G2P2C – A Deep Reinforcement Learning Algorithm for Glucose Control by Glucose Prediction and Planning in Type 1 Diabetes

Chirath Hettiarachchi[a,*], Nicolo Malagutti[b], Christopher J Nolan[c], Hanna Suominen[a,d] and Elena Daskalaki[a]

[a] *School of Computing, College of Engineering & Computer Science, Australian National University, Canberra, Australia*
[b] *School of Engineering, College of Engineering & Computer Science, Australian National University, Canberra, Australia*
[c] *ANU Medical School, College of Health & Medicine, Australian National University, Canberra, Australia*
[d] *Department of Computing, Faculty of Technology, University of Turku, Turku, Finland*

ARTICLE INFO

Keywords:
Algorithms
Control Systems
Diabetes Mellitus, Type 1
Insulin Infusion Systems
Pancreas, Artificial
Reinforcement Learning

ABSTRACT

Developing diagnostic and treatment solutions for medical applications is often challenging due to the complex dynamics, partial observability, high inter- and intra-population variability, and the presence of unknown delays and disturbances. A characteristic case is the control of glucose concentration in people with Type 1 Diabetes (T1D) through the administration of exogenous insulin. The above complexities, enhanced by the significant cognitive burden associated with the estimation of optimal insulin dosing related to daily activities such as food intake and exercise, call for advanced insulin administration solutions towards a fully automated Artificial Pancreas System (APS). Reinforcement Learning (RL) is currently being explored in the development of APS thanks to its demonstrated potential in problems characterized by complex dynamics and uncertainties. Despite the progress, RL algorithms in T1D still require manual estimation and announcement of meal carbohydrate (CHO) content or rely on small meal scenarios. In this study, we proposed G2P2C, a deep RL algorithm, which aims to fully automate glucose control in T1D, eliminating the need for CHO estimation and announcement. G2P2C was designed based on the state-of-the-art Proximal Policy Optimization (PPO) algorithm, augmented by two novel optimization phases: (i) model learning and (ii) planning. The former integrated an auxiliary learning task to learn a glucose dynamics model. The latter fine-tuned the learned control strategy to a short-time horizon by simulating glucose trajectories into the future. We evaluated the performance of G2P2C *in-silico* on a challenging meal protocol (180g of CHO per day) using an open-source version of a FDA-approved T1D simulator for 20 subjects (10 adults and 10 adolescents). G2P2C was compared against the PPO algorithm and two basal-bolus (BB) clinical treatment strategies, which involve manual meal announcement and CHO estimation with automated correction insulin boli for elevated glucose. G2P2C obtained statistical significant ($P < 0.05$) reward improvements compared to PPO in 18 out of 20 subjects, while maintaining a lower failure rate. In addition, G2P2C achieved a time in range of 73% and 64% for the adult and adolescent cohorts, respectively, outperforming BB strategies in the adult cohort although no meal announcement was performed. The control performance and algorithmic characteristics of G2P2C show promise as a candidate algorithm for glucose control in APS. We released the codebase of G2P2C (`https://github.com/chirathyh/G2P2C`) and an online demonstration tool (`http://capsml.com/`), where users can perform custom simulations to compare G2P2C against BB strategies, under the MIT license.

1. Introduction

Type 1 Diabetes (T1D) is a chronic disease that affects millions of people worldwide, leading to a life-long optimization problem of blood glucose regulation [1]. In healthy individuals, the endocrine pancreas maintains glucose homeostasis through regulated insulin secretion. However, in people with T1D, this process fails due to autoimmune destruction of the insulin producing cells. Hence, an appropriate amount of insulin must be administered from exogenous sources to control the blood glucose concentrations. Glucose control is challenging due to the varying insulin requirements related

[*] Corresponding author

`chirath.hettiarachchi@anu.edu.au` (C. Hettiarachchi); `nicolo.malagutti@anu.edu.au` (N. Malagutti); `christopher.nolan@anu.edu.au` (C.J. Nolan); `hanna.suominen@anu.edu.au` (H. Suominen); `eleni.daskalaki@anu.edu.au` (E. Daskalaki)

ORCID(s): 0000-0002-7702-0718 (C. Hettiarachchi); 0000-0003-4967-3348 (N. Malagutti); 0000-0002-6964-3819 (C.J. Nolan); 0000-0002-4195-1641 (H. Suominen); 0000-0002-7665-7039 (E. Daskalaki)

to sleep patterns, meals, and exercise. The objective is to improve the time spent in the normoglycemic range (70–180 mg/dL) while minimizing low blood glucose (hypoglycemia) and high blood glucose (hyperglycaemia) which are detrimental to health. Commercial hybrid closed-loop systems have recently been introduced in which glucose levels in subcutaneous interstitial fluid are continuously monitored through a continuous glucose monitor (CGM), and insulin is infused subcutaneously via a pump attached to the body [2, 3, 4]. These systems use Proportional Integral Derivative (PID) [5] controllers and Model Predictive Controllers (MPC) [6] to calculate the insulin requirement. However, they are not fully automatic and require manual meal announcement and calculation of an insulin ***bolus***, a process that adds a substantial cognitive burden to pump users [7]. In particular, they must first estimate their meal's carbohydrate (CHO) content, typically 20 minutes before consumption and enter this amount into the pump system. The accuracy of each meal-related insulin bolus also depends on the insulin to CHO parameter settings entered to the pump system by the user [8, 9]. Depending on the hybrid closed-loop system being used, the user also has to decide on whether to manually initiate correction insulin boli according to insulin sensitivity factor settings [8, 9]. Moreover, the human error associated with manual insulin estimation often leads to sub-optimal glucose control [10]. Hence, these systems are fully automated only for controlling the background insulin levels (also known as ***basal*** insulin), which are associated with fasting periods (e.g., sleeping) and underlie the meal boli during the active part of the day based on manual user input. Ongoing research has been exploring control strategies for an Artificial Pancreas System (APS), which aims to automate insulin administration completely to alleviate the burden on users [11].

The glucoregulatory system is a complex non-linear dynamical system where high inter- and intra- population variability is present [12], as well as uncertainties and disturbances associated with meals, exercise, stress, and other daily events [13]. Furthermore, the delays associated with subcutaneous glucose sensing [14] and insulin action [15], add to the complexity of glucose control. Existing control strategies explored in this problem, such as PID and MPC, are affected by these challenging characteristics [16]. Reinforcement Learning (RL) [17] is a class of machine learning algorithms that have been shown to perform well under unknown variable delays (through delayed reward mechanisms); in learning complex non-linear dynamics; handling uncertainties and disturbances; and in personalization [16]. Hence, RL-based algorithms are currently being explored in glucose control in T1D [18], among other healthcare applications such as propofol dosing in general anaesthesia [19] and multi-cytokine therapy for sepsis [20].

In a RL algorithm, an ***agent*** is trying to achieve a specified goal, by interacting with its underlying environment. The agent takes ***actions***, which result in state transitions and a feedback signal ***(reward)*** which evaluates the transitions related to the goal pursued. The agent uses its ***experiences*** (state, action, reward transitions) to learn a control strategy (***policy***) to maximize the expected cumulative reward ***(return)*** from any given initial state. The expected return for a given state is also called the ***value*** (please refer to the Method Section below for the formal definitions of the terminology). RL has been successfully applied in games [21], continuous control problems such as 3D humanoid motion problems, and physics simulations [22]. However, the use of RL in real-world artificial intelligence applications is still in the early stages with challenges related to critical safety constraints, transferability (*in-silico* to *in-vivo* in a sample-efficient manner), explainability, partial observability, and high-dimensional continuous state or action spaces still requiring extensive research [23].

The aim of this paper was to introduce a novel RL-based algorithm, which fully automates glucose control by eliminating the requirement for CHO estimation and manual insulin dose calculation. Our algorithm extended the state-of-the-art model-free Proximal Policy Optimization (PPO) algorithm [22], used in continuous control applications, by introducing two novel optimization phases, model learning and planning, to improve its performance in the challenging glucose control problem. In view of these novel added features, we named our algorithm G2P2C — Glucose Control by Glucose Prediction and Planning (Figure 1). In the model-learning phase, a glucose dynamics model is learned as an auxiliary learning task followed by the planning phase to fine-tune the learned control strategy to a short-time horizon by conducting model-based simulations to improve safety (e.g., to avoid short-term severe hypoglycemia). Therefore the proposed approach could combine the characteristics of both model-free and model-based methods. The performance of G2P2C was assessed *in-silico* using an open-source T1D simulator based on the FDA-approved UVA/PADOVA 2008 model [12]. We comparatively assessed G2P2C against PPO and a basal-bolus (BB) clinical insulin treatment strategy in which meal announcement and CHO estimation are required.

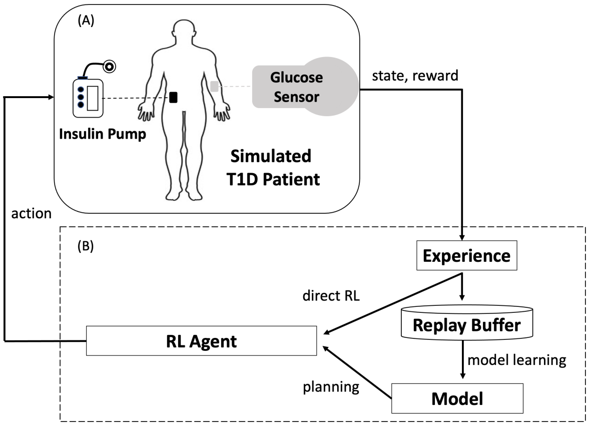

Figure 1: An Artificial Pancreas System (APS) consists of (A) a glucose sensor and an insulin pump attached to a Type 1 Diabetes (T1D) patient and (B) a control algorithm. In this study, we proposed a Reinforcement Learning (RL) based algorithm, G2P2C (Glucose Control by Glucose Prediction and Planning) as the control algorithm, which integrates two novel optimization phases; model learning and planning.

2. Related Work

The use of deep RL for continuous control has gained much attention due to applications such as locomotion, self-driving, and dexterous manipulation tasks [24, 25, 26, 27]. Both on-policy (REINFORCE [28], TRPO [29], A3C [30], PPO [22]) and off-policy (DDPG [24], Soft Actor Critic (SAC) [25]) algorithms have been used in model-based [31] and model-free [32] schemes. Model-based RL algorithms rely on learning a model of the environment in order to optimize their control policy. They are more sample-efficient than model-free methods and benefit from generalizing well to new tasks and environments [33]. On the other hand, they heavily depend on the accuracy of the learnt model [34]. Model-free RL algorithms have outperformed model-based algorithms in various domains where model learning is challenging. However, they require millions of trials for learning [33, 21]. Several prior works have focused on combining model-free and model-based paradigms to design RL systems [35, 36, 33, 21].

PPO [22] is a widely used model-free on-policy algorithm that has shown promise in many RL tasks [37]. It is an actor-critic method consisting of policy and value functions. The optimization objective of the policy function is to learn a stochastic policy, while the objective of the value function is to learn the value of the system state. In standard on-policy policy gradient methods, past data cannot be used to improve the current policy. Hence, they are sample-inefficient. The PPO algorithm improves the sample efficiency by using a clipped objective function, which enables multiple updates for a given data sample. This objective also avoids excessive changes to the policy while training. PPO is implemented using either a shared neural network for the policy and value functions or separate neural networks [38]. The former facilitates feature sharing between the two functions, while the latter avoids interference between the two optimization objectives. The Phasic Policy Gradient (PPG) algorithm [38] is a variant of PPO, which uses separate neural networks and introduces an additional auxiliary learning task of value function estimation. The auxiliary learning facilitates the distillation of features between the two networks and further improves the sample efficiency. Similar ideas based on auxiliary learning tasks have been explored in Deep Q-Learning approaches where the sample efficiency and policy adaptation during deployment were improved [39, 40]. Hence, auxiliary learning tasks are increasingly being used in deep RL algorithms. Model-learning is a popular auxiliary learning task that is particularly useful in updating policy and value functions through planning and action selection [41]. Previous work in RL carries out auxiliary model-learning by predicting latent state representations [39, 42], predicting observations / system states [21, 33], and through estimating future rewards, value and policy functions [41].

A learnt model of the environment can be used for simulations and planning. *Planning* is a process that uses a learned model to improve the policy, where the RL algorithm interacts with the modelled environment. The two distinct approaches to planning are state-space planning and plan-space planning [17]. In state-space planning, the RL algorithm uses the model to simulate experiences, which are then used to update the value function and, ultimately, the policy function. Simulating experiences is valuable when real experiences are costly and limited. Dyna-Q [35] is such an algorithm where the experiences of the RL algorithm are used for both model-learning and planning in an online manner. In plan-space planning, the planning is conducted as a search over the space of plans (e.g., partial order planning). Anthony et al. [43] have introduced such a method in which an expert policy based on Monte Carlo Tree Search (MCTS) is used for planning. The planning horizon reflects the objective of the control task. Most real-life control problems often have both short-term and long-term objectives. Designing reward functions to capture both objectives is challenging or even infeasible [44]. Khorasgani et al. [44] have proposed a model-based RL approach based on the Q-learning algorithm to focus on the two objectives by modelling short/long-term prediction models. In the discounted RL setting, a discount factor (γ) is used to optimize an exponentially decreasing function of the future return. The magnitude of γ establishes an effective horizon for optimizing the agent [17]. Several prior studies have also focused on learning value functions over multiple time horizons [45, 46, 47] to address short/long-term objectives.

The application of RL on the regulation of blood glucose levels in T1D dates back to 2012 when Daskalaki et al. [18, 48] explored the use of actor-critic methods. However, due to the limitations of the simulators available at the time, the task was restricted to daily updates to the control strategy [48, 49, 50]. The development of the open source Simglucose simulator in 2018 [51] (based on FDA-approved UVA/PADOVA-2008 Model [12]) and UVA/PADOVA (2014) simulator [52] has resulted in recent studies with real-time control strategies (e.g., every minute). Many of these studies are hybrid and based on Deep Q-Learning approaches that require manual decision-making [53, 54, 55, 56, 57, 58]. Lim et al. [59] has developed a system that uses a SAC algorithm to estimate insulin guided by PID control. The system also uses machine learning (i.e., a random forest regressor and a dual attention network) for glucose prediction. These systems have predominantly used discrete handcrafted insulin action spaces [53, 54, 55]. The latest research focus is on developing fully autonomous RL systems for glucose control [60]. Fox et al. [61] have proposed a system based on the SAC algorithm and Lee et al. [62] a bio-inspired RL approach using PPO where reward functions and discount factors of the algorithm reflect subject-specific pharmacological characteristics and temporal homeostatic objectives. A systematic review of studies on RL for glucose control is presented in Tejedor et al. [60].

3. Method

The problem formulation and proposed G2P2C algorithm are presented in Sections 3.1 and 3.2 respectively, followed by an introduction to the T1D simulator used in this study in Section 3.3. Section 3.4 highlights the benchmark algorithms used to compare the performance of G2P2C while Section 3.5 presents the experimental setup and evaluation metrics.

3.1. Problem Formulation

In RL, the environment is typically a Markov Decision Process (MDP) where all states are known. The environment for the closed-loop glucose control problem in T1D is the glucoregulatory system of the human body, which is a partially observable complex dynamical system [12]. The glucoregulatory system is observed through noisy glucose sensor measurements and controlled via an insulin infusion pump. A Partially Observable Markov Decision Process (POMDP) is defined as a tuple $\langle S^\star, S, O, A, P, R \rangle$, where $S^\star$ is the set of true environment states, S the set of states observed by an observation function O, and A the set of actions. The transition function P: $(s^\star, a) \rightarrow s'$ represents the system dynamics, where at each step, the RL agent is in a state $s^\star \in S^\star$, takes an action $a \in A$, and moves from $s^\star$ to the next state $s' \in S^\star$. The observation function $O : s^\star \rightarrow s$ maps the true environment states to the observed states $s \in S$, while the reward function R: $(s, a) \rightarrow r$ provides a reward $r \in \mathbb{R}$ for taking action a at an observed state s. The task of the RL agent is to achieve a defined goal by learning a mapping from states to actions which is called a policy $\big(\pi(a|s)\big)$. We define the glucose control problem as a continuing task, with the goal to maximize the average reward [63, 17]

$$R_{avg}(\pi) \doteq \lim_{h\rightarrow\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[r_t | s_0, a_{0:t-1} \sim \pi]. \tag{1}$$

The policy π induces a value function

$$v^\pi(s_t) \doteq \mathbb{E}[G_t | s_t \sim \pi], \tag{2}$$

which estimates the expected return G_t when starting from state s_t and following the policy π. The return G_t is estimated according to

$$G_t \doteq (r_{t+1} - \hat{R}_t^{avg}) + (r_{t+2} - \hat{R}_t^{avg}) + \cdots + (r_{t+n} - \hat{R}_t^{avg}) + \hat{v}^{\pi}(s_{t+n}) \tag{3}$$

where n denotes the number of transition steps, and $\hat{v}^{\pi}(s_{t+n})$ is the bootstrapped value function estimate at end state s_{t+n}. $\hat{R}_t^{avg}$ is an estimate of $R_{avg}(\pi)$ at time t. The advantage function

$$A^{\pi}(s_t, a_t) \doteq G_t - v^{\pi}(s_t) \tag{4}$$

is defined as the difference between the return and the value function estimate and measures whether a target action is better or worse than the average actions.

3.2. G2P2C (Glucose Control by Glucose Prediction and Planning)

The design and implementation details of the G2P2C algorithm are presented in this section. First, the observation space, action space, and reward function are formulated. Next, the neural network architecture design of G2P2C is presented. Finally, the G2P2C algorithm and the optimization procedure are introduced.

Observation Space. The observation function O: $s_t^{\star} \rightarrow (g_{t-k:t}, i_{t-k:t})$ was designed to map the true states $s_t^{\star}$ at time t to glucose sensor observation g_t and administered insulin i_t augmented by their past k historical measurements. Hence, the observed state space was defined as,

$$s_t = (g_{t-k:t}, i_{t-k:t}). \tag{5}$$

Action Space. We used a continuous action space $A \in [-1, 1]$. The choice of a continuous action space was made to provide additional flexibility to the RL agent to learn a control strategy as opposed to a handcrafted discrete action space. The insulin infusion rate of the insulin pump $I_{pump} \in [0, 5]$ U/min was defined as the control space. The action space was mapped to the control space using a non-linear translation function introduced in our previous work [64] according to Equation (6), where the parameter η was set to 4.0 and I_{max} to 5 U/min.

$$I_{pump} = I_{max} \cdot e^{\eta(a-1)}, a \in [-1, 1]. \tag{6}$$

Reward Function. The reward function was formulated based on the blood glucose Risk Index (RI), similar to [55, 61]. The RI is the sum of the Low Blood Glucose Risk Index (LBGI) and High Blood Glucose Index (HBGI) proposed by [65]. We additionally introduced a penalty for hypoglycemia ($g \leq 39$ mg/dL) and normalised the RI to [0, -1] for the rest of the glucose range (Equation (7)).

$$R(s_t, a_t) = \begin{cases} -15 & \text{if } g_{t+1} \leq 39 \text{ mg/dL} \\ -1 \cdot RI(g_{t+1})_{Normalized} & \text{else} \end{cases}. \tag{7}$$

Architecture. G2P2C was implemented using two separate neural networks with similar architecture; the Actor-Network (Π_θ) and the Critic-Network (V_ϕ) parameterized by θ and ϕ respectively (Figure 2). The feature extractor modules E^Π and E^V each consisted of a single-layer Long Short-Term Memory (LSTM) network [66] with 16 hidden units, where the input was the observed state (s_n) and the output was the hidden state vector (h_n) of the LSTM. The Actor-Network (Π_θ) included the policy module π, which represented the policy function, while the Critic-Network (V_ϕ) included the value module v which represented the value function. The output of the policy module (π) was formulated as a normal distribution ($\mathcal{N}(\mu, \sigma)$) over the action space, where both μ and σ parameters were learned. The output of the value module (v) was trained to predict the expected return.

We introduced a novel glucose prediction module (M^Π and M^V) for each network. M^Π and M^V modules were trained to learn the glucose dynamics of the target T1D subject. We integrated h_n and a_n as inputs to the glucose prediction modules, where the output was the one-step ahead glucose sensor measurement (g_{n+1}) represented as a normal distribution. This design was expected to further facilitate the learning of a dynamical system state representation ($s_n^\star$) at the hidden state (h_n) of the LSTM networks. This was because the hidden states (h_n) of LSTM networks are capable of learning a representation of the state space ($s_n^\star$) of a dynamical system [67]. The glucose prediction modules were implemented using similar architectures and trained for similar tasks in the two networks to facilitate feature distillation between the networks. This idea was inspired by [38]. The π and v modules consisted of 3 dense layers with 32 units each, while M^Π, and M^V modules consisted of 1 dense layer with 16 units. The networks were implemented using PyTorch [68] and optimized using the Adam optimizer [69] (see Appendices A for the resulting hyperparameters).

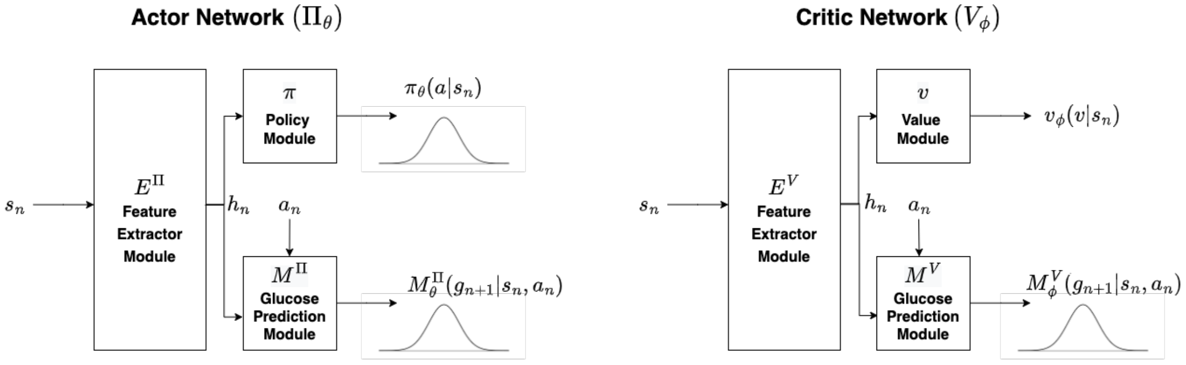

Figure 2: Schematic diagram of actor and critic networks.

Algorithm. G2P2C was based on the PPO algorithm enhanced by two novel optimization phases, namely, model learning and planning. PPO was selected as the baseline RL algorithm because of its demonstrated efficiency in safety-critical applications, where excessive changes in the control strategy could lead to unexpected behaviour. G2P2C alternated between sampling and optimization phases for multiple iterations. During the sampling phase, the current policy was used by w parallel agents and simulations were rolled out for n time steps. The resulting trajectory $(s_1, a_1, r_1, s_2, a_2 \cdots)$ information was stored in a data buffer (D). Once the sampling procedure was complete, the optimization procedure commenced, consisting of three sequential update phases. The first phase used the standard policy and value update of PPO [22], the second phase was the model-learning update, and the third phase was the planning update.

3.2.1. PPO Phase

During the first phase, the standard PPO optimization update was carried out, where the optimization objective of the policy module was to maximize the objective function $L^\pi(\theta)$ defined in Equation (8) where $\pi_{\theta_{old}}$ is the policy prior to the update. Excessive changes between the new and old policies were constrained by clipping the probability ratios of the policies to the interval $[1-\epsilon, 1+\epsilon]$. $H(\pi(\cdot|s_t))$ represented the entropy term used in the optimization to facilitate exploration where β_s was a hyperparameter. The return G_t defined in Equation (3) was used to calculate the advantage function estimate $\hat{A}_t$ and value function targets $\hat{v}_t^{target}$. The value module objective was to minimize the objective function $L^{v}(\phi)$ defined in Equation (9).

$$L^{\pi}(\theta) = \hat{\mathbb{E}}_t\left[min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t, clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right) + \beta_s H\left(\pi(\cdot|s_t)\right)\right]. \tag{8}$$

$$L^{v}(\phi) = \hat{\mathbb{E}}_t\left[\frac{1}{2}\left(v_\phi(s_t) - \hat{v}_t^{target}\right)^2\right]. \tag{9}$$

3.2.2. Model Learning Phase

The model learning phase succeeded the PPO phase. We introduced an auxiliary learning task to learn a model of the glucose dynamics of the target subject, at each of the two networks. The learned glucose dynamics model in the actor network was later used in the planning phase. The two modules M^{Π} and M^{V} were integrated to the actor and critic networks respectively and trained to learn the one-step ahead glucose prediction for the target subject. We designed a replay buffer (B) which stored the latest trajectories experienced by the algorithm as triplets of s_t, a_t, and g_{t+1}. The model-learning update commenced, once B was filled and was based on maximum likelihood estimation, where the objective was to minimize $L^{M^{\Pi}}(\theta)$ and $L^{M^{V}}(\phi)$ defined in Equation (10), (11). The Kullback-Leiber divergence (d_{KL}) and Mean Squared Error (MSE) penalties applied in $L^{M^{\Pi}}(\theta)$ and $L^{M^{V}}(\phi)$ respectively aimed to minimize the divergence from the already learned policy $\pi_{\theta_{ppo}}$ and value function $v_{\phi_{ppo}}$ after the PPO update phase. Hyperparameters β_1 and β_2 were introduced to regularize the respective penalties.

$$L^{M^{\Pi}}(\theta) = \hat{\mathbb{E}}_t\left[-log\left(M_\theta^{\Pi}(g_{t+1}|s_t, a_t)\right) + \beta_1 d_{KL}\left[\pi_{\theta_{ppo}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\right]\right]. \tag{10}$$

$$L^{M^{V}}(\phi) = \hat{\mathbb{E}}_t\left[-log\left(M_\phi^{V}(g_{t+1}|s_t, a_t)\right) + \beta_2\frac{1}{2}\left(v_{\phi_{ppo}}(s_t) - \hat{v}_\phi(s_t)\right)^2\right]. \tag{11}$$

3.2.3. Planning Phase

Following the model-learning phase, the third and final update was performed during the planning phase. We introduced a plan-space planning approach to improve the learned policy by integrating a short-term optimization objective. The planning phase only used the Π_θ network since it focused on fine-tuning the learned policy module. Once M^{Π} achieved a prediction accuracy of Root Mean Squared Error (RMSE) $< e_{target}$ (15mg/dL), the planning phase commenced. M^{Π} was used to carry out m number of short-horizon ($n_{plan} = 6$ simulation steps (30 minutes)) Monte Carlo rollouts (τ) for each state stored in the buffer D. For each state, the rollout with the best *simulated-return* (τ^*) was identified according to Equation (12). The planning phase fine-tuned the policy toward the best action (a_t^*) associated with the target state (s_t). The planning objective was to minimize $L^{plan}(\theta)$ which was designed based on maximum likelihood estimation (Equation (13)). During the planning phase, the Π_θ network weights associated with E^{Π} and M^{Π} modules were kept fixed, and only the weights associated with the π module were updated. This was done to ensure that the planning phase reflected a fixed M^{Π}. The steps of the G2P2C algorithm are summarized in Algorithm 1.

$$\tau^* = \arg\max_{\tau}\left[(\sum_{q=1}^{n_{plan}} R_q) + v(s_{n_{plan}})\right]. \tag{12}$$

$$L^{plan}(\theta) = \hat{\mathbb{E}}_t\left[-log\left(\pi_\theta(a_t^*|s_t)\right)\right]. \tag{13}$$

Algorithm 1 G2P2C

```
Initialize an empty auxiliary buffer ($B$) of the size $N_B$.
Initialize the average reward estimate ($\hat{R}^{avg} = 0$, $N_{total} = 0$).
Initialize Actor-Network ($\Pi$) and Critic-Network ($V$) weights ($\theta, \phi$).
for $iteration = 1, 2, 3, \cdots$ do
    Initialize an empty data buffer ($D$) of the size $N_D$.
    Perform $n$-step rollouts for $w$ parallel workers under current policy $\pi_{\theta_{old}}$ and store $(s_n, a_n, r_n)$ transitions $\rightarrow D$.
    Store $(s_n, a_n, g_{n+1})$ transitions $\rightarrow B$ .
    Compute the advantages $\hat{A}_t$ & value function targets $\hat{v}_t^{target}$.
    Update the average reward estimate ($\bar{R}_D = \frac{\sum r \in D}{N_D}$):
        $N_{total} \leftarrow N_{total} + N_D$,
        $\hat{R}^{avg} \leftarrow \hat{R}^{avg} + \frac{N_D}{N_{total}}(\bar{R}_D - \hat{R}^{avg})$.

    (1) PPO Phase:
    for $epoch = 1, 2, 3, \cdots, E_\pi$ do
        optimize $L^\Pi$ wrt $\theta$, on all data in $D$:
            $\theta \leftarrow \theta + \alpha_2 \cdot \nabla_\theta L^\Pi(\theta)$.
        Early stop:
            $d_{KL}[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)] > d_{target}$.
    end for
    for $epoch = 1, 2, 3, \cdots, E_V$ do
        optimize $L^v$ wrt $\phi$, on all data in $D$:
            $\phi \leftarrow \phi - \alpha_3 \cdot \nabla_\phi L^V(\theta)$.
    end for

    (2) Model Learning Phase:
    if $B$ is filled then
        for $epoch = 1, 2, 3, \cdots, E_M$ do
            optimize $L^{M^\Pi}$ wrt $\theta$, on all data in $B$:
                $\theta \leftarrow \theta - \alpha_4 \cdot \nabla_\theta L^{M^\Pi}(\theta)$.
            optimize $L^{M^V}$ wrt $\phi$, on all data in $B$:
                $\phi \leftarrow \phi - \alpha_5 \cdot \nabla_\phi L^{M^V}(\phi)$.
        end for
    end if

    (3) Planning Phase:
    if $M^\Pi_{pred-error} < e_{target}$ then
        Fix parameters related to $E^\Pi$, $M^\Pi$ modules of the actor network ($\Pi$).
        for $epoch = 1, 2, 3, \cdots, E_{plan}$ do
            optimize $L^{plan}$ wrt $\theta$, on all data in $D$:
                $\theta \leftarrow \theta - \alpha_5 \cdot \nabla_\theta L^{plan}(\theta)$.
        end for
    end if
    $\theta_{old} \leftarrow \theta$.
    $\phi_{old} \leftarrow \phi$.
    if $N_{total} > I_{total}$ then
        Stop.
    end if
end for
```

3.3. T1D Simulator

In this study, following established best practices in the development of control algorithms for glucose regulation,[1] we used the UVA/PADOVA T1D simulator [12]—currently the only FDA-approved T1D simulator. For reproducibility and to leverage the existing PyTorch [68] machine learning framework for the RL algorithm development, we used an open-source Python implementation of this simulator [51], used in previous work [55, 61, 70]. The simulator comprised a cohort of 30 *in-silico* subjects of three age categories (adults, adolescents, and children). The cohorts represented the patient variability found in a real T1D population, making meaningful statistical results available for the evaluation of our proposed approach. The simulator also included models of commercially available insulin pumps and glucose sensors and allowed for the definition of different meal protocols for simulations.

3.4. Benchmark Algorithms

We benchmarked G2P2C against a BB clinical insulin treatment strategy with meal announcement and a PPO algorithm without meal announcement (Table 1). For BB, a basal insulin infusion rate is used to provide background insulin requirements while a meal insulin bolus dose is used to counter meals and a correction insulin bolus dose to counter high blood glucose levels. BB treatment was used as the gold standard benchmark for evaluation due to its widespread recognition among clinicians [71]. This treatment strategy is based on patient-specific characteristics and requires prior knowledge about the CHO content of future meals (typically 20 minutes in advance).

We replicated two versions of the BB treatment based on the previous work of [61]; the Basal-Bolus Ideal (BBI) case, where the CHO of announced meals was provided accurately, and the realistic case, which considered the human inaccuracy in CHO estimation named Basal Bolus Human Error (BBHE). The CHO counting error was calculated based on a mathematical model developed to conduct *in-silico* trials [72]. The model used the real CHO content of the meal and meal type to simulate the decision of the patient with T1D.

For the replication of the two BB methods, we used the patient-specific characteristics provided by the T1D simulator and parameters proposed in previous work [61]. The final insulin delivered over time I_t is presented in Equation (14). Here c_t represents the meal CHO estimate, g_t the current glucose value, and g_{target} (140 mg/dL) the target glucose for correction boli. The Total Daily Insulin (TDI) provided by the simulator for each subject was used to personalise the Basal Rate $\left(\mathrm{BR} = 0.48 \cdot \mathrm{TDI\ U/day}\right)$, Carbohydrate Insulin Ratio $\left(\mathrm{CIR} = \frac{500}{\mathrm{TDI}}\ \mathrm{g/U}\right)$, and Insulin Sensitivity Factor $\left(\mathrm{ISF} = \frac{1800}{\mathrm{TDI}}\ \mathrm{mg/dL\ per\ U}\right)$ [73]. The meal bolus was calculated as $\frac{c_t}{\mathrm{CIR}}$ U, and delivered 20 minutes before a meal.

The correction insulin bolus was calculated as $\frac{(g_t - g_{target})}{\mathrm{ISF}}$ U, and was delivered when the glucose level increased above the correction threshold of 150 mg/dL. The correction insulin dose was replicated according to [61] and was only applied during meal events ($c_t > 0$). The application of the correction bolus had a further condition, where it was only applied when no other meals were present for a past 3 hour duration. This ensured that each meal was only corrected for once. The *cool* parameter was set to match this criteria, where *cool* was set to 1 if no other meals were present for the 3 hour duration and to 0 otherwise.

$$I_t = \mathrm{BR} + (c_t > 0) \cdot \left(\frac{c_t}{\mathrm{CIR}} + cool \cdot \frac{g_t - g_{target}}{\mathrm{ISF}} \right). \tag{14}$$

The PPO algorithm was implemented using the same hyperparameters used in G2P2C and a comparable neural network architecture. The only difference between the network architectures was the additional M^{Π} and M^{V} modules present in G2P2C. We used the results presented in previous RL-based glucose control algorithms research for completeness in comparisons and provide a discussion in Section 5.

[1] Simulators function as a replacement for animal studies conducted before clinical evaluation in humans.

Table 1
Summary of glucose control algorithms used in this study. Acronyms: BBI: Basal Bolus Ideal, BBHE: Basal Bolus Human Error, CHO: Carbohydrate, G2P2C: Glucose Control by Glucose Prediction and Planning, PPO: Proximal Policy Optimization, RL: Reinforcement Learning, TDI: Total Daily Insulin.

Algorithm	Meal announcement & CHO estimation required	Characteristics
BBI	✓	Basal-Bolus insulin infusion strategy based on personalised TDI. Fixed basal insulin delivery rate (0.48 x TDI U/day). Meal insulin bolus (based on CIR of $\frac{500}{\text{TDI}}$ g/U) - 20 minutes in advance based on accurate CHO information. Correction insulin bolus (based on ISF of $\frac{1800}{\text{TDI}}$ mg/dL per U and target blood glucose of 140 mg/dL) applied automatically when $g_t > 150mg/dL$.
BBHE	✓	Same as for BBI, but including human error in CHO estimation of meals.
PPO	✗	A state-of-the-art RL algorithm. Implemented as two neural networks (consists $E^{\Pi}, E^{V}, \pi, \upsilon$ modules - Section 3.2).
G2P2C	✗	Proposed RL-based algorithm. Designed based on PPO. Novel model-learning (additional M^{Π}, M^{V} modules) & planning phases introduced.

3.5. Experiment Setup & Evaluation Metrics

We trained G2P2C and PPO *in-silico* using the 10 adult and 10 adolescent cohorts of the Simglucose T1D simulator [51]. The Insulet pump and the GuardianRT glucose sensor provided by the simulator were used with a sampling time of 5 minutes [70]. We trained separate G2P2C and PPO algorithms for each *in-silico* subject. A challenging meal protocol where meals were uniformly randomized based on CHO content and time was designed for the training simulations (Table 2). The probability of occurrence of a meal was set to 0.95 to replicate missed meals by the user. The training was conducted for multiple iterations until $800,000$ interactions were completed. An *interaction* was defined as an insulin delivery action taken every 5 minutes, and total interactions were the summation of interactions by all parallel agents used in the algorithm as discussed in Section 3.2. For each subject the experiments were conducted for three different random seeds.

The evaluation of G2P2C, as well as of PPO and BB treatment methods was performed using the same two cohorts of the simulator. To support comparisons with other works, the evaluation meal protocol used in [62] was chosen for all algorithms. The evaluation meal protocol spanned 24 hours starting at 00:00hrs fixed with three meals: 40g of CHO for breakfast at 8:00hrs, 80g of CHO for lunch at 13:00hrs, and 60g of CHO for dinner at 20:00hrs. We restricted the initial blood glucose level between 110–130 mg/dL for the evaluation simulations.

We used the evaluation protocol to analyze the performance of G2P2C and PPO algorithms during training and to compare all candidate algorithms upon the conclusion of training. The training of G2P2C and PPO alternated between sampling and optimization phases conducted for multiple iterations. After the conclusion of each training iteration, 60 evaluation simulations (i.e., 3 random seeds x 20 simulations / iteration) were conducted for each subject. These simulations provided insights on the training characteristics of G2P2C and PPO algorithms. Once training was fully complete (at $800,000$ interactions), the RL algorithms were evaluated using $1,500$ evaluation simulations (i.e., 3 random seeds x 500 simulations / subject) conducted for each subject. We also conducted $1,500$ evaluation simulations for each subject under the BBI and BBHE approaches for comparison (Table 1).

Table 2
Training Meal Protocol.

Meal Type	Time [*hours*]	Carbohydrate (CHO) content [*g*]
Breakfast	07:00–09:00	30–60
Lunch	12:00–14:00	70–100
Dinner	19:00–21:00	50–110

Both clinical metrics and RL-based metrics were used for the evaluation. The clinical metrics included the standard T1D metrics of glucose risk indices (RI, HGBI, LGBI) [65] and the percentage of time spent in different glucose regions. The clinical objective was to minimize all risk indices, improve the time spent in the normoglycemic range (also called the Time in Range (TIR)), and minimize the time spent in other glucose regions (severe hypoglycemia (<50 mg/dL), hypoglycemia (50–70 mg/dL), hyperglycemia (180–300 mg/dL), and severe hyperglycemia (>300 mg/dL)). Additionally, we defined ***catastrophic failures*** as those simulations that recorded glucose levels outside the detectable range (39–600 mg/dL) of the glucose sensor. Such simulations were terminated, and a Failure Rate (FR) was calculated as the number of catastrophic failures over the total evaluation simulations.

G2P2C and PPO algorithms were compared using RL-based reward metrics. The total reward achieved by an RL algorithm in an evaluation simulation (R^{eval}), as a percentage of the maximum achievable reward ($R^{\star}$), was defined as the Percentage Reward ($PR = (\frac{R^{eval}}{R^{\star}}) * 100$). The maximum achievable reward represented the total reward achievable by following a perfect glucose control strategy based on the defined RL goal. The RL objective was to maximise the PR metric. We assumed the $800,000$ training interactions were sufficient for the policies to converge. The instability of the converged policies were compared using a modified version of the post-convergence instability (PCI) metric proposed in [74]. We used the evaluation simulations of both PPO and G2P2C ($1,500$ each) to calculate the 95% confidence interval ($[R^{eval}_{lower}, R^{eval}_{upper}]$) for R^{eval}, across the algorithms for each subject. PCI was calculated as the percentage of evaluation simulations where R^{eval} is below R^{eval}_{lower}. Hence, $PCI(\pi) = \big(\frac{\sum_{i=1}^{1500} \mathbf{1}(R^{eval}_{i} < R^{eval}_{lower})}{1500}\big) * 100$, where $\mathbf{1}(.)$ was an indicator function. The objective was to minimize PCI, which reflects better stability.

A statistical analysis of the identified clinical and RL-based metrics was conducted. A Shapiro-Wilk Test [75] was performed to check the normality, Mann-Whitney U Test [76] was conducted to evaluate significance, and effect size calculated using the Pearson product-moment correlation coefficient (r) [77]. The statistical analysis was conducted using the IBM SPSS Statistics Software (Version-28.0.1.1).

4. Results

4.1. Analysis of RL-based Metrics

G2P2C achieved a statistically significant ($P < 0.05$) PR improvement compared to PPO for 18 of the 20 subjects (Table 3). The performance improvement was non-significant for Adolescent0 and Adolescent4. Adolescent0 was the easiest to control under both RL-based methods (97.50%, 97.99%) leaving no margin for improvement for G2P2C. Adolescent6 (72.89%, 80.55%) was the hardest to control for both RL algorithms. G2P2C gave the largest improvement in the PR performance for Adolescent6, where it achieved a PR of 80.55% compared to 72.89% in PPO. Figure 3 summarizes the PCI metric for the target subjects. PPO showed higher instability compared to G2P2C in all subjects except for Adult9 and Adolescent4. The failure in G2P2C to achieve significant PR improvements in Adolescent4 could be related to the observed instability compared to PPO. But this should be further investigated. For some subjects (Adult0, Adult4, Adolescent1, Adolescent3, Adolescent6), G2P2C showed a very large improvement (> 25%) in PCI.

The standard deviation of the PR of the evaluation simulations were reduced in 18 out of the 20 subjects (except for Adult5 and Adult6) (Table 3); this behaviour is beneficial for the glucose control task to reduce the uncertainty. The reduction in the standard deviation was also visible through a qualitative analysis of the reward curves. The reward curves present the learning behaviour of RL algorithms during training (Figures 4 and 5). The reduction in the standard deviation was attributable to the effect of the planning phase proposed in G2P2C. Initially during training the reward curves had a similar standard deviation for both PPO and G2P2C (Figures 4 and 5). However, the planning phase of G2P2C was automatically initiated at approximately $200,000$ learning interactions once the learned glucose prediction module (M^{Π}) achieved the predefined accuracy threshold as described in Section 3.2. By analyzing the reward curves it can be observed that the standard deviation began to reduce in G2P2C compared to PPO once the planning was initiated. This reduction in the standard deviation was more prominent in some subjects (e.g., Adult0, Adolescent0) compared to others (e.g., Adult3, Adolescent6).

Table 3
Comparison of Percentage Reward (PR) for all subjects based on evaluation simulations. The significance level, $P = 0.05$, The effect size, $r > 0.1$: small effect, $0.3 < r < 0.5$: moderate effect, $r > 0.5$: large effect. Acronyms: G2P2C: Glucose Control by Glucose Prediction and Planning, PPO: Proximal Policy Optimization.

Subject	PPO	G2P2C	Significance (PPO - G2P2C)
Adult0*	94.00±4.42	95.50±1.02	$P < .001$, $r = 0.34$
Adult1*	95.69±3.39	96.31±0.99	$P < .001$, $r = 0.14$
Adult2*	87.33±14.822	91.92±6.53	$P < .001$, $r = 0.44$
Adult3*	88.71±12.13	90.64±9.86	$P < .001$, $r = 0.21$
Adult4*	92.40±9.21	94.73±5.03	$P < .001$, $r = 0.51$
Adult5*	88.17±9.49	88.19±14.80	$P < .001$, $r = 0.30$
Adult6*	92.07±7.51	92.32±9.91	$P < .001$, $r = 0.22$
Adult7*	94.77±3.39	95.57±1.77	$P < .001$, $r = 0.22$
Adult8*	89.82±6.26	91.18±1.41	$P < .001$, $r = 0.19$
Adult9*	92.46±5.85	92.85±1.08	$P < .001$, $r = 0.06$
Adolescent0	97.50±5.14	97.99±0.52	$P = .686$, $r = 0.01$
Adolescent1*	79.58±6.67	81.63±5.57	$P < .001$, $r = 0.30$
Adolescent2*	89.68±5.14	90.01±3.44	$P = .01$, $r = 0.05$
Adolescent3*	87.44±7.41	90.08±4.42	$P < .001$, $r = 0.42$
Adolescent4	87.98±7.48	88.17±3.77	$P = .831$, $r = 0.00$
Adolescent5*	90.09±11.12	91.10±9.34	$P < .001$, $r = 0.21$
Adolescent6*	72.89±12.57	80.55±7.59	$P < .001$, $r = 0.44$
Adolescent7*	80.52±9.11	82.67±3.15	$P < .001$, $r = 0.17$
Adolescent8*	92.99±4.33	93.10±2.06	$P < .001$, $r = 0.09$
Adolescent9*	86.60±9.42	87.81±8.55	$P < .001$, $r = 0.18$

* Statistical significance ($P < 0.05$).

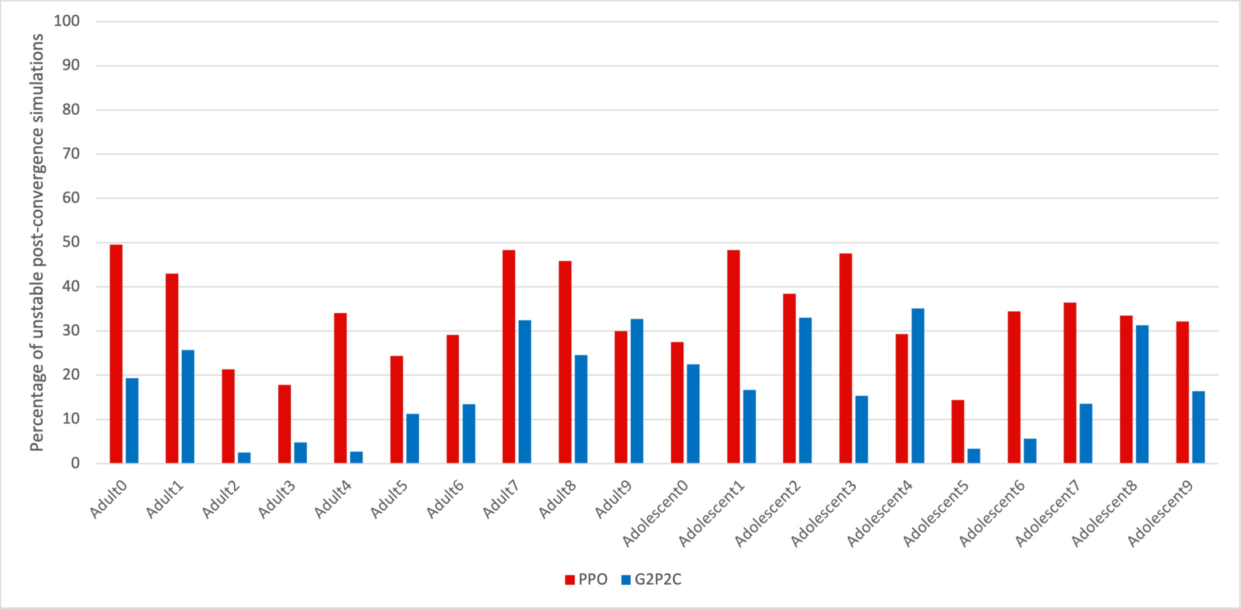

Figure 3: Post-convergence instability (PCI) comparison between PPO and G2P2C algorithms calculated based on evaluation simulations. Acronyms: G2P2C: Glucose Control by Glucose Prediction and Planning, PPO: Proximal Policy Optimization.

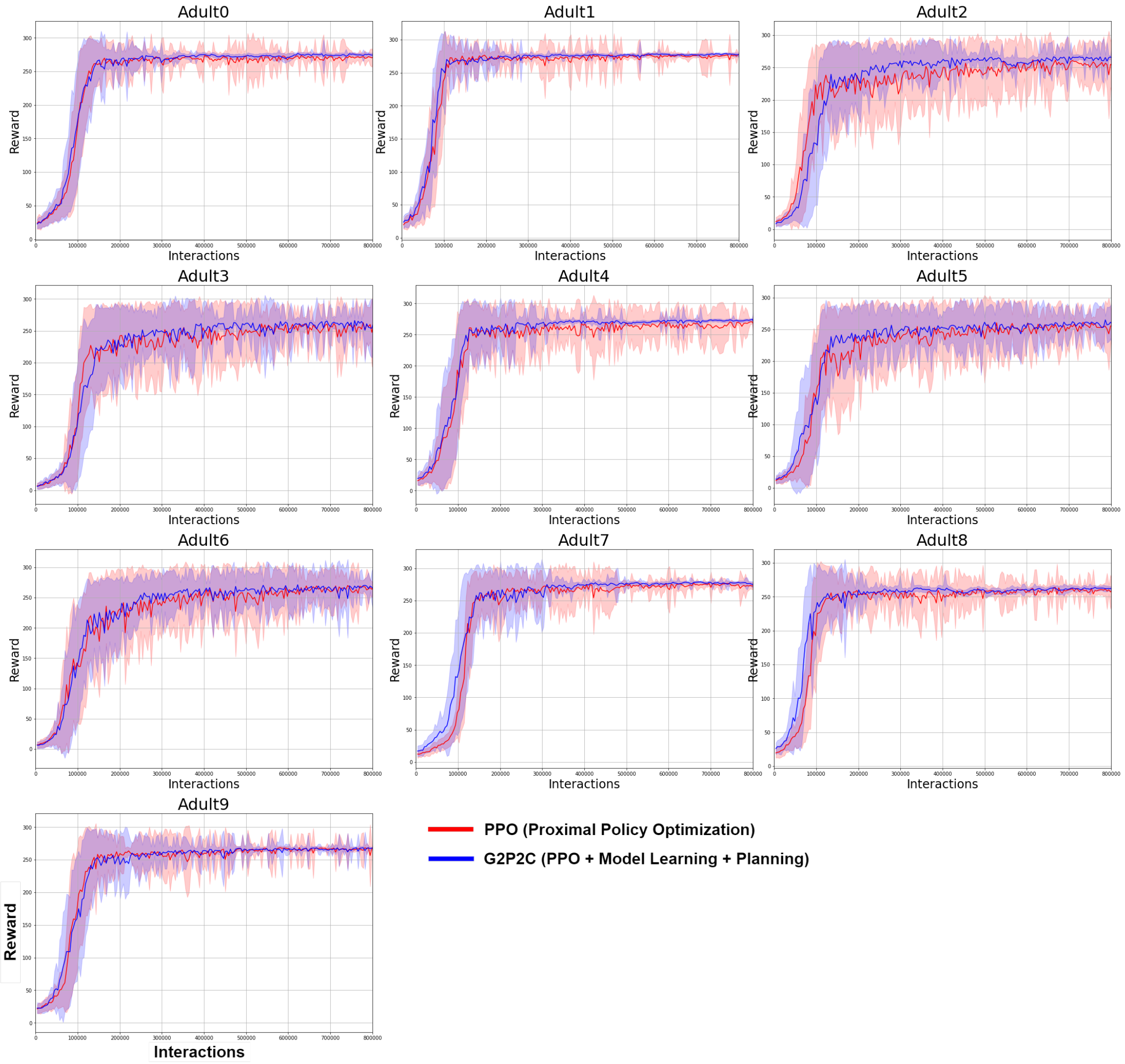

Figure 4: Comparison of PPO, G2P2C algorithms during training on the adult cohort. The mean and standard deviation (shaded area) of the total cumulative reward for evaluation simulations (3 random seeds x 20 evaluation scenarios / iteration) achieved is presented against $800,000$ learning interactions. Acronyms: G2P2C: Glucose Control by Glucose Prediction and Planning, PPO: Proximal Policy Optimization.

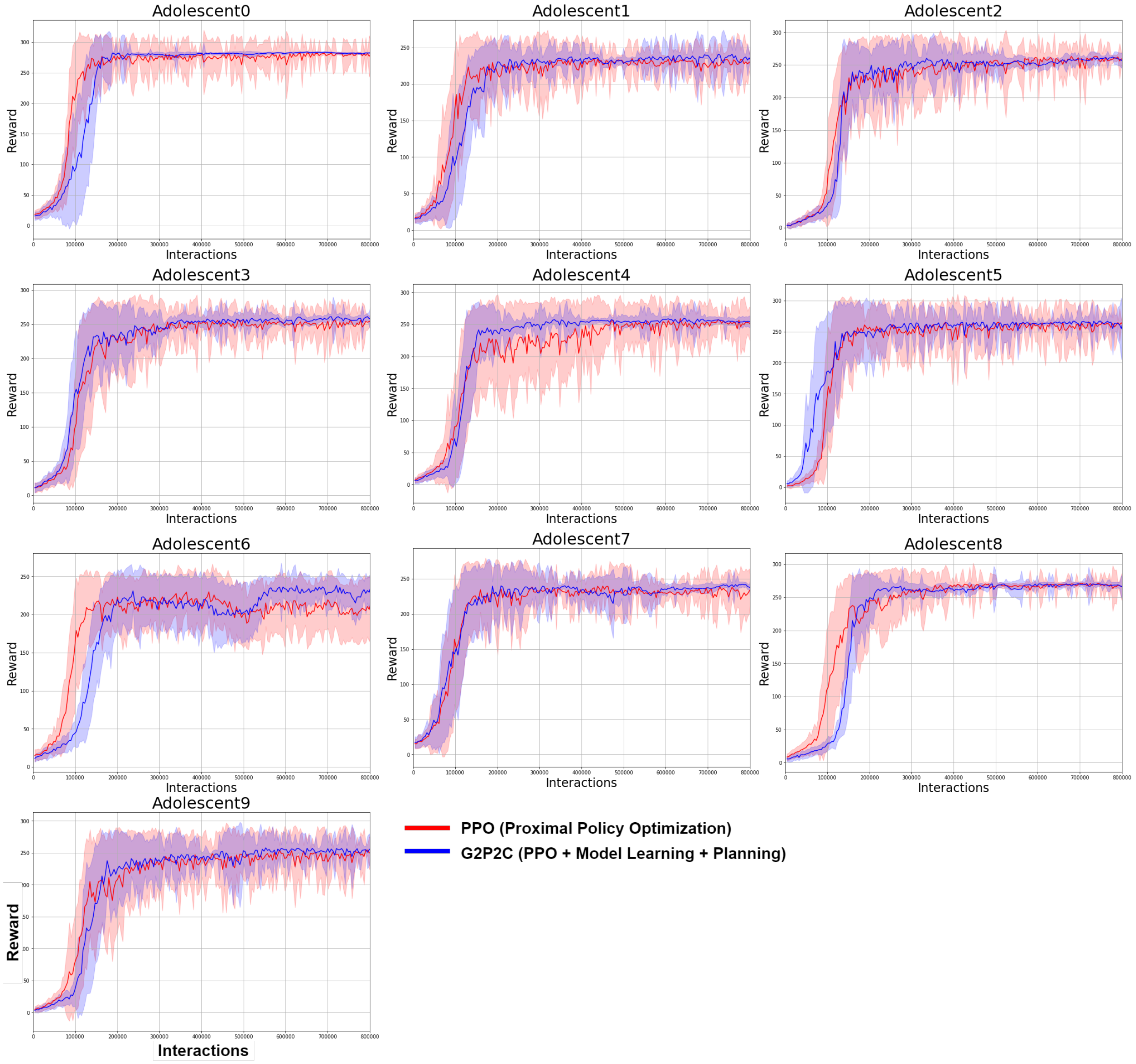

Figure 5: Comparison of PPO, G2P2C algorithms during training on the adolescent cohort. The mean and standard deviation (shaded area) of the total cumulative reward for evaluation simulations (3 random seeds x 20 evaluation scenarios / iteration) achieved is presented against $800,000$ learning interactions. Acronyms: G2P2C: Glucose Control by Glucose Prediction and Planning, PPO: Proximal Policy Optimization.

4.2. Analysis based on Clinical Metrics

For the adult cohort, G2P2C achieved a mean TIR of 72.69%, which was higher than BBI (70.83%) and BBHE (69.79%). The improvement was statistically significant ($P < 0.05$) compared to BBI ($r = 0.09$) and BBHE ($r = 0.14$). The clinical performance of the candidate algorithms based on T1D-related criteria are summarized in Table 4. The improvement of G2P2C compared to BB methods were mainly attributable to the reduction of the time in the hyperglycemic range without an increase in the hypoglycemic zones. This is especially important considering the fact that G2P2C included no prior meal announcement. All algorithms (RL-based and BB treatment) presented comparable hyper- and hypoglycemic risk profiles, as demonstrated by the HBGI and LBGI indices. In comparison to PPO, G2P2C achieved an improved performance in all clinical metrics without any compromises while also reducing the standard deviation. Finally, the RL-based algorithms showed a higher failure rate compared to the standard treatment methods. However, G2P2C managed to reduce the failure rate to 1.62% compared to 2.79% in PPO.

For the adolescent cohort, G2P2C achieved a mean TIR of 64.33% compared to 71.43% and 70.23% of BBI and BBHE. The difference was mainly observable in the time spent in the severe hyperglycemia zone, while the time in the hypoglycemic zones was not heavily affected. This was reflected in the hyper- and hypoglycemic indices, with the RL-based algorithms maintaining a moderate-risk profile in terms of HGBI (10.0≤HGBI≤15 [52]) and a low-risk profile in terms of LGBI(1.1≤LGBI≤2.5 [52]). The RL-based algorithms presented higher failure rates compared to the BB treatment methods. In this respect, the contribution of G2P2C was outstanding in reducing the failure rates compared to PPO with 1.48% and 4.93% failure rate respectively.

5. Discussion

In this study, we proposed a novel algorithm, named G2P2C, for glucose control in T1D which (1) provides fully-automated insulin infusion estimates, including both basal and bolus, and (2) does not require meal announcement and CHO estimation. We introduced two novel phases to the state-of-the-art RL algorithm PPO, namely, model learning and planning, to address the challenges of complex dynamics; partial observability; high inter- and intra-population variability; safety; and unknown delays and disturbances associated with glucose control. We evaluated the clinical performance of G2P2C based on a number of T1D-related metrics and compared against standard treatment methods commonly used in clinical practice. G2P2C achieved a higher TIR of 72.69% for the adult cohort compared to BBI (71.02%) and BBHE (69.78%) while eliminating the need for CHO estimation and meal announcement, as well as corrective boli, upon which both BB methods rely. For the case of the adolescent cohort, which is much more challenging than the adult cohort, the TIR achieved by G2P2C was lower than the BB methods without and with human error (64% compared to 71.43% and 70.23% respectively). However, the performance comparison should account for G2P2C receiving no prior information related to upcoming meals.

In our previous work, we applied PPO for the glucose control problem focusing on a long horizon optimization objective [70]. The PPO algorithm was selected due to its demonstrated efficient and stable learning achieved through a clipped optimization objective which is beneficial for safety-critical applications [22]. A long horizon was selected to reflect the clinical objective of T1D to improve the TIR. However, we observed that optimizing for the long horizon was insufficient as it failed to improve glucose control in short horizons, leading to large unwanted glucose fluctuations in the short term, ultimately leading to catastrophic failures. Hence, we introduced a planning phase in the G2P2C algorithm to focus on the short horizon. The planning phase used the learned glucose dynamics model in the model learning phase, to simulate short-term trajectories and fine-tuned the learnt policy for the short horizon. Through this mechanism, G2P2C was able to improve the safety by reducing the catastrophic failures to 1.62% and 1.48% in the adult and adolescent cohorts respectively compared to 2.79% and 4.93% in PPO while also enhancing the clinical performance. It also improved the sample efficiency by re-using collected experiences and facilitating the simulation-based exploration of the glucose state space using the learned model. Hence, it should become valuable in continuous online learning in real-world applications where data is limited to tune the policy. The learned model in G2P2C is expected to be valuable in feature distillation between the actor and value networks and in designing safety modules for an APS where the glucose dynamics model can be used to predict potential high and low blood glucose events targeting safety.

Research on glucose control often benchmarks the performance of algorithms against standard clinical treatment approaches and guidelines, as presented above in this work. Comparisons with prior work are rare and complicated due to the different cohorts, protocols, and simulator versions used. Despite differences in their methodology and for the sake of a more thorough assessment, we have included a comparison of our proposed algorithm's performance with

Table 4
Comparison of clinical performance. The median, inter-quartile range followed by the mean (standard deviation) are presented. Acronyms: BBHE: Basal Bolus Human Error, BBI: Basal Bolus Ideal, G2P2C: Glucose Control by Glucose Prediction and Planning, HBGI: High Blood Glucose Index, LBGI: Low Blood Glucose Index, PPO: Proximal Policy Optimization, RI: Risk Index, TIR: Time in Range.

Method	Failure	Severe	Hypo.	Normo. (TIR)	Hyper.	Severe	RI	LBGI	HBGI
	(%)	Hypo.(%)	(%)	(%)	(%)	Hyper.(%)			
Adults									
BBI	0.39	0.03†	0.00	70.83	26.74	0.00	7.68	0.81	6.72
		0.00-0.00*	0.00-0.00	61.46-79.17	19.79-34.38	0.00-0.00	6.07-9.38	0.28-1.68	4.97-7.94
		0.03(0.23)‡	0.67(2.11)	71.02(11.29)	27.18(10.59)	1.10(3.33)	8.35(4.05)	1.33(1.60)	7.02(2.79)
BBHE	0.35	0.00	0.00	69.79	28.47	0.00	7.62	0.41	6.92
		0.00-0.00	0.00-0.00	60.42-78.47	21.18-35.42	0.00-0.00	5.78-8.69	0.11-0.93	5.11-8.11
		0.02(0.20)	0.42(1.50)	69.78(11.29)	28.66(10.81)	1.12(3.32)	8.00(3.74)	0.88(1.37)	7.11(2.67)
PPO	2.79	0.00	0.00	69.44	26.74	0.00	9.56	0.89	8.05
		0.00-0.00	0.00-1.04	62.15-76.04	21.18-32.64	0.00-4.86	6.89-12.04	0.28-2.17	5.86-10.10
		0.19(1.04)	1.31(3.14)	69.12(10.53)	26.72(9.27)	2.65(3.76)	9.79(3.66)	1.64(2.11)	8.14(3.05)
G2P2C	1.62	0.00	0.00	72.57	23.96	0.00	9.00	1.04	7.42
		0.00-0.00	0.00-1.04	66.32-79.86	18.75-29.51	0.00-3.82	6.21-11.04	0.43-2.14	5.18-9.35
		0.13(0.89)	1.21(2.78)	72.69(9.53)	24.10(8.39)	1.88(2.74)	8.94(3.18)	1.58(1.74)	7.36(2.60)
Adolescent									
BBI	0.00	0.00	0.00	67.71	24.65	0.00	7.93	0.43	7.56
		0.00-0.00	0.00-0.00	63.54-76.39	18.75-32.64	0.00-9.72	5.21-14.10	0.09-1.40	5.04-12.26
		0.02(0.22)	0.45(1.30)	71.43(12.31)	24.62(11.73)	3.48(5.30)	9.26(4.83)	1.07(1.57)	8.19(3.78)
BBHE	0.00	0.00	0.00	66.67	25.69	0.00	8.06	0.21	7.94
		0.00-0.00	0.00-0.00	63.19-73.26	19.10-34.72	0.00-10.07	5.68-14.11	0.01-0.87	5.67-12.59
		0.00(0.01)	0.21(0.81)	70.23(12.52)	25.78(12.31)	3.78(5.55)	9.15(4.51)	0.69(1.11)	8.46(3.82)
PPO	4.93	0.00	0.00	60.42	24.65	8.68	14.66	1.21	12.75
		0.00-0.00	0.00-1.39	54.17-70.14	19.79-30.56	4.51-18.06	11.16-20.63	0.49-2.45	9.73-18.84
		0.16(0.94)	1.42(3.11)	63.72(13.95)	23.93(9.63)	10.77(8.59)	15.40(6.67)	1.82(1.99)	13.58(6.55)
G2P2C	1.48	0.00	0.00	60.76	24.65	7.64	14.29	1.17	12.24
		0.00-0.00	0.00-1.04	55.56-70.14	20.49-30.56	4.17-18.75	11.20-20.74	0.52-2.26	9.75-19.48
		0.09(0.64)	1.15(2.57)	64.33(13.18)	24.29(9.28)	10.14(7.83)	15.10(6.50)	1.65(1.64)	13.45(6.23)

† Median, * Inter-quartile range, ‡ Mean(Standard Deviation).

previous work in terms of TIR (Table 5). In our comparison we consider RL-based fully autonomous systems as well as the study [59], which proposed a RL-based hybrid method.

In order to make meaningful comparisons with previous work, it is important to first discuss the essential differences among the presented studies. One of the main differences relates to the the type of population used for the algorithmic assessment. Some previous studies have only considered handpicked individual subjects [55] or adult-only cohorts [62]. [61] has considered child, adult and adolescent cohorts, however, reports the overall median TIR values. [59] has used the adolescent and adult cohorts and present the mean TIR for each cohort. In our work, we have also used both adolescent and adult cohorts and assessed our algorithm's performance separately for each. This is particularly important as we have seen that the adolescent cohort is much harder to control than the adult cohort. Another important difference among studies is related to the level of complexity of the testing meal protocol. Some studies have used meal protocols with reasonably low CHO contents [55, 61] while others have focused on more challenging meal protocols with 180g of CHO daily [62, 59]. Similar to those, in this work we have used a high CHO meal scenario. Finally, all the reported studies have used the UVA/PADOVA simulator, which was originally designed based on the MATLAB framework [78]. However, previous studies have used different versions and implementations of the simulator. This is mainly due to the lack of support extended by existing simulators for the integration with the Python framework

[79] which is predominantly used for designing RL algorithms due to its favourable characteristics (e.g., the ability to simulate parallel environments and use existing machine learning frameworks). Hence, [59] and [62] have used independent customized platforms based on the UVA/PADOVA simulator while [61] has adopted an open-source version. In our work we used the open-source version to ensure the reproducibility of our work. [59] and [62] are the most comparable to this study due to the similar challenging meal protocol used. Compared to [59], which is a hybrid approach, G2P2C improved the performance in both the adult and adolescent cohorts. However, the performance on the adult cohort was less than [62]. The evaluation trials conducted were different in these studies, where [59] used a 10-day simulation for each subject without multiple repetitions, while [62] conducted seven random daily trials, as opposed to $1,500$ random daily trials performed in our study. The *catastrophic failure* rate was not presented in [59] and [62] studies, while [61] reported a FR of 0%. However, they defined failures as simulations where glucose levels are ≤ 5 mg/dL. In contrast, in our work we defined a much strict FR where glucose levels ≤ 40 mg/dL or ≥ 600 mg/dL were considered failures. The lack of appropriate benchmarks in this field of research is a hurdle towards meaningful comparisons.

Table 5

Comparison of percentage time in normoglycemia (TIR) with previous work based on RL algorithms. The TIR results are presented as mean±std for Lee et al. [62], Lim et al. [59], and our work. The median TIR is presented for Fox et al. [61]. Acronyms: CHO: Carbohydrate, G2P2C: Glucose Control by Glucose Prediction and Planning, MA: Meal Announcement, PPO: Proximal Policy Optimization, RL: Reinforcement Learning, SAC: Soft Actor Critic.

[a] Results presented as the median TIR for all cohorts. This is the only RL-based study with the Child cohort.

[b] Meal intake information required, scenario with multiple small meals is used (40g, 50g, 20g, 50g, 20g ±12.5%).

*Hybrid system (require at least meal announcement).

† Fully autonomous system.

Approach	Adults	Adolescents	Average daily CHO (g)	Simulator
Fox et al. [61][a]				
SAC (RL-MA)*	77.12		Low CHO Meals	Simglucose 2018
SAC (RL-Scratch)†	72.68		(values not provided)	(UVA/PADOVA 2008)
Lee et al. [62]				
PPO†	89.30±4.19	-	180	Custom platform based on UVA/PADOVA 2013
Lim et al. [59] [b]				
SAC*	65.93±17.29	62.20±19.99	180 (157.5-202.5)	Custom platform based on UVA/PADOVA 2013
Our Work				
PPO†	69.12±10.53	63.73±13.95	180	Simglucose 2018
G2P2C†	72.69±9.53	64.33±13.18		(UVA/PADOVA 2008)

The successful real-world application of the proposed G2P2C algorithm requires further research on the areas of safety, personalization, transferability of the *in-silico* learned strategy to real-life, and explainability of the control strategy. We have considered all these aspects in the design of G2P2C and reserve them for future work. Specifically, we aim to improve the *safety* of the algorithm by using the learned glucose dynamics model to design a safety module. The impact of the model error of the learned glucose dynamics model on the control performance was not analyzed in this study. We reserve it for future work along with the exploration of the effect of using glucose prediction for different time-horizons (e.g., 30, 60 minutes), in contrast to the one-step (5 minute) ahead predictions used in this study. An inter- and intra-population variability in the control performance was observed in this study. Designing a reward function, which reflects individual subject characteristics is expected to benefit in *personalizing* G2P2C to learn better control strategies, which we explore in future work. The common reward function used across the subjects in this study limits the algorithms capacity to learn a more personalised treatment strategy. The *transferability* requires sufficient real-world training, which could be infeasible and extremely dangerous to be conducted on-line. In future work, we explore the use of off-line patient data to fine-tune the glucose dynamics modules in G2P2C to learn personalized glucose dynamics of the target subject and safe-methods towards transferability. A fundamental limitation to research in the area of RL-based glucose control algorithms is the restrictions present in current T1D simulators. We aim to incorporate the latest version (2018) of the UVA/PADOVA simulator in our future experiments.

In our envisioned future work, we will prioritize on designing tools and methods to improve the *explainability* of G2P2C. As a first step, as part of this paper, we have provided an online demonstration tool (`http://capsml.com/`) of G2P2C for users to experiment with the algorithm and compare its performance against clinical treatment strategies for custom simulations.

6. Conclusion

In this research, we have proposed G2P2C, an RL-based APS, for the challenging glucose control problem in people with T1D. In G2P2C, we have introduced a model learning phase that is beneficial to capturing the glucose dynamics of the target T1D subject and a planning phase that optimizes for the short-term resulting in a control strategy that improves safety. G2P2C has improved the TIR performance of the adult cohort compared to BB-based clinical treatment strategies and improved the FR compared to PPO in both adult and adolescent cohorts. To facilitate the development of RL-based APS, we open-source the codebase of G2P2C (`https://github.com/chirathyh/G2P2C`) and provide an online demonstration tool for G2P2C (`http://capsml.com/`). This research is expected to be valuable for the T1D diabetes community through the exploration of solutions to reduce the cognitive burden and for the RL community through the development of new RL algorithms targeting real-world applications. The control performance and algorithmic characteristics of G2P2C show promise as a candidate algorithm for glucose control in APS.

7. Software and Data

We provide the source code and an online demonstration tool of G2P2C under the MIT license.

- Source code and experimental data: `https://github.com/chirathyh/G2P2C`.
- Online demonstration tool for G2P2C, where custom simulations can be performed: `http://capsml.com/`

8. Acknowledgement

This research was funded by and has been delivered in partnership with the Australian National University (ANU), School of Computing and the Our Health in Our Hands (OHIOH) grand challenge, a strategic initiative of the ANU, which aims to transform healthcare by developing new personalised health technologies and solutions in collaboration with patients, clinicians, and healthcare providers. This research was also supported by the National Computation Infrastructure (NCI Australia), an NCRIS enabled capability supported by the Australian Government. The authors wish to thank Dr David O'Neal, Dr Barbora Paldus, and Dr Dale Morrison from the Diabetes Technology Research Group, St Vincent's Hospital for their valuable insights towards this research.

A. Hyperparameters

Table 6
Hyperparameters. Acronyms: PPO: Proximal Policy Optimization.

Hyperparameter	Symbol	Value
Sample time		5 minutes
Glucose sensor		Guardian RT
Insulin pump		Insulet
Augmented state history	k	12 (1-hour)
Total number of Interactions	I_{total}	$800,000$
Batch size of policy, value, model-learning, and planning		$1,024$
Number of steps per rollout	$n_{rollout}$	256
Number of workers	w	16
Data buffer (D) size	N_D	$n_{rollout} \cdot w$
Auxiliary buffer (B) size	N_B	$25,000$
No. of policy epochs / value epochs / model-learning epochs	E_Π, E_V, E_M	5
No. of planning epochs	E_{plan}	1
Entropy Coefficient	β_s	0.001
Penalty Coefficient aux-policy / aux-value	β_1, β_2	0.01
Learning rate of policy / value / model-learning / planning	$\alpha_2, \alpha_3, \alpha_4, \alpha_5$	3×10^{-4}
PPO clip range	ϵ	0.1
Target Kullback-Leiber divergence (d_{KL}) threshold	d_{target}	0.01
Target glucose prediction error threshold	e_{target}	15mg/dL
Planning trajectories	m	50 (per state)
Planning horizon	n_{plan}	6 (30-minutes)

References

[1] L. A. DiMeglio, C. Evans-Molina, R. A. Oram, Type 1 diabetes, The Lancet 391 (2018) 2449–2462.

[2] M. D. Breton, B. P. Kovatchev, One year real-world use of the Control-IQ advanced hybrid closed-loop technology, Diabetes Technology & Therapeutics (2021).

[3] A. Saunders, L. H. Messer, G. P. Forlenza, Minimed 670g hybrid closed loop artificial pancreas system for the treatment of type 1 diabetes mellitus: Overview of its safety and efficacy, Expert Review of Medical Devices 16 (2019) 845–853.

[4] L. Leelarathna, P. Choudhary, E. G. Wilmot, A. Lumb, T. Street, P. Kar, S. M. Ng, Hybrid closed-loop therapy: Where are we in 2021?, Diabetes, Obesity and Metabolism 23 (2021) 655–660.

[5] G. M. Steil, Algorithms for a closed-loop artificial pancreas: The case for proportional-integral-derivative control, Journal of Diabetes Science and Technology 7 (2013) 1621–1631.

[6] B. W. Bequette, Algorithms for a closed-loop artificial pancreas: the case for model predictive control, Journal of Diabetes Science and Technology 7 (2013) 1632–1643.

[7] R. B. Shah, M. Patel, D. M. Maahs, V. N. Shah, Insulin delivery methods: Past, present and future, International Journal of Pharmaceutical Investigation 6 (2016) 1.

[8] D. Control, C. T. R. Group, The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus, New England Journal of Medicine 329 (1993) 977–986.

[9] D. Slattery, S. Amiel, P. Choudhary, Optimal prandial timing of bolus insulin in diabetes management: A review, Diabetic Medicine 35 (2018) 306–316.

[10] A. Brazeau, H. Mircescu, K. Desjardins, C. Leroux, I. Strychar, J. Ekoé, R. Rabasa-Lhoret, Carbohydrate counting accuracy and blood glucose variability in adults with type 1 diabetes, Diabetes Research and Clinical Practice 99 (2013) 19–23.

[11] C. Cobelli, E. Renard, B. Kovatchev, Artificial pancreas: past, present, future, Diabetes 60 (2011) 2672–2682.

[12] B. P. Kovatchev, M. Breton, C. Dalla Man, C. Cobelli, In silico preclinical trials: A proof of concept in closed-loop control of type 1 diabetes, Journal of Diabetes Science and Technology 3 (2009) 44–55.

[13] A. Cinar, Multivariable adaptive artificial pancreas system in type 1 diabetes, Current Diabetes Reports 17 (2017) 1–11.

[14] W. Villena Gonzales, A. T. Mobashsher, A. Abbosh, The progress of glucose monitoring — A review of invasive to minimally and non-invasive techniques, devices and sensors, Sensors 19 (2019) 800.

[15] J. Vliebergh, E. Lefever, C. Mathieu, Advances in newer basal and bolus insulins: Impact on type 1 diabetes, Current Opinion in Endocrinology, Diabetes and Obesity 28 (2021) 1–7.

[16] M. K. Bothe, L. Dickens, K. Reichel, A. Tellmann, B. Ellger, M. Westphal, A. A. Faisal, The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas, Expert Review of Medical Devices 10 (2013) 661–673.

[17] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

[18] E. Daskalaki, P. Diem, S. G. Mougiakakou, An actor–critic based controller for glucose regulation in type 1 diabetes, Computer Methods and Programs in Biomedicine 109 (2013) 116–125.

[19] A. Vajapey, Predicting optimal sedation control with reinforcement learning, Ph.D. thesis, Massachusetts Institute of Technology, 2019.

[20] B. K. Petersen, J. Yang, W. S. Grathwohl, C. Cockrell, C. Santiago, G. An, D. M. Faissol, Precision medicine as a control problem: Using simulation and deep reinforcement learning to discover adaptive, personalized multi-cytokine therapy for sepsis, arXiv preprint arXiv:1802.10440 (2018).

[21] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al., Mastering atari, go, chess and shogi by planning with a learned model, Nature 588 (2020) 604–609.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[23] G. Dulac-Arnold, D. Mankowitz, T. Hester, Challenges of real-world reinforcement learning, arXiv preprint arXiv:1904.12901 (2019).

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

[25] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al., Soft actor-critic algorithms and applications, arXiv preprint arXiv:1812.05905 (2018).

[26] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, A. Shah, Learning to drive in a day, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 8248–8254.

[27] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al., Solving Rubik's cube with a robot hand, arXiv preprint arXiv:1910.07113 (2019).

[28] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (1992) 229–256.

[29] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International Conference on Machine Learning, PMLR, 2015, pp. 1889–1897.

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

[31] A. Nagabandi, G. Kahn, R. S. Fearing, S. Levine, Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 7559–7566.

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).

[33] S. Bansal, R. Calandra, K. Chua, S. Levine, C. Tomlin, MBMF: Model-based priors for model-free reinforcement learning, arXiv preprint arXiv:1709.03153 (2017).

[34] C. Xiao, Y. Wu, C. Ma, D. Schuurmans, M. Müller, Learning to combat compounding-error in model-based reinforcement learning, arXiv preprint arXiv:1912.11206 (2019).

[35] R. S. Sutton, Dyna, an integrated architecture for learning, planning, and reacting, ACM Sigart Bulletin 2 (1991) 160–163.

[36] S. Gu, T. Lillicrap, I. Sutskever, S. Levine, Continuous deep q-learning with model-based acceleration, in: International Conference on Machine Learning, PMLR, 2016, pp. 2829–2838.

[37] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, et al., What matters for on-policy deep actor-critic methods? A large-scale study, in: International Conference on Learning Representations, 2020.

[38] K. W. Cobbe, J. Hilton, O. Klimov, J. Schulman, Phasic policy gradient, in: International Conference on Machine Learning, PMLR, 2021, pp. 2020–2027.

[39] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, P. Bachman, Data-efficient reinforcement learning with self-predictive representations, arXiv preprint arXiv:2007.05929 (2020).

[40] N. Hansen, R. Jangir, Y. Sun, G. Alenyà, P. Abbeel, A. A. Efros, L. Pinto, X. Wang, Self-supervised policy adaptation during deployment, arXiv preprint arXiv:2007.04309 (2020).

[41] M. Hessel, I. Danihelka, F. Viola, A. Guez, S. Schmitt, L. Sifre, T. Weber, D. Silver, H. Van Hasselt, Muesli: Combining improvements in policy optimization, in: International Conference on Machine Learning, PMLR, 2021, pp. 4214–4226.

[42] A. X. Lee, A. Nagabandi, P. Abbeel, S. Levine, Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model, Advances in Neural Information Processing Systems 33 (2020) 741–752.

[43] T. Anthony, Z. Tian, D. Barber, Thinking fast and slow with deep learning and tree search, Advances in Neural Information Processing Systems 30 (2017).

[44] H. Khorasgani, C. Zhang, C. Gupta, S. Serita, Long-term planning, short-term adjustments (2019).

[45] R. S. Sutton, Td models: Modeling the world at a mixture of time scales, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 531–539.

[46] W. Fedus, C. Gelada, Y. Bengio, M. G. Bellemare, H. Larochelle, Hyperbolic discounting and learning over multiple horizons, arXiv preprint arXiv:1902.06865 (2019).

[47] J. Romoff, P. Henderson, A. Touati, E. Brunskill, J. Pineau, Y. Ollivier, Separating value functions across time-scales, in: International Conference on Machine Learning, PMLR, 2019, pp. 5468–5477.

[48] E. Daskalaki, P. Diem, S. G. Mougiakakou, Personalized tuning of a reinforcement learning control algorithm for glucose regulation, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2013, pp. 3487–3490.

[49] Q. Sun, M. V. Jankovic, J. Budzinski, B. Moore, P. Diem, C. Stettler, S. G. Mougiakakou, A dual mode adaptive basal-bolus advisor based on reinforcement learning, IEEE Journal of Biomedical and Health Informatics 23 (2018) 2633–2641.

[50] Q. Sun, M. V. Jankovic, S. G. Mougiakakou, Reinforcement learning-based adaptive insulin advisor for individuals with type 1 diabetes patients under multiple daily injections therapy, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2019, pp. 3609–3612.

[51] J. Xie, Simglucose v0. 2.1 (2018), Available at: https://github.com/jxx123/simglucose (2018).

[52] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, C. Cobelli, The UVA/PADOVA type 1 diabetes simulator: New features, Journal of Diabetes Science and Technology 8 (2014) 26–34.

[53] T. Zhu, K. Li, P. Georgiou, A dual-hormone closed-loop delivery system for type 1 diabetes using deep reinforcement learning, arXiv preprint arXiv:1910.04059 (2019).

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

[55] I. Fox, J. Wiens, Reinforcement learning for blood glucose control: Challenges and opportunities, Available at: https://openreview.net/forum?id=ByexVzSAs4 (2019).

[56] J. N. Myhre, I. K. Launonen, S. Wei, F. Godtliebsen, Controlling blood glucose levels in patients with type 1 diabetes using fitted q-iterations and functional features, in: 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2018, pp. 1–6.

[57] P. D. Ngo, S. Wei, A. Holubová, J. Muzik, F. Godtliebsen, Control of blood glucose for type-1 diabetes by using reinforcement learning with feedforward algorithm, Computational and Mathematical Methods in Medicine 2018 (2018).

[58] P. D. Ngo, S. Wei, A. Holubová, J. Muzik, F. Godtliebsen, Reinforcement-learning optimal control for type-1 diabetes, in: 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), IEEE, 2018, pp. 333–336.

[59] M. H. Lim, W. H. Lee, B. Jeon, S. Kim, A blood glucose control framework based on reinforcement learning with safety and interpretability: In silico validation, IEEE Access 9 (2021) 105756–105775.

[60] M. Tejedor, A. Z. Woldaregay, F. Godtliebsen, Reinforcement learning application in diabetes blood glucose control: A systematic review, Artificial Intelligence in Medicine 104 (2020) 101836.

[61] I. Fox, J. Lee, R. Pop-Busui, J. Wiens, Deep reinforcement learning for closed-loop blood glucose control, in: Machine Learning for Healthcare Conference, PMLR, 2020, pp. 508–536.

[62] S. Lee, J. Kim, S. W. Park, S.-M. Jin, S.-M. Park, Toward a fully automated artificial pancreas system using a bioinspired reinforcement learning design: In silico validation, IEEE Journal of Biomedical and Health Informatics 25 (2020) 536–546.

[63] A. Naik, R. Shariff, et al., Discounted reinforcement learning is not an optimization problem, arXiv preprint arXiv:1910.02140 (2019).

[64] C. Hettiarachchi, N. Malagutti, C. Nolan, H. Suominen, E. Daskalaki, Non-linear continuous action spaces for reinforcement learning in type 1 diabetes, in: 2022 35th Australasian Joint Conference on Artificial Intelligence (AJCAI), Springer, 2022, pp. in–press.

[65] B. P. Kovatchev, W. L. Clarke, M. Breton, K. Brayman, A. McCall, Quantifying temporal glucose variability in diabetes via continuous glucose monitoring: mathematical methods and clinical application, Diabetes Technology & Therapeutics 7 (2005) 849–862.

[66] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

[67] L. Ljung, C. Andersson, K. Tiels, T. B. Schön, Deep learning and system identification, IFAC-PapersOnLine 53 (2020) 1175–1181.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019) 8026–8037.

[69] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[70] C. Hettiarachchi, N. Malagutti, C. Nolan, E. Daskalaki, H. Suominen, A reinforcement learning based system for blood glucose control without carbohydrate estimation in type 1 diabetes: In silico validation, in: 2022 35th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE, 2022, pp. 950–956.

[71] R. M. Bergenstal, W. V. Tamborlane, A. Ahmann, J. B. Buse, G. Dailey, S. N. Davis, C. Joyce, T. Peoples, B. A. Perkins, J. B. Welsh, et al., Effectiveness of sensor-augmented insulin-pump therapy in type 1 diabetes, New England Journal of Medicine 363 (2010) 311–320.

[72] C. Roversi, M. Vettoretti, S. Del Favero, A. Facchinetti, G. Sparacino, H.-R. Consortium, Modeling carbohydrate counting error in type 1 diabetes management, Diabetes Technology & Therapeutics 22 (2020) 749–759.

[73] J. Walsh, R. Roberts, T. Bailey, Guidelines for optimal bolus calculator settings in adults, Journal of Diabetes Science and Technology 5 (2011) 129–135.

[74] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, T. Hester, An empirical investigation of the challenges of real-world reinforcement learning, arXiv preprint arXiv:2003.11881 (2020).

[75] S. S. Shapiro, M. B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (1965) 591–611.

[76] H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, The Annals of Mathematical Statistics (1947) 50–60.

[77] C. O. Fritz, P. E. Morris, J. J. Richler, Effect size estimates: current use, calculations, and interpretation., Journal of Experimental Psychology: General 141 (2012) 2.

[78] L. Li, Matlab user manual, Matlab: Natick, MA, USA (2001).

[79] G. vanRossum, F. L. Drake, Python reference manual, Python Software Foundation: Amsterdam, Netherlands (2010).