Metabolic Rate (BMR) from their age, weight, and height [38]. The virtual patients' genders are unknown so the Harris Benedict equation for men is used as this will only ever overestimate BMR. From the patient's BMR, the patient's daily carbohydrate intake can be calculated by accounting for 40% of calories coming from carbohydrate (multiply by 0.4), exercise (multiply by 1.2), and the number of calories per carbohydrate (divide by 4).

The expected Carbohydrate (CHO) calculation is given by:

$$\begin{aligned}\text{BMR} = 66.473 &+ (13.7516 \cdot \text{weight in kg}) \\ &+ (5.0033 \cdot \text{height in cm}) - (6.755 \cdot \text{age in years})\end{aligned} \tag{5}$$

$$\text{ExpectedCarbs} = (\text{BMR} \cdot 0.45)/4 \tag{6}$$

There are 5 possible meals – breakfast, snack, lunch, snack, dinner. The timing and amount of these meals are both sampled from a normal distribution with a specified mean and standard deviation to account for real word variability in a patient's diet. For our analysis, we use the Harris Benedict meal scenario as this is simply an improvement on the random scaled meal scenario as it considers the patient's age and height as well as their weight.

E. State Space

Our state $\mathbf{s_t}$ at timestep t contains the last 4 hours of BG data $\mathbf{b^t}$ and the last 4 hours of insulin data $\mathbf{i^t}$ with a sample time of 3 minutes – usefully, the insulin term gives some notion of Insulin on Board (IOB). A window of 4 hours was chosen as Fox et al. empirically found it led to strong performance.

Since we aim for a generalised model, we include the age, weight, and height of the patient into the state space to allow the controller to differentiate between patient classes and between patients within these classes. Note, the height of the virtual patients is not provided by the simulator so we simply take the average height for the patient's age.

Finally, we include the Total Daily Insulin (TDI) of the patient to allow for the controller to give insulin doses with respect to the patient's insulin sensitivity. The only other way to leverage information about the patient's insulin sensitivity is implicitly by directly training on the patient i.e. the model directly observes how the patient's BG responds to insulin dosages. This is obviously not possible as we are aiming for a generalised model. In practice, the TDI of a patient is either already known by the patient, provided by an insulin pump (if they are already using one), calculated naively from their weight, or calculated through consultation with a doctor when switching to this AP in a real-world setting. If we wanted, we could consider adding Gaussian noise to the TDI state element to reflect error in the real-world calculation of the patient's TDI.

Our final state space is given by:

$$\begin{gathered}\mathbf{s_t} = [\mathbf{b^t}, \mathbf{i^t}, \text{weight}, \text{age}, \text{height}, \text{TDI}] \\ \text{where } \mathbf{b^t} = [b_{t-79}, b_{t-78}, \ldots, b_t], \mathbf{i^t} = [i_{t-79}, i_{t-78}, \ldots, i_t]\end{gathered} \tag{7}$$

We could also consider other parameters in our state space. We could include the Insulin-to-Carbohydrate Ratio (ICR) or the Insulin Sensitivity Factor (ISF). However, these are derived from TDI so this information should be implicitly held in the current state space. Moreover, ICR and ISF vary throughout the day so could induce poor performance if they are given as fixed values.

We could also include BMR however, as above, BMR is also simply a linear combination of weight, height, and age so should be implicitly present in the state space. More importantly, we believe this gives too much information with respect to the ground truth meal intake since the CHO content for each meal is derived directly from BMR in the Harris Benedict meal schedule. Similar to TDI, we could add some Gaussian noise to the BMR state element which would replicate some notion of real-world inaccuracy between the patient's estimated BMR and their actual BMR (which is still just an estimate of how much they actually eat).

We also normalise all of our state elements to between 0 and 1 for better training efficiency and performance. We know the upper and lower bounds for BG and insulin doses as they are hard coded into the

environment. For the remaining state elements, we take a lower/upper bound that is a little less/more than the minimum/maxmimum of that value across all of our virtual patients. This is not the best choice in practice since real-world patients with out-of-bounds age, weight, height, or TDI would cause a normalised value of less than 0 or more than 1. Ideally, we would replace this linear normalisation with something like logistic normalisation, rescaled with respect to the distribution of the given state element.

F. Action Space

The action is given simply by the insulin dosage i_t at timestep t. The default action range for the simulator is between -1 and 1 where negatve insulin dosages are mapped to 0.

We restrict the action range upper bound to 0.3 as this reduces the size of the action space to speed up training and improve performance. Fox et al. restrict the upper bound to 0.1, however, this restricts the size of the possible insulin doses too much such that the controller may want to give a higher dosage but is unable to. In our case, actions are always below 0.3 so this doesn't restrict the size of the insulin dosages that can be given.

We also raise the lower bound to -10% of the upper bound to again reduce the size of the action space. Most importantly, this gives further resolution for positive insulin dosages for fine control as they take up a larger proportion of the action range.

However, we do not raise the lower bound all the way to 0 as we still require it to be somewhat below 0 to allow the agent to reliably select the 0 insulin dosage action – this is particularly important for the children who are more insulin sensitive. The lower bound has to be sufficiently below zero since actions in the extreme ranges of the action space are harder to select due to the tanh rescaling in the final layer of the actor's neural network.

Raising the lower bound also has the secondary effect of mimicking a basal-bolus insulin regime in which background basal insulin is given during fasting and corrective boluses are given after meals. This is a very reliable control approach that is used in classical T1D management.

Note that this is achieved with only a 1-dimensional action space i.e. we do not have two actions – one for basal and one for bolus insulin. Note also that in practice there are two types of insulin used for T1D management – long-acting for basal and rapid-acting for bolus. However, our simulator is only equipped with one type of insulin. Having multiple types of insulin could be used to further improve performance, although this would require a higher dimensional action space.

G. Reward Function

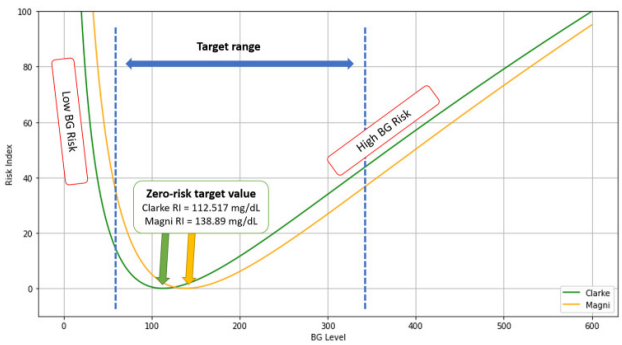


Fig. 3: Clarke and Magni Risk Indices [21]

Our reward function is derived from the Risk Index (RI). Specifically, we use the Clarke RI [39] instead of the alternative Magni RI [40] since the zero-risk target value of the Clarke RI is closer to the centre of the TIR interval. as shown in figure 3, so gives better performance with respect to this metric. This allows for postprantial BG spikes to not stray as far from the TIR interval, thus giving a higher TIR overall. The alternative argument for using the Magni RI is that its tendency towards higher BG may give a safer control policy as lower BG is more dangerous.

The Clarke and Magni Risk Index (RI) are given by:

$$\begin{aligned} \text{RiskIndex}(b_t) &= 10 \cdot (c_0 \cdot \log(b_t)^{c_1} - c_2)^2 \\ \text{where for the Clarke RI } & c_0, c_1, c_2 = 1.509, 1.084, 5.381 \\ \text{and for the Magni RI } & c_0, c_1, c_2 = 3.5506, 0.8353, 3.7932 \end{aligned} \tag{8}$$

Normally the negative of the RI is taken as the reward, however, this causes the agent to learn an optimal policy that ends episodes as quickly as possible to prevent the accumulation of negative reward. One solution is to add a large negative termination penalty but the magnitude of this is difficult to tune. Moreover, Fox et al. experience degenerate behaviour on one virtual patient due to the exploitation of the BG bound for which the hyperglycemia termination penalty is administered.

A different solution that we employ is to rescale the reward to be positive by subtracting the RI at a given timestep from the maximum possible RI. This is a better alternative since the incentive for the agent is to stay alive for as long as possible to keep accumulating positive rewards. This reward is also not subject to exploitation issues surrounding the termination penalty.

H. Training

When training our model we use a Transfer Learning (TL) approach – we first train our model on a no-meal scenario for 250k timesteps before further training this model on a Harris Benedict meal scenario for an extra 500k timesteps. This allows our agent to get to grips with classical BG control before having to deal with large fluctuations in BG due to CHO intake from meal ingestion. Alternatively, we could train our model from scratch on the Harris Benedict meal scenario, however, this takes longer and does not produce results that are as good.

Since we are producing a generalised model we reserve a subset of the virtual patients for testing. We also reserve a subset of the virtual patients for validation to assess how our model performs on unseen patients that do not coincide with our test set. As such, we split our cohort of 30 virtual patients into training/validation/testing sets containing the first 4/3/3 patients of each patient class respectively.

All our training and results are gathered using the seed 42 for reproducibility. We choose 250k and 500k timesteps as from the policy loss it seems that there is no further improvement to the model towards the end of this time period. We choose to evaluate our model on the validation set every 10k timesteps during training and save our model if we receive a new lowest average risk. The remaining hyperparameter choices can be found in the hyperparameters file in our provided code.

Finally, note that actions are selected from the model with exploration noise during training and without exploration noise during evaluation/gathering of results.

I. Models

Primarily, we produce a fully generalised Closed-Loop (CL) Reinforcement Learning (RL) Soft Actor Critic (SAC) Artificial Pancreas (AP) with respect to the methodology described above. We refer to this model as our **generalised model**.

For a lower-bound on performance, we tune a non-generalised Proportional Integral Derivative (PID) baseline model as provided

by the simulator. Importantly, the parameters for PID are loop tuned specific to the patient via a simple grid search, accurate to the correct order of magnitude. We refer to this model as our **PID baseline**.

For an upper-bound on performance, we tune a non-Closed-Loop (CL) meal oracle model by adding the ground truth meal intake to the state space. This is given by the last 4 hours of normalised carbohydrate data with a sample time of 3 minutes as with the BG and insulin state space elements. We refer to this model as our **meal oracle**.

Note that this oracle is actually slightly worse than a meal announcement approach since the meal quantities are given to the model at the exact time of ingestion rather than before. As such, this model circumnavigates the BG/CGM detection delays but not the insulin absorption delay. If it were to be a true meal announcement model, it would additionally preempt the BG/CGM detection delay.

IV. Results

Model		Eug (%)	Hypo (%)	Hyper (%)	Risk
Generalised Model	Avg	80.1	5.3	14.6	3.9
	Train	80.9	5.0	14.0	3.3
	Val	78.9	6.3	14.8	4.0
	Test	80.3	4.5	15.2	4.4
Meal Oracle	Avg	86.8	1.6	11.6	2.6
	Train	89.2	1.1	9.7	2.1
	Val	86.8	0.9	12.3	2.5
	Test	83.6	2.8	13.6	3.2
PID Baseline		77.3	9.7	13.0	4.1
Fox et al. [19]		72.7	0.7	26.2	6.5

TABLE I: Average Metrics for our Models

The average risk and time in eug/hypo/hyperglycemia metrics for all our above mentioned models are given in Table I. The metrics for our two generalised models are split into their respective training, validation, and testing groups. We also include the main results from Fox et al. [19] for comparison as long as we respect the considerations stated in the section II-C.

Our patient-specific plots, from which these average metrics are derived, can be found in appendix I. These include BG trace curves, BG range bar graphs, risk bar graphs, Control-Variability Grid Analysis (CVGA) graphs, and 1-day simulation animations.

Our models themselves and the raw data for our results can be found in our provided code. This also includes tensorboard logs of parameters and metrics recorded throughout training, some of which we include in appendix I for evaluation.

V. Evaluation

In section V-A and section V-B we first evaluate our model with respect to the metrics presented in Table I. We come to the conclusion that our model has comparable performance with the state-of-the-art and performance between that of the lower-bound baseline and upper-bound oracle. We then perform patient-specific analysis between our models in section V-C using the patient-specific graphs presented in appendix I, concluding that some virtual patients are simply more difficult to control due to their metabolic parameters. In section V-D, we take an alternative look at the performance of our model from the perspective of extreme BG vales. In section V-E, we specifically look at the actions (insulin dosages) chosen by our model with respect to the current state using the 1-day simulation animations presented in appendix I. We also briefly analyse the training progress of our model in section V-F using the training plots also presented in appendix I. In section V-G, we argue a case for a level of generalisation between that of full generalisation and patient-specific generalisation – namely class-specific generalisation. Finally, we conclude with a discussion on future work in section V-H.

A. Comparison with the Upper and Lower Bounds

By inspecting the average percentage time in euglycemia (equivalently TIR), we clearly see that our generalised model has impressive BG control with an average TIR (equivalently percentage time in euglycemia) and average Clarke Risk Index (RI) between that of our non-generalised PID baseline (lower bound) and our non-CL meal oracle (upper bound). This metric also shows that our model produces a TIR higher than the recommended goal of 70% for T1D patients and a TIR much higher than the average of 50-60% for T1D patients [41].

Similarly, our generalised model produces an average percentage time in hypoglycemia that is between that of the lower and upper bound models as expected. However, we find that our PID baseline has a higher average time in hyperglycemia than our generalised model. This is likely because the PID algorithm has no concept of the riskiness of different BG levels, only the absolute distance from the optimal BG level of 112mg/dl, which explains its higher time in the riskier state of hypoglycemia.

B. Comparison with State of the Art

With the considerations stated in section II-C, we may compare our results with those of Fox et al. [19]. We see that our model performs much better with respect to the arguably most important TIR metric. This is likely due to the addition of TDI into the state space of our model as it gives important information regarding the patient's metabolic parameters. The justification of the inclusion of TDI into the state space was given in section III-E.

Interestingly, we see that our model has a higher time in hypoglycemia but lower time in hyperglycemia, this is likely due to Fox et al. using the Magni RI whilst we use the Clarke RI which has a lower zero-risk target value as shown in fig. 3.

C. Patient Specific Analysis

When we consider the BG range and risk bar charts in fig. 6 and fig. 7, it is worth remembering that for our fully generalised models, both with and without meal announcement, the first 4 patients of each class were used for training, the following 3 for validation, and the final 3 for testing.

Expectedly, across all models, we see that there are certain patients that are much harder to control – specifically adolescent#002. This is the same patient that Fox et al. also experienced degenerate behaviour with in section III-G. Yamagata et al. [42] also state that they experience poor control over this patient. Alternatively, they use a model-based RL approach which suggests that this is not a model-specific issue but is rather down to the intrinsic metabolic parameters of the patient.

Interestingly, we also notice some patients, unique to specific models, that are particularly hard to control – such as child#005 for our generalised model which tends towards hypoglycemia. This could be due to training/evaluation stochasticity – gathering results from more than one simulated scenario would reduce any observed randomness. Nevertheless, we see no control failures, where the patient's BG becomes too low or too high and terminates.

D. Extreme Blood Glucose

When considering the BG trace curve of our generalised model shown in fig. 4a, we see that on average BG is optimally controlled within the recommended range. However, it is important to consider the extreme BG values attained as these can be extremely harmful as stated in section I-B.

We can roughly see from the BG trace curves that a maximum BG of around 400mg/dl is attained for one/some of our virtual patients. From the CVGA plot shown in fig. 5a, we can clearly identify that this is an outlier that corresponds to only one of our virtual patients. Again, this may be due to training stochasticity but it is also likely that this would be mitigated with a larger training cohort.

From the CVGA plot for our meal oracle model shown in fig. 5b, we see that our CL approach has much better control over extreme BG values with a clustering much closer to the B-zones. On the other hand, the PID baseline, shown in fig. 5c, tends towards the lower-D zone. Loop tuning our patient-specific PID parameters with a more sophisticated approach, such as the Ziegler-Nichols or Cohen-Coon method, would increase the BG control of this baseline.

It would be useful to colour code the virtual patients within the CVGA graphs with respect to their training/validation/testing allocation to observe any differences between the cohorts and assess the generalisation of our models.

E. Insulin Dosages

We now consider the one-day simulation animations in fig. 8 to investigate the actions chosen by our generalised model with respect to the current state.

Firstly, we observe that our controller initially injects a high amount of insulin since our virtual patient starts with no Insulin on Board (IOB). It then mimics the basal-bolus approach as mentioned in section III-F. As expected, we see that our model injects more insulin for higher BG values which correspond to a higher Clarke RI and a lower reward.

We can also see that our model appropriately responds to meal ingestion by injecting a higher dosage of insulin. It would be useful to see the exact insulin response time with respect to meal ingestion as delay is an important factor in T1D control. This includes CHO absorption delay, BG increase delay, CGM detection delay, controller decision delay, and insulin absorption delay. Reducing the controller decision delay is important in reducing the overall delay of the control system to get insulin into the body as soon as possible when it is needed.

We could also consider an input-output relevance analysis on our generalised model by using layer-wise relevance propagation, as performed by Lee et al. [37], to interpret the insulin dosage decisions with respect to the current state. This would further reduce the black-box nature of RL and allow us to understand the relative importance of features within the state space at different times – for example fasting vs postprandial decision making. This would be particularly useful to mitigate safety concerns given its application in healthcare.

F. Training Exploration

We now consider the training parameter plots in fig. 9 to analyse the training pipeline of our generalised model.

From the policy loss shown in fig. 9b we can clearly see the improvement of our model during training over time. It seems that the policy loss has plateaued, meaning that there is no possible further improvement to the model. This is supported by the plateauing of the cumulative reward during training over time in fig. 9d, however, this graph is particularly noisy due to the constant switching of virtual patients/ environments and only reflects the performance on the training cohort of virtual patients (not generalised). We could consider partitioning this graph with respect to the virtual patients to get smoother plots for analysis and to determine which patients were harder to train on.

Alternatively, the validation Risk Index (RI) – given by the Clarke RI on unseen patients in the validation cohort of virtual patients – shown in fig. 9e, provides a look at the generalised performance of our model during training. From this plot, it seems that our model has not finished improving but this is due to the smoothing effect applied to our graphs. From this graph, we see very large dips in performance. This is due to a combination of training stochasticity, a small validation cohort, and failed runs accumulating a disproportionately large Clarke RI.

From the policy loss shown in fig. 9b, we can also see the increase in policy loss during Transfer Learning (TL) as a result of the increase in difficulty of the environment from no-meals to full meals. This correlates with the spike in entropy temperature in fig. 9c as the agent needs to explore the new environment a lot more given that the model is highly-tuned to the no-meal scenario.

From the entropy temperature shown in fig. 9c, we can also see the decrease in entropy temperature causing the model to increase exploitation over exploration as it tends towards an optimal solution – this is particularly evident towards the end of the no-meal scenario.

G. Class-Specific Generalisation

From the range and risk bar graphs for our generalised model shown in fig. 6a and fig. 7a, we see that BG control is best over the adult cohort, worst over the child cohort, and somewhere in between for the adolescent cohort. This is partially due to the more difficult nature of the child environments due to their higher insulin sensitivity; partially also due to the higher metabolic parameter variability of the adolescent cohort.

It is also due to the action range – currently set to deliver a maximum possible insulin dosage of 0.3 units as justified in section III-F. This is evident in the one-day simulation animations shown in fig. 8 where child#008 uses no more than 0.2 units of insulin at any given time so a third of the action range is not in use. This reduces the resolution and fine control of the agent which is particularly important for the child cohort due to their high insulin sensitivity.

Therefore, a possible solution is to use a looser definition of generalisation between that of full generalisation and patient-specific generalisation – namely class-specific generalisation. The three class-specific models can then be given an appropriate action space upper bound of 0.3, 0.2, and 0.1 for the adult, adolescent, and child models respectively. This should increase performance as it lessens the amount of generalisation required by the model whilst also increasing the resolution of the agent's actions within the action space.

In a real-world setting, three models is just as easy to provide to patients as one model by simply allocating the appropriate model with respect to the patient's age. Therefore, this simpler definition of generalisation does not conflict with one of the primary goals of our research – to increase the ease of switching to an AP system.

H. Future Work

Foremost we are limited by the environment – namely the outdated simulator and the lack of virtual patients. A wider variety of virtual patients would increase the reliability and performance of our model when operating on unseen patients with wildly different parameters. This is particularly problematic since we are splitting a cohort of only 30 virtual patients into 3 subsets leaving few patients for training, validating, and testing.

One of the biggest simulation limitations of the environment is the lack of physical activity which can cause extremely large fluctuations in BG – much like meal intake. More advanced simulators containing physical activity simulation do already exist, but we are yet to see them reformatted for reinforcement learning.

We could also consider developing a dual-action model that utilises insulin types of different rapidity or a dual-hormone model that

utilises secondary hormones such as glucagon to further control BG. However, we are once again limited by the available simulators.

There is still work to be done fine-tuning the state space elements such as the length and resolution of the BG and insulin history. The action space could also be automatically tuned, in a non-generalised setting, to give an optimal patient-specific action range. This would reduce the size of the unused action space, increase the resolution, and increase the fine control of insulin dosage selection. This was hinted at when we suggested splitting our model into three, one for each patient class, and adjusted the upper bound accordingly.

A further investigation into the training and testing consistency is also required. The current model is also somewhat dependent on the adherence to the Harris Benedict meal schedule – although this scenario does include some randomness. Therefore, an investigation into unlikely meal scenarios, such as large single CHO ingestions, would also be beneficial.

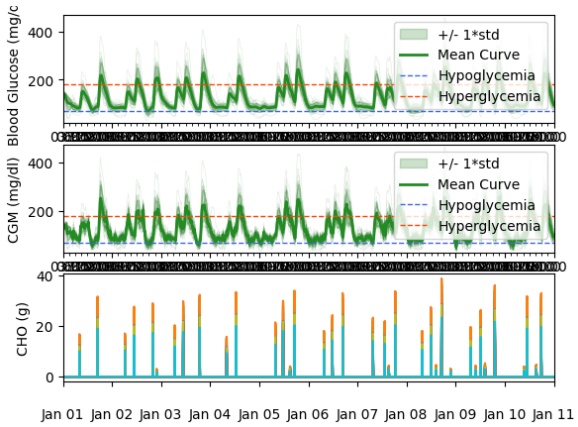
We should also consider safety policies. This could include a fallback algorithm in case our model fails or a separate module that controls the safety aspect of our algorithm. We could also consider other modules such as meal detection, exercise detection, or similar to further improve control.

The final step is clinical trials on real patients so long as we have sufficient safety policies in place.

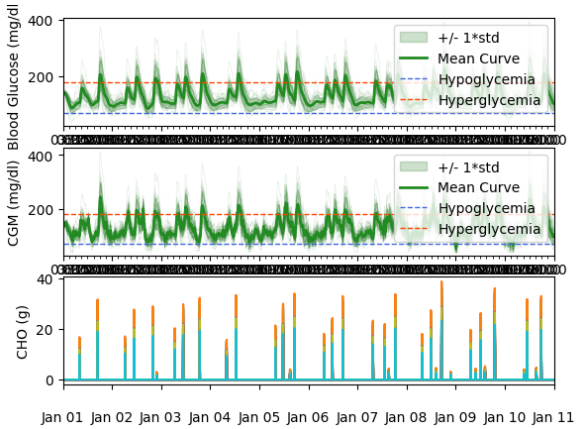
VI. Conclusion

We produce a generalised Closed-Loop (CL) Reinforcement Learning (RL) Artificial Pancreas (AP) using the algorithm Soft Actor Critic (SAC) with the two extensions of a Prioritised Experience Replay (PER) buffer and Emphasising Recent Experiences (ERE). We split the cohort into 3 sets of training, validation, and testing patients. For training, we make multiple justified implementational design choices regarding the state space, action space, and reward function. Importantly, we provide various corrected meal scenarios for the python port simglucose of the FDA-approved UVA/Padova simulator. Our model is trained using a Transfer Learning (TL) approach from a no-meal scenario to the Harris Benedict meal scenario with the additional use of a learning rate scheduler. We save the model that produces the lowest Clarke Risk Index (RI) on an unseen validation set. Finally, we test this fully trained model on an unseen testing set to achieve a model with an average 80.3% Time in Range (TIR). This surpasses our PID baseline, as well as other state-of-the-art approaches, but does not achieve the performance of our non-CL ground truth meal oracle. Still, our CL approach is able to control BG to a high degree of accuracy without any human intervention. Our unique contribution of generalisation increases the ease and accessibility of switching to a CL AP system in a real-world setting as no training is required on unseen patients. To further improve performance it may be worth considering class-specific generalisation – a level of generalisation between that of full generalisation and patient-specific generalisation – which does not affect the ease of switching to an AP system. Our clean, well-documented code with associated models provides a useful tool for further research within the field.

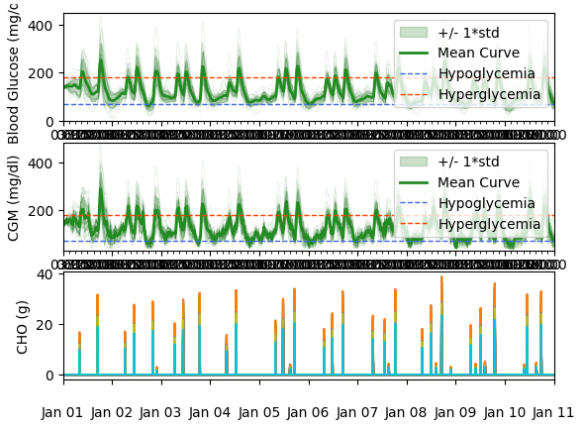
Appendix I



(a) Generalised Model

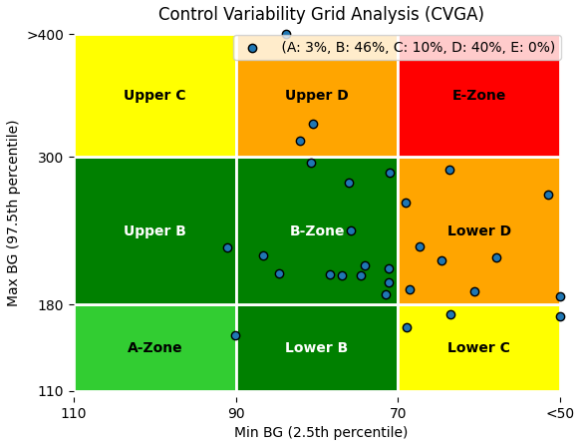


(b) Meal Oracle

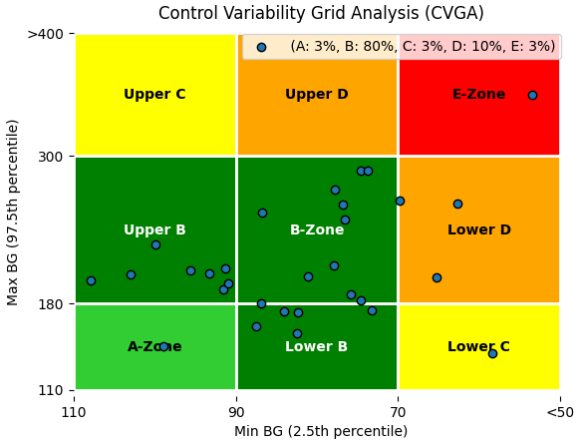


(c) PID Baseline

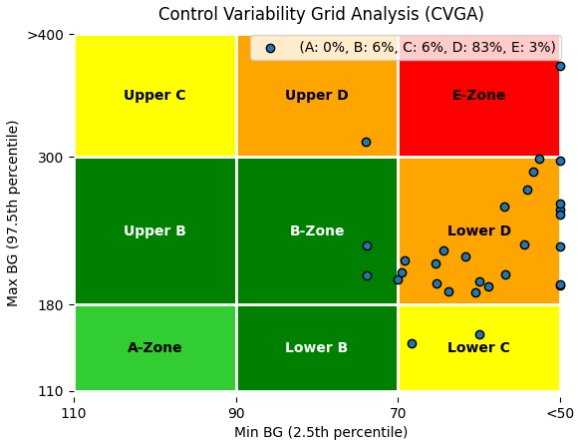
Fig. 4: Blood Glucose trace curves for all 30 patients over a 10 day Harris Benedict meal scenario. The mean and standard deviation of the Blood Glucose and Continuous Glucose Monitor traces over all patients are shown with respect to time. The meal ingestion quantities are also shown with respect to time. Note that we can clearly see that the meal sizes scale with respect to the patient's Basal Metabolic Rate as expected.



(a) Generalised Model

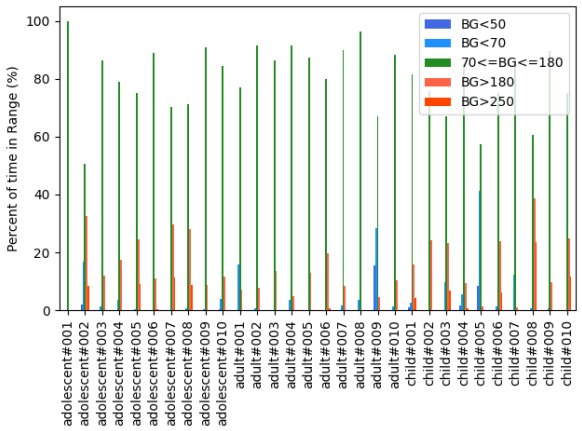


(b) Meal Oracle

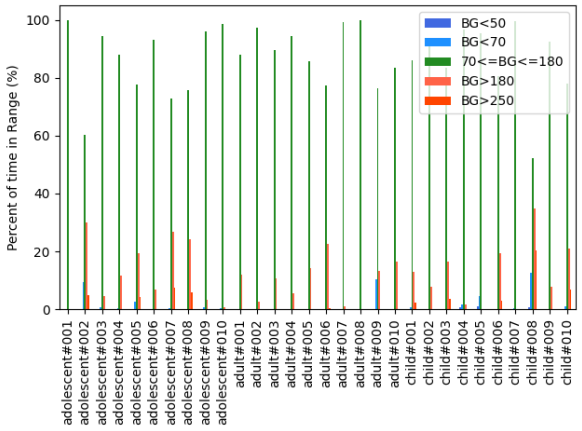


(c) PID Baseline

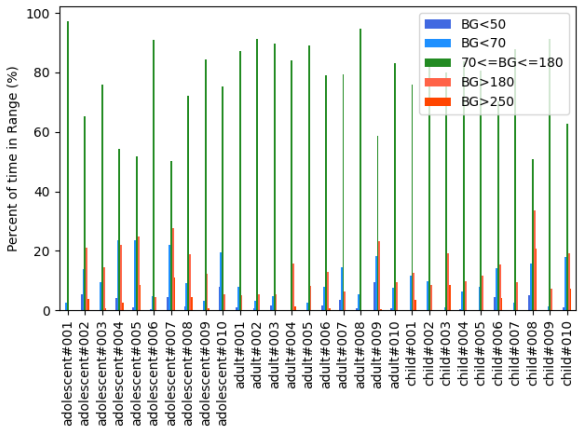
Fig. 5: Control-Variability Grid Analysis plots show the quality of glycemic control with respect to all 30 virtual patients over the 10 day scenario. Each patient is represented by a blue dot that is positioned with respect to the patient's minimum and maximum Blood Glucose values over the 10 day scenario.



(a) Generalised Model

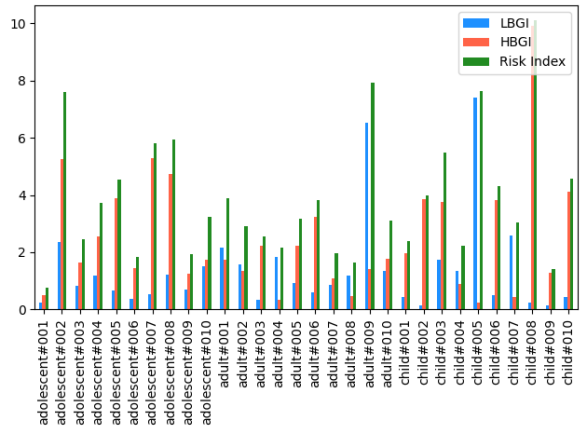


(b) Meal Oracle

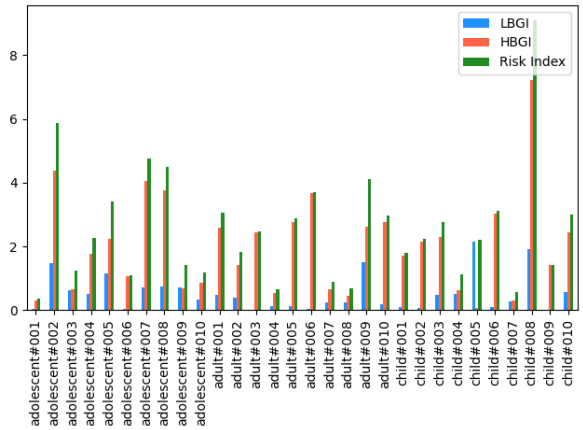


(c) PID Baseline

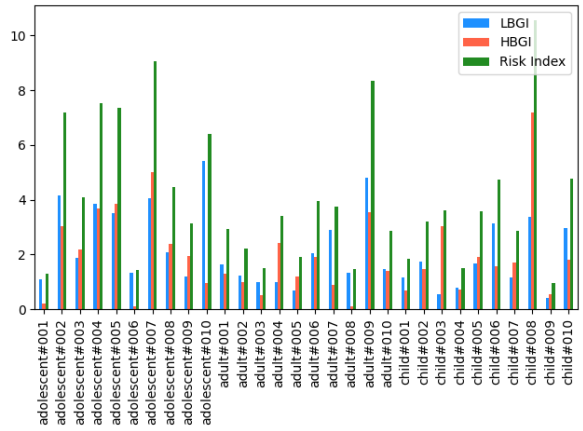
Fig. 6: Blood Glucose range bar graphs show the percentage time spend in a discrete set of pre-defined ranges for each of the 30 virtual patients over the 10 day scenario. The ranges are as follows: extreme hypoglycemia, hypoglycemia, euglycemia, hyperglycemia, extreme hyperglycemia. The percentage time in euglycemia is analagous to the metric Time in Range.



(a) Generalised Model

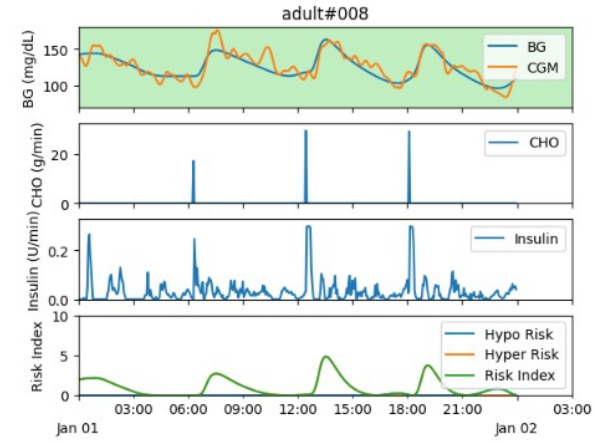


(b) Meal Oracle

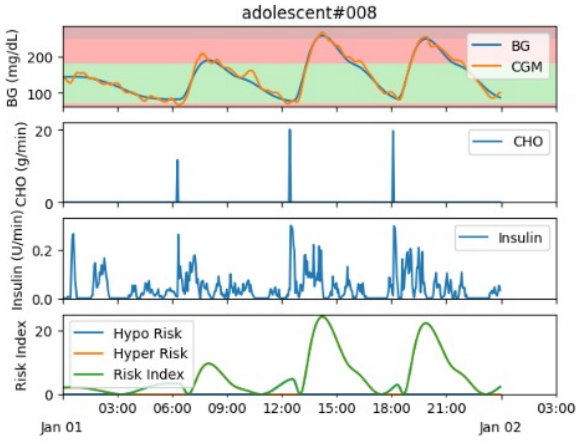


(c) PID Baseline

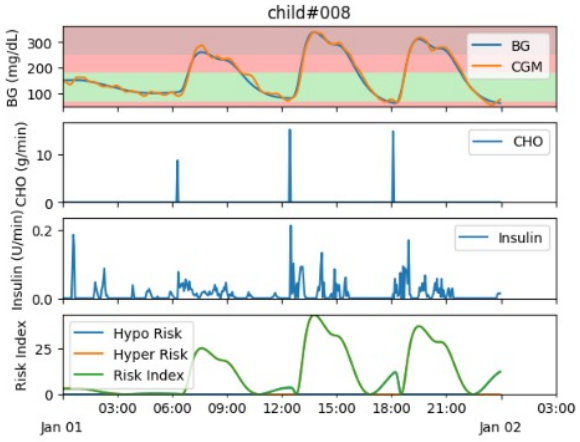
Fig. 7: Blood Glucose risk bar graphs show the Clarke Risk Index for each of the 30 virtual patients over the 10 day scenario. The Clarke Risk Index is partitioned into both high and low Blood Glucose risk such that the sum of the two gives the total Clarke Risk Index.



(a) Adult

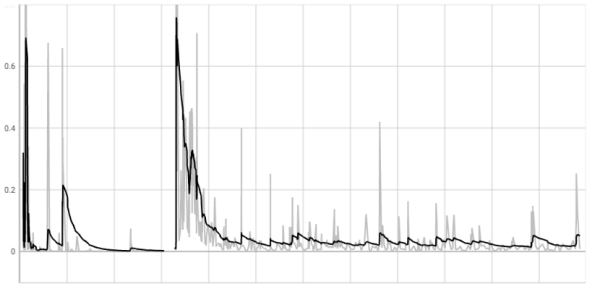


(b) Adolescent

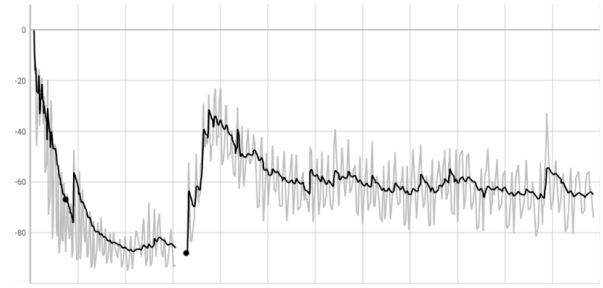


(c) Child

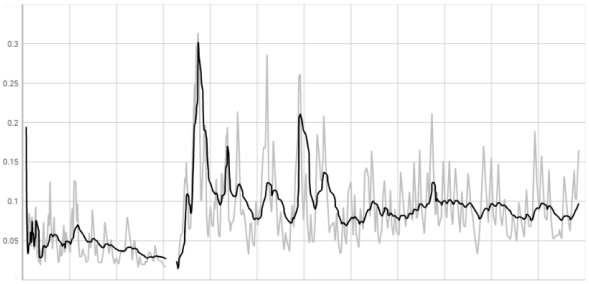
Fig. 8: One-day simulations of our generalised model for the first patient in the testing cohort of each patient class. The blood glucose, meal ingestion, insulin dosage, and Clarke Risk Index are shown with respect to time.



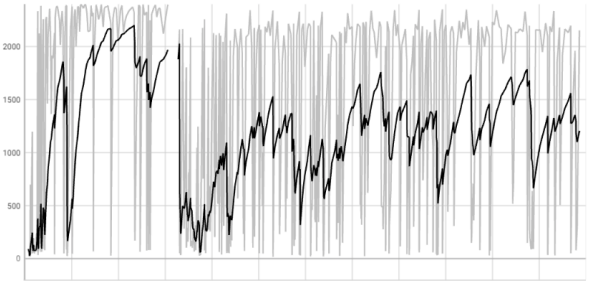
(a) Critic Loss



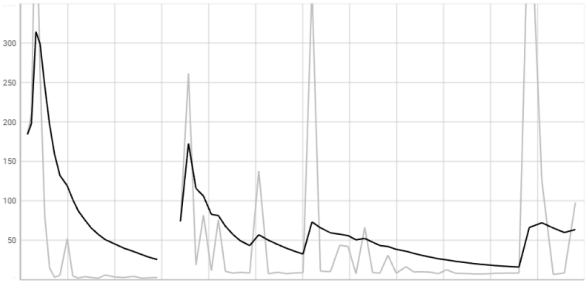
(b) Policy Loss



(c) Entropy Temperature



(d) Cumulative Reward on the Training Set



(e) Clarke Risk Index on the Validation Set

Fig. 9: A selection of our generalised model's training parameters with respect to time during the full training pipeline. The raw data is presented in grey and the smoothed curves in black.

Acronyms

AC	Actor-Critic.
AI	Artificial Intelligence.
AP	Artificial Pancreas.
BG	Blood Glucose.
BMR	Basal Metabolic Rate.
CGM	Continuous Glucose Monitor.
CHO	Carbohydrate.
CL	Closed-Loop.
CSII	Continuous Subcutaneous Insulin Infusion.
CVGA	Control-Variability Grid Analysis.
DDPG	Deep Deterministic Policy Gradient.
DM	Diabetes Mellitus.
DSS	Decision Support System.
ERE	Emphasising Recent Experiences.
FDA	Food and Drug Administration.
FLC	Fuzzy Logic Control.
HbA1c	Glycated Haemoglobin.
ICR	Insulin-to-Carbohydrate Ratio.
IIT	Intensive Insulin Therapy.
IOB	Insulin on Board.
ISF	Insulin Sensitivity Factor.
MAPS	Multiple-Input Artificial Pancreas System.
MDI	Multiple Daily Injections.
MDP	Markov Decision Process.
ML	Machine Learning.
MPC	Model Predictive Control.
PER	Prioritised Experience Replay.
PID	Proportional Integral Derivative.
POMDP	Partially-Observable Markov Decision Process.
PPO	Proximal Policy Optimisation.
RI	Risk Index.
RL	Reinforcement Learning.
SAC	Soft Actor Critic.
SMBG	Self Monitoring of Blood Glucose.
T1D	Type 1 Diabetes.
T2D	Type 2 Diabetes.
TD	Temporal Difference.
TDI	Total Daily Insulin.
TIR	Time in Range.
TL	Transfer Learning.

References

[1] *Diabetes*. URL: https://www.who.int/news-room/fact-sheets/detail/diabetes.

[2] B Zhou et al. "Worldwide trends in diabetes since 1980: A pooled analysis of 751 population-based studies with 4·4 million participants". In: *The Lancet* 387.10027 (2016), pp. 1513–1530. DOI: 10.1016/s0140-6736(16)00618-8.

[3] Bernhard Wernly et al. "Hypoglycemia but not hyperglycemia is associated with mortality in critically ill patients with diabetes". In: *Medical Principles and Practice* 28.2 (2018), pp. 186–192. DOI: 10.1159/000496205.

[4] Viral N. Shah et al. "Closed-loop system in the management of diabetes: Past, present, and future". In: *Diabetes Technology & Therapeutics* 16.8 (2014), pp. 477–490. DOI: 10.1089/dia.2014.0193.

[5] Sherita Hill Golden and Tamar Sapir. "Methods for insulin delivery and glucose monitoring in diabetes: Summary of A comparative effectiveness review". In: *Journal of Managed Care Pharmacy* 18.6 Supp A (2012), pp. 1–17. DOI: 10.18553/jmcp.2012.18.s6-a.1.

[6] B. Wayne Bequette. "A critical assessment of algorithms and challenges in the development of a closed-loop artificial pancreas". In: *Diabetes Technology & Therapeutics* 7.1 (2005), pp. 28–47. DOI: 10.1089/dia.2005.7.28.

[7] Nichole S. Tyler and Peter G. Jacobs. "Artificial Intelligence in decision support systems for type 1 diabetes". In: *Sensors* 20.11 (2020), p. 3214. DOI: 10.3390/s20113214.

[8] 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 (2020), p. 101836. DOI: 10.1016/j.artmed.2020.101836.

[9] Katrin Lunze et al. "Blood glucose control algorithms for type 1 diabetic patients: A methodological review". In: *Biomedical Signal Processing and Control* 8.2 (2013), pp. 107–119. DOI: 10.1016/j.bspc.2012.09.003.

[10] A.S. Brazeau et al. "Carbohydrate counting accuracy and blood glucose variability in adults with type 1 diabetes". In: *Diabetes Research and Clinical Practice* 99.1 (2013), pp. 19–23. DOI: 10.1016/j.diabres.2012.10.024.

[11] Anthony Pease et al. "Time in range for multiple technologies in type 1 diabetes: A systematic review and network meta-analysis". In: *Diabetes Care* 43.8 (2020), pp. 1967–1975. DOI: 10.2337/dc19-1785.

[12] Saúl Langarica et al. "A meta-learning approach to personalized blood glucose prediction in type 1 diabetes". In: *Control Engineering Practice* 135 (2023), p. 105498. DOI: 10.1016/j.conengprac.2023.105498.

[13] K. van Heusden et al. "Control-relevant models for glucose control using a priori patient characteristics". In: *IEEE Transactions on Biomedical Engineering* 59.7 (2012), pp. 1839–1849. DOI: 10.1109/tbme.2011.2176939.

[14] 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 (2013), pp. 661–673. DOI: 10.1586/17434440.2013.827515.

[15] *MiniMed™ 780G system campaign page*. URL: https://www.medtronic-diabetes.com/en-gb/minimed-780g-system-info.

[16] Hamid R. Berenji. "A reinforcement learning–based architecture for Fuzzy Logic Control". In: *Readings in Fuzzy Sets for*

Intelligent Systems (1993), pp. 368–380. DOI: 10.1016/b978-1-4832-1450-4.50043-2.

[17] Taiyu Zhu et al. "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 (2021), pp. 1223–1232. DOI: 10.1109/jbhi.2020.3014556.

[18] Taiyu Zhu et al. "An insulin bolus advisor for type 1 diabetes using Deep Reinforcement Learning". In: *Sensors* 20.18 (2020), p. 5058. DOI: 10.3390/s20185058.

[19] Ian Fox et al. "Deep reinforcement learning for closed-loop blood glucose control". In: *arXiv.org* (Sept. 2020). URL: https://arxiv.org/abs/2009.09051.

[20] Francesco Di Felice, Alessandro Borri, and Maria Domenica Benedetto. "Deep reinforcement learning for closed-loop blood glucose control: Two approaches". In: *IFAC-PapersOnLine* 55.40 (2022), pp. 115–120. DOI: 10.1016/j.ifacol.2023.01.058.

[21] Phuwadol Viroonluecha, Esteban Egea-Lopez, and Jose Santa. "Evaluation of blood glucose level control in type 1 diabetic patients using deep reinforcement learning". In: (2021). DOI: 10.21203/rs.3.rs-1095721/v1.

[22] Min Hyuk Lim et al. "A blood glucose control framework based on reinforcement learning with safety and interpretability: In silico validation". In: *IEEE Access* 9 (2021), pp. 105756–105775. DOI: 10.1109/access.2021.3100007.

[23] *MLD3 / RL4BG*. URL: https://gitlab.eecs.umich.edu/mld3/rl4bg.

[24] Chirath Hettiarachchi et al. "Integrating multiple inputs into an artificial pancreas system: Narrative literature review". In: *JMIR Diabetes* 7.1 (2022). DOI: 10.2196/28861.

[25] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *arXiv.org* (Aug. 2018). URL: https://arxiv.org/abs/1801.01290.

[26] Benjamin Eysenbach and Sergey Levine. "If Maxent RL is the answer, what is the question?" In: *arXiv.org* (Oct. 2019). URL: https://arxiv.org/abs/1910.01913.

[27] Tuomas Haarnoja et al. "Soft Actor-Critic Algorithms and Applications". In: *arXiv.org* (Jan. 2019). URL: https://arxiv.org/abs/1812.05905.

[28] BY571. *Soft-Actor-Critic-and-Extension*. URL: https://github.com/BY571/Soft-Actor-Critic-and-Extensions.

[29] Che Wang and Keith Ross. "Boosting Soft Actor-Critic: Emphasizing Recent Experience without Forgetting the Past". In: *arXiv.org* (June 2019). URL: https://arxiv.org/abs/1906.04009.

[30] Boris P. Kovatchev et al. "In silicopreclinical trials: A proof of concept in closed-loop control of type 1 diabetes". In: *Journal of Diabetes Science and Technology* 3.1 (2009), pp. 44–55. DOI: 10.1177/193229680900300106.

[31] Chiara Dalla Man et al. "The UVA/PADOVA Type 1 Diabetes Simulator: New Features". In: *Journal of Diabetes Science and Technology* 8.1 (2014), pp. 26–34. DOI: 10.1177/1932296813514502.

[32] 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 (2018), pp. 273–281. DOI: 10.1177/1932296818757747.

[33] jxx123. *JXX123/simglucose: A type-1 diabetes simulator implemented in Python for Reinforcement Learning Purpose*. URL: https://github.com/jxx123/simglucose.

[34] Mudassir Rashid et al. "Simulation software for assessment of nonlinear and adaptive multivariable control algorithms: Glucose–insulin dynamics in type 1 diabetes". In: *Computers and Chemical Engineering* 130 (2019), p. 106565. DOI: 10.1016/j.compchemeng.2019.106565.

[35] Navid Resalat et al. "A statistical virtual patient population for the glucoregulatory system in type 1 diabetes with integrated exercise model". In: *PLOS ONE* 14.7 (2019). DOI: 10.1371/journal.pone.0217301.

[36] Roberto Visentin, Claudio Cobelli, and Chiara Dalla Man. "The padova type 2 diabetes simulator from triple-tracer single-meal studies: In silico trials also possible in rare but not-so-rare individuals". In: *Diabetes Technology & Therapeutics* 22.12 (2020), pp. 892–903. DOI: 10.1089/dia.2020.0110.

[37] Seunghyun Lee et al. "Toward a fully automated artificial pancreas system using a bioinspired reinforcement learning design: In silico validation". In: *IEEE Journal of Biomedical and Health Informatics* 25.2 (2021), pp. 536–546. DOI: 10.1109/jbhi.2020.3002022.

[38] J. Arthur Harris and Francis G. Benedict. "A biometric study of human basal metabolism". In: *Proceedings of the National Academy of Sciences* 4.12 (1918), pp. 370–373. DOI: 10.1073/pnas.4.12.370.

[39] William Clarke and Boris Kovatchev. "Statistical tools to analyze continuous glucose monitor data". In: *Diabetes Technology and Therapeutics* 11.S1 (2009). DOI: 10.1089/dia.2008.0138.

[40] Lalo Magni et al. "Model predictive control of type 1 diabetes: An in silico trial". In: *Journal of Diabetes Science and Technology* 1.6 (2007), pp. 804–812. DOI: 10.1177/193229680700100603.

[41] *Time in range*. Mar. 2023. URL: https://diatribe.org/time-range.

[42] Taku Yamagata et al. "Model-Based Reinforcement Learning for Type 1 Diabetes Blood Glucose Control". In: *arXiv.org* (Oct. 2020). URL: https://arxiv.org/abs/2010.06266.