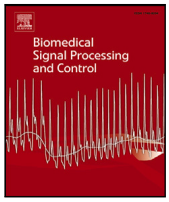
Biomedical Signal Processing and Control 90 (2024) 105839

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc



G2P2C — A modular 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,c,d], Elena Daskalaki [a]

[a] School of Computing, College of Engineering, Computing & Cybernetics, The Australian National University, Canberra, Australia
[b] School of Engineering, College of Engineering, Computing & Cybernetics, The Australian National University, Canberra, Australia
[c] School of Medicine and Psychology, College of Health & Medicine, The 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 APSs 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 modular 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 (180 g of CHO per day) for 20 subjects (10 adults and 10 adolescents) using an open-source version of a T1D simulator approved by the United States Food and Drug Administration (FDA). G2P2C was compared with state-of-the-art RL algorithms 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 statistically 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 APSs. We released the codebase of G2P2C (https://github.com/chirathyh/G2P2C) and an online demonstration tool (https://capsml.com/), where users can perform custom simulations to compare G2P2C with 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

* Corresponding author.
E-mail addresses: 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).

https://doi.org/10.1016/j.bspc.2023.105839
Received 3 June 2023; Received in revised form 13 November 2023; Accepted 9 December 2023
Available online 28 December 2023

varying insulin requirements related to sleep patterns, meals, and exercise. The objective is to improve the time spent in the normoglycemic range (70–180 mg/dL) while minimizing periods of low blood glucose (hypoglycemia) and high blood glucose (hyperglycemia) which are detrimental to health. Recently-introduced commercial hybrid closed-loop systems can continuously monitor glucose levels in subcutaneous interstitial fluid through a Continuous Glucose Monitor (CGM) and infuse insulin subcutaneously via a pump attached to the body [2–4]. 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 min 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]. As a result of these limitations, 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). Additional meal boli are delivered during the active part of the day according to manual user input. Research into control strategies for an Artificial Pancreas System (APS), which aims to automate insulin administration completely to alleviate the burden on users is ongoing [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. These challenging conditions result in a feedback control problem that cannot be fully handled by existing control strategies such as PID and MPC [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 for task requirements [16]. Hence, RL-based algorithms are currently being explored in multiple health care applications, including glucose control in T1D [18], propofol dosing in general anaesthesia [19], and multi-cytokine therapy for sepsis [20]. In a RL algorithm, an ***agent*** seeks 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 board/computer games [21], continuous control problems such as 3D humanoid motion problems, and physics simulations [22]. However, real-world reinforcement learning systems must contend with many challenges, such as partial observability, high-dimension state/action spaces, large sensor/actuator delays, safety constraints, formulation of reward functions, explainability, and practical learning constraints [23]. The application of RL to the continuous control problem of glucose regulation in Type 1 Diabetes (T1D) [1] presents all of the above challenges.

In this study, we propose a fully automated RL-based APS while focusing on a subset of the identified challenges. Specifically, we explore the impact towards safety, resulting from the lack of knowledge to formulate an ideal reward function and subsequent practical limitations

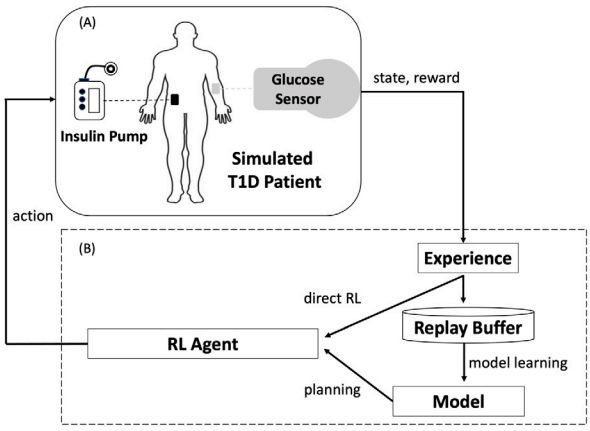


Fig. 1. An Artificial Pancreas System (APS) consists of (A) a glucose sensor and an insulin pump attached to a Type 1 Diabetes (T1D) patient and (B) a control algorithm, such as G2P2C based on Reinforcement Learning (RL) we proposed in this paper.

of the standard RL optimization objective. Although most real-life control problems tend to have both short- and long-term objectives, formulating reward functions to capture them both is challenging or even infeasible [24]. To target capturing the two objectives at once, prior work has proposed model-based RL approaches that leverage short/long-term prediction models [24] and learning value functions over multiple time horizons [25–27]. However, these algorithmic advances remain insufficient in glucose control, where safety is a primary consideration related to device use or technology characterization [28]: The long-term objective is to improve the time spent in the desirable normoglycemic range in the presence of the hard constraint of low blood glucose (hypoglycemia) and high blood glucose (hyperglycemia) being detrimental to health [1]. Because of large inter- and intra-population variability in glucose dynamics in people with T1D, developing personalized reward functions presents many challenges. Ideally, optimizing for the long-term through an infinite horizon RL objective is expected to result in learning a policy that is also optimal in the short-term. Yet, in practice, the learned policies for glucose control are sub-optimal in the short term, resulting in the potential for catastrophic failures [29]. Even though infrequent, these failures are unacceptable for a high-risk medical application since they could result in death (e.g., severe short-term hypoglycemia). As part of this work, instead of using a personalized reward function designed to capture long/short-term characteristics, we contribute by introducing a safe scalable approach for the planning phase of RL, a novel optimization procedure with a fine-tuning mechanism to a short-time horizon as a way to improve the learned policy by integrating a short-term optimization objective. Hence, we prioritize safety in our study that proposes and validates an entire end-to-end RL method transparently, using accessible and curated materials; makes an open-source code release to assure the repeatability and reproducibility of our research; and completes this by having implemented an online demonstration system.

Previous research on the safety of RL has explored safe exploration strategies [30,31] and the use of human feedback for avoiding failures [32]. APSs are categorized as high-risk medical devices, and the best practice for developing control algorithms is first to use Food & Drug Administration (FDA)-approved simulators [12] which provide *in-silico* patient populations which reflect real-world T1D populations [33]. These algorithms are later fine-tuned and personalized periodically to individuals [34]. Designing an RL-based APS, which learns online on real-world people, is practically infeasible as RL algorithms typically require large amounts of experience to learn a stable policy for which years of training is required (one experience would

require five minutes). Offline strategies would also still require extensive experience and current clinical treatment strategies used to collect such experience may not provide valuable information due to their static rule-based nature [34]. Hence, the most appropriate solution is to design the system using the simulators and focus on transfer learning approaches to fine-tuning. Consequently, in this work, we do not focus on the safety requirements in exploration but only on learning a reasonable control policy. We reserve, as future work, to explore strategies to safely transfer and personalize the learned policy to the real world.

To address identified challenges, we propose a modular RL algorithm to (i) fully automate glucose control to reduce the cognitive burden on the users and (ii) improve performance and safety by focusing on the short-term outcomes of the problem. Our algorithm extended the state-of-the-art model-free Proximal Policy Optimization (PPO) algorithm [22], used in continuous control applications, by introducing two optimization phases, model learning and planning. In view of these added features, we named our algorithm G2P2C — Glucose Control by Glucose Prediction and Planning (Fig. 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 policy to a short-time horizon by conducting model-based simulations to improve safety. Therefore the proposed approach could combine the characteristics of both model-free and model-based methods. The G2P2C algorithm improves sample efficiency and possesses desirable characteristics towards offline personalization, which is expected to be valuable. Furthermore, G2P2C, aims to fully automate glucose control in T1D, eliminating the need for CHO estimation and meal announcement. 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 evaluate G2P2C with standard state-of-the-art RL algorithms and standard clinical treatment benchmarks.

2. Related work

Reinforcement Learning for Continuous Control. The use of RL for continuous control has gained much attention due to applications such as locomotion, self-driving, and dexterous manipulation tasks [35–38]. Both on-policy (Advantage Actor Critic (A2C) [39], PPO [22]) and off-policy (Soft Actor Critic (SAC) [36]) algorithms have been used in model-based [40] and model-free [41] 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 [42]. On the other hand, they heavily depend on the accuracy of the learned model [43]. 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 [21,42]. Several prior works have focused on combining model-free and model-based paradigms to design RL systems [21,42,44,45].

PPO [22] is a widely used model-free on-policy algorithm that has shown promise in many RL tasks [46]. In standard on-policy policy gradient methods, past experience 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 sample of experiences. 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 [47]. The former facilitates feature sharing between the two functions, while the latter avoids interference between the two optimization objectives.

Auxiliary Learning. The Phasic Policy Gradient (PPG) algorithm [47] 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 [48,49]. 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 [50]. Previous work in RL carries out auxiliary model learning by predicting latent state representations [48,51], predicting observations/system states [21,42], and through estimating future rewards, value and policy functions [50].

Planning. A learned 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 [44] 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. Previous work has introduced such methods based on Monte Carlo Tree Search (MCTS), where an expert policy is used for planning [52].

Reinforcement Learning for Glucose Control. The application of RL in regulating blood glucose levels in T1D dates back to 2012 when Daskalaki et al. [18,53] 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 [53–55]. The development of the open source Simglucose simulator in 2018 [56] (based on FDA-approved UVA/PADOVA-2008 Model [12]) and UVA/PADOVA (2014) simulator [57] has resulted in recent studies with real-time control strategies (e.g., every 5 min). Many of these studies are hybrid and based on Deep Q-Learning approaches that require manual decision-making [58–63]. Lim et al. [64] has developed a system that uses a SAC algorithm to estimate insulin guided by PID control. These systems have predominantly used discrete handcrafted insulin action spaces [58–60]. The latest research focuses on developing fully autonomous RL systems for glucose control [65]. Fox et al. [66] have proposed a system based on SAC, Lee et al. [67] a bio-inspired RL approach using PPO, and Emerson et al. [68] a method based on offline RL. Furthermore, Sun et al. [69] and Askari et al. [70] have proposed adaptive MPC algorithms for glucose regulation. A systematic review of studies on RL for glucose control is presented in Tejedor et al. [65] and Yau et al. [71].

3. Method

3.1. Formulating glucose control as a RL task

The glucoregulatory system is only partially observable through noisy glucose sensor measurements [12]. Hence, the environment of the RL task was formulated as a Partially Observable Markov Decision Process (POMDP). The 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 $(\pi(a|s))$.

We defined the glucose control problem as a continuing task, with the goal to maximize the average reward [17,72]

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

The policy π induced a value function $(v^{\pi}(s_t) \doteq \mathbb{E}[G_t|s_t])$ to estimate the expected return G_t at time t when starting from the state s_t and following the policy π. The return G_t was estimated according to

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

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

$$A^{\pi}(s_t, a_t) \doteq G_t - v^{\pi}(s_t) \quad (3)$$

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

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,

$$s_t = (g_{t-k:t}, i_{t-k:t}). \quad (4)$$

Action Space. We used a continuous action space $A([-1, 1])$, as it provides additional flexibility to the RL agent to learn a control policy compared to a handcrafted discrete action space. In previous work Hettiarachchi et al. [73] has analysed the use of continuous action spaces for this problem and proposed a non-linear control space representation (Eq. (5)). We use this formulation, where the action space (A) is mapped to the control space (insulin infusion rate) of the insulin pump $(I_{pump} \in [0, 5]$ U/min). The parameter η was set to 4.0 and I_{max} to 5 U/min [73].

$$I_{pump} = I_{max} \cdot e^{\eta(a-1)}, a \in [-1, 1]. \quad (5)$$

Reward Function. The reward function was formulated based on the blood glucose Risk Index (RI), similar to the work of Hettiarachchi et al. [29],Fox et al. [66],Emerson et al. [68],Mackey and Furey [74]. The RI proposed by Kovatchev et al. [75] weights the risk of blood glucose levels based on the clinical metrics Low Blood Glucose Risk Index (LBGI) and High Blood Glucose Risk Index (HBGI) (Fig. 2). In this study we use the fine-tuned reward function proposed in our previous work [29], where we additionally introduced a penalty for hypoglycemia ($g \leq 39$ mg/dL) and normalized the RI $(\overline{RI})$ to [0, −1] for the rest of the glucose range (Eq. (6)). The additional penalty targets severe hypoglycemia, which is more frequent and life-threatening.

$$R(s_t, a_t) = \begin{cases} -15 & \text{if } g_{t+1} \leq 39 \text{ mg/dL} \\ -1 \cdot \overline{RI}(g_{t+1}) & \text{else} \end{cases}. \quad (6)$$

3.2. Algorithm: G2P2C (glucose control by glucose prediction and planning)

G2P2C is a modular RL algorithm based on PPO, which introduce two additional optimization phases, namely, model learning and planning. PPO was selected as the basis for designing G2P2C due to its demonstrated efficiency in safety-critical applications, where excessive changes in the control policy could lead to unexpected behaviour. G2P2C was implemented using two separate neural networks with similar architecture; the Actor-Network (Π_{θ}) and the Critic-Network (V_{ϕ}) (Fig. 3).

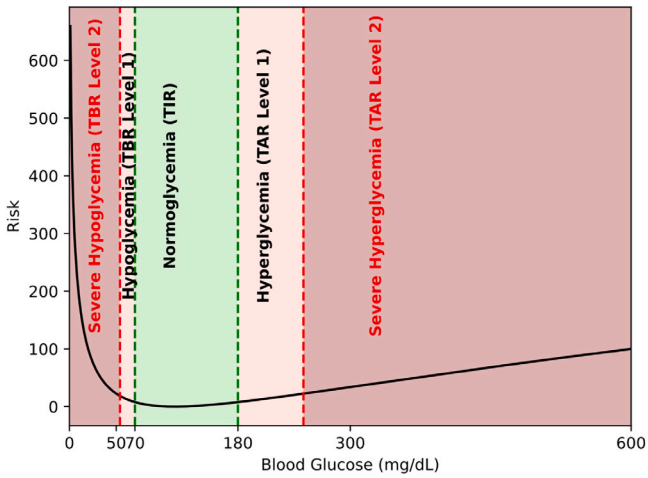


Fig. 2. Blood glucose risk index (severe hypoglycemia (<54 mg/dL), hypoglycemia (<70 mg/dL), normoglycemia (70–180 mg/dL), hyperglycemia (>180 mg/dL), and severe hyperglycemia (>250 mg/dL)).

Both networks consisted of initial feature extractor modules E^{Π} and E^{V} based on a Long Short-Term Memory (LSTM) network [76], where the input was the observed state (s_n) and the output was the hidden state vector (h_n) of the LSTM. The Π_{θ} network included the policy module π, which represented the policy function, while V_{ϕ} included the value module v which represented the value function. They used h_n as the input. 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 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 estimate (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, as hidden states (h_n) of LSTM networks are capable of learning a representation of the state space $(s_n^{\star})$ of a dynamical system [77]. The glucose prediction modules were implemented using similar architectures and trained for similar tasks to facilitate feature distillation between the networks, inspired by Cobbe et al. [47].

G2P2C alternated between sampling and optimization. During sampling, 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 Eq. (7) 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 Eq. (2) was used to calculate the advantage function estimate $\hat{A}_t$ and value function targets $\hat{v}_t^{target}$.

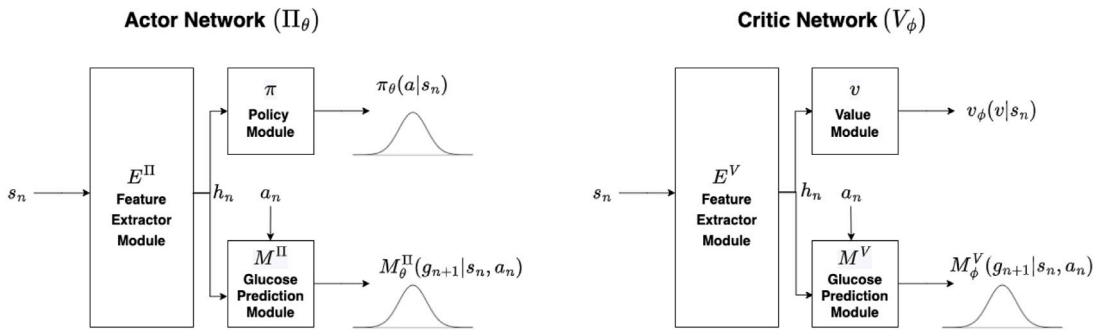


Fig. 3. Schematic diagram of actor and critic networks.

The value module objective was to minimize the objective function $L^{v}(\phi)$ defined in Eq. (8).

$$L^{\pi}(\theta) = \hat{\mathbb{E}}_t \left[min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t, clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) + \beta_s H\left(\pi(\cdot|s_t)\right) \right]. \quad (7)$$

$$L^{v}(\phi) = \hat{\mathbb{E}}_t \left[\frac{1}{2} \left(v_\phi(s_t) - \hat{v}_t^{target} \right)^2 \right]. \quad (8)$$

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 Eq. (9), (10). The Kullback–Leibler 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.

$$L^{M^{\Pi}}(\theta) = \hat{\mathbb{E}}_t \left[-log\left(M^{\Pi}_\theta(g_{t+1}|s_t, a_t) \right) + \beta_1 d_{KL}\left[\pi_{\theta_{ppo}}(\cdot|s_t), \pi_\theta(\cdot|s_t) \right] \right]. \quad (9)$$

$$L^{M^{V}}(\phi) = \hat{\mathbb{E}}_t \left[-log\left(M^{V}_\phi(g_{t+1}|s_t, a_t) \right) + \beta_2 \frac{1}{2} \left(v_{\phi_{ppo}}(s_t) - \hat{v}_\phi(s_t) \right)^2 \right]. \quad (10)$$

3.2.3. Planning phase

Following the model learning phase, the third and final update was performed during the planning phase (Fig. 4). 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 actor network Π_θ 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 min)) 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 Eq. (11). 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 (Eq. (12)). 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 G2P2C are summarized in Algorithm 1.

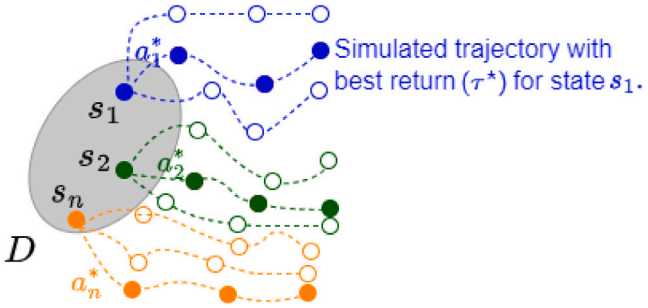


Fig. 4. In this illustration of the planning phase, we simulate short-horizon trajectories using the learned glucose dynamics model for the states stored in the rollout buffer D. Planning updates use the initial action with the best trajectory for each state.

$$\tau^* = \arg\max_{\tau} \left[\left(\sum_{q=1}^{n_{plan}} R_q \right) + v(s_{n_{plan}}) \right]. \quad (11)$$

$$L^{plan}(\theta) = \hat{\mathbb{E}}_t \left[-log\left(\pi_\theta(a_t^*|s_t) \right) \right]. \quad (12)$$

3.3. Experiments

3.3.1. T1D simulator and experiment setup

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 [78] machine learning framework for the RL algorithm development, we used an open-source Python implementation of this simulator named Simglucose [56], used in previous work [29,60,66]. 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 simulates the errors associated with the selected sensors and pumps while allowing for the definition of different meal protocols for simulations. For the experiments, the 10 adult and 10 adolescent cohorts were used along with the Insulet pump and the GuardianRT glucose sensor provided by the simulator [29].

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

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 (Π) and Critic-Network (V) weights (θ, ϕ).
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^{Π} wrt θ, 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 ϕ, on all data in D:
      $\phi \leftarrow \phi - \alpha_3 \cdot \nabla_\phi L^{V}(\phi)$.
  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 θ, on all data in B:
        $\theta \leftarrow \theta - \alpha_4 \cdot \nabla_\theta L^{M^{\Pi}}(\theta)$.
      optimize $L^{M^{V}}$ wrt ϕ, 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^{Π}, M^{Π} modules of the actor network (Π).
    for $epoch = 1, 2, 3, \cdots, E_{plan}$ **do**
      optimize L^{plan} wrt θ, 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

The G2P2C algorithm (Section 3.2), state-of-the-art RL benchmarks (Section 3.4), and standard clinical benchmarks (Section 3.5) were implemented as candidate control algorithms for the experiment. The study used two challenging meal protocols with large CHO contents; (1) a training protocol — for conducting training simulations for the RL-based algorithms and (2) an evaluation protocol — for conducting evaluation simulations for the candidate algorithms. The training protocol was challenging as the meals were uniformly randomized based on CHO content and time of the day (Table 1). The probability of occurrence of a meal was set to 0.95 to replicate missed meals by the user.

Table 1
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

The evaluation of the candidate algorithms was performed using the same two cohorts of the simulator. To support comparisons with other works, the evaluation meal protocol used in [67] was chosen for all algorithms. The evaluation meal protocol spanned 24 h starting at 00:00 h fixed with three meals: 40 g of CHO for breakfast at 8:00 h, 80 g of CHO for lunch at 13:00 h, and 60 g of CHO for dinner at 20:00 h. We restricted the initial blood glucose level between 110–130 mg/dL for the evaluation simulations, to ensure fair comparisons between the candidate algorithms. The research domain of developing control algorithms for T1D lacks established standardized benchmarks for meal protocols. Hence, challenging protocols with large CHO contents identified in previous work was used in this study [18,53,67]. However, to provide interested readers to evaluate the proposed G2P2C algorithm on custom meal protocols, we developed an online demonstration tool named CAPSML (https://capsml.com/). We also used the results presented in previous RL-based glucose control algorithms research for completeness in comparisons[2] and provide a discussion in Section 5.

3.4. RL benchmarks

The state-of-the-art RL algorithms A2C, PPO, and SAC were implemented as RL benchmarks. G2P2C and all benchmark RL algorithms used comparable network architectures (see Appendix A). A2C and PPO algorithms were also designed based on the average reward RL setting. SAC was the only off-policy algorithm implemented and used automated temperature tuning [36]. For each RL-based algorithm, unique models were trained for each of the *in-silico* subjects. All the candidate RL algorithms were trained for 800,000 environment steps[3] for three random seeds (see Appendix B for their hyperparameters). 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 the RL algorithms. Once training was fully complete, the RL algorithms were evaluated using 1,500 evaluation simulations (i.e., 3 random seeds x 500 simulations/subject) conducted for each subject.

3.5. Clinical benchmarks

Basal-Bolus (BB) insulin treatment strategies, which use meal announcement and CHO estimation, were used as clinical benchmarks (Table 2). 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 [79]. This treatment strategy is based on patient-specific characteristics and requires prior knowledge about the CHO content of future meals (typically 20 min in advance).

[2] Please note that such comparisons with prior work are rare due to differences in experimental conditions; research on glucose control tends to benchmark the performance of algorithms with standard clinical treatment approaches and guidelines, as presented in this work.

[3] An environment step refers to an insulin delivery action taken every 5 min based on the sampling rate of the insulin pump and glucose sensor. The total environment steps (800,000) were the summation of steps by all parallel agents used in the RL algorithms (e.g., see Section 3.2).

Table 2
Summary of candidate glucose control algorithms used in this study.

Algorithm	Meal announcement & CHO estimation required	Characteristics	Reference
BBI	Yes	**A clinical basal-bolus strategy.** Basal-Bolus insulin infusion strategy based on personalized 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) applied 20 min in advance based on accurate CHO information. Correction insulin bolus (based on ISF of $\frac{1800}{\text{TDI}}$ mg/dL per U & target blood glucose of 140 mg/dL) applied automatically when $g_t > 150$ mg/dL.	[8,66,79]
BBHE	Yes	**A clinical basal-bolus strategy.** Same as for BBI, but including human error in CHO estimation of meals.	[8,66,79]
A2C	No	**A state-of-the-art RL algorithm.** Implemented as two neural networks (Appendix A).	[39]
PPO	No	**A state-of-the-art RL algorithm.** Implemented as two neural networks (Appendix A).	[22]
SAC	No	**A state-of-the-art RL algorithm.** Implemented as two neural networks (Appendix A).	[36]
G2P2C	No	**The Proposed RL-based algorithm.** Designed based on PPO. Novel model-learning (additional M^R, M^V modules) and planning phases introduced.	

Acronyms: A2C: Advantage Actor Critic, 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, SAC: Soft Actor Critic, TDI: Total Daily Insulin.

We replicated two versions of the BB treatment based on the previous work of [66]; 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 [80]. 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 [66]. The final insulin delivered over time I_t is presented in Eq. (13). 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 personalize the Basal Rate $\left(\text{BR} = 0.48 \cdot \text{TDI}\ \text{U/day}\right)$, Carbohydrate Insulin Ratio $\left(\text{CIR} = \frac{500}{\text{TDI}}\ \text{g/U}\right)$, and Insulin Sensitivity Factor $\left(\text{ISF} = \frac{1800}{\text{TDI}}\ \text{mg/dL per U}\right)$ [81]. The meal bolus was calculated as $\frac{c_t}{\text{CIR}}$ U, and delivered 20 min before a meal.

The correction insulin bolus was calculated as $\frac{(g_t - g_{target})}{\text{ISF}}$ 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 [66] 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 h 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 h duration and to 0 otherwise. We also conducted 1,500 evaluation simulations for each subject under the BBI and BBHE approaches for comparison. A summary of the candidate algorithms are presented in Table 2.

$$I_t = \text{BR} + (c_t > 0) \cdot \left(\frac{c_t}{\text{CIR}} + cool \cdot \frac{g_t - g_{target}}{\text{ISF}} \right). \tag{13}$$

3.6. Evaluation metrics

Both clinical metrics and RL-based metrics were used for the evaluation. The clinical metrics included the standard T1D metrics of glucose risk indices (Risk Index (RI), High Blood Glucose Index (HGBI), Low Blood Glucose Index (LGBI)) [75] 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)**, 70–180 mg/dL), and minimize the time spent in other glucose regions (severe hypoglycemia (<54 mg/dL), hypoglycemia (<70 mg/dL), hyperglycemia (>180 mg/dL), and severe hyperglycemia (>250 mg/dL)). These metrics were standardized by the Advanced Technologies & Treatment for Diabetes (ATTD) Congress in 2019 through consensus received from a panel of international physicians, researchers, and individuals with T1D, to address the lack of consistency in the practical application of the metrics [82].

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 ***failures*** were calculated as a percentage of the total evaluation simulations. A glucose level ≤ 40 mg/dL is life-threatening and can result in major cardiovascular and cerebrovascular problems [83]. A glucose value ≥ 600 mg/dL is also a life-threatening emergency referred to as the Hyperosmolar hyperglycemic state [84].

The RL algorithms were trained based on the objective to maximize the expected cumulative reward. Hence the ***total reward*** $\in [-15, 288]$ achieved in evaluation simulations was used to compare the performance of RL algorithms. Based on the defined reward function (Section 3.1) theoretical maximum total reward achievable by following a perfect glucose control strategy was bounded by 288, while the minimum was −15.

Statistical significance analysis was conducted to compare selected candidate algorithms. A Shapiro–Wilk Test [85] was performed to check the normality, Mann–Whitney U Test [86] was conducted to evaluate significance, and effect size calculated using the Pearson product-moment correlation coefficient (r) [87]. The statistical analysis was conducted using the IBM SPSS Statistics Software (Version-28.0.1.1). The analysis first compared the total reward performance of G2P2C with the best performing state-of-the-art RL algorithm on an individual *in-silico* subject level. Next, the TIR performance of G2P2C was compared with standard clinical treatment approaches (BBI, BBHE) on a cohort level.

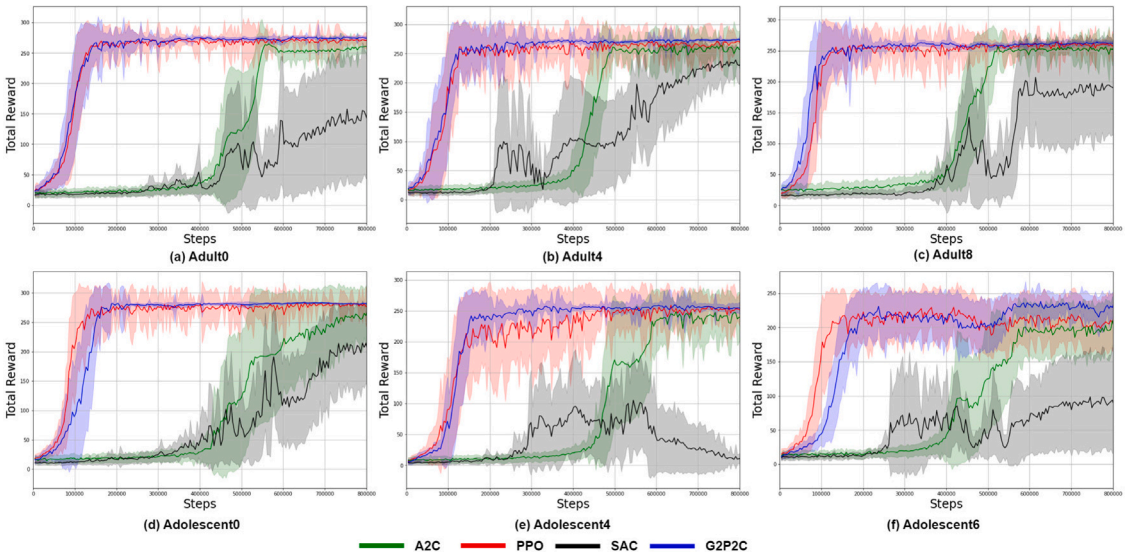


Fig. 5. Training curves for selected *in-silico* subjects. Mean and standard deviation of the total reward achieved against environment steps for evaluation simulations are presented. G2P2C and PPO achieves steady learning. Complete results in Appendix C.

Table 3
Performance comparison of candidate glucose control algorithms averaged for adult and adolescent cohorts. See Table 5 for detailed clinical results.

Algorithm	TIR	Failures	Total reward
Adults			
BBI	71.02 ± 11.29	0.39	–
BBHE	69.78 ± 11.29	0.35	–
A2C	59.06 ± 14.31	9.11	244 ± 47
PPO	69.12 ± 10.53	2.79	264 ± 26
SAC	61.76 ± 21.01	59.49	146 ± 98
G2P2C	72.69 ± 9.53	1.62	268 ± 21
Adolescents			
BBI	71.43 ± 12.31	0.00	–
BBHE	70.23 ± 12.52	0.00	–
A2C	56.03 ± 14.40	14.41	227 ± 48
PPO	63.72 ± 13.95	4.93	249 ± 31
SAC	65.62 ± 20.16	82.06	107 ± 89
G2P2C	64.33 ± 13.18	1.48	254 ± 22

Acronyms: A2C: Advantage Actor Critic, BBHE: Basal Bolus Human Error, BBI: Basal Bolus Ideal, G2P2C: Glucose Control by Glucose Prediction and Planning, PPO: Proximal Policy Optimization, SAC: Soft Actor Critic, TIR: Time In Range.

4. Results

4.1. RL analysis

G2P2C outperformed all RL algorithms in terms of TIR, failure, and total reward metrics (Table 3). A2C and SAC performed the lowest, where SAC had the highest failures (Fig. 5). PPO and G2P2C were the best performing algorithms. However, 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 improving the reward performance.

G2P2C achieved a statistically significant ($P < 0.05$) total reward improvement compared to PPO for 18 of the 20 subjects (Table 4). The performance improvement was non-significant for Adolescent0 and Adolescent4. Adolescent0 was the easiest to control under both RL-based methods (280.79, 282.20) leaving no margin for improvement for G2P2C. Adolescent6 (209.93, 231.97) was the hardest to control for both RL algorithms. G2P2C gave the largest improvement in the total reward performance for Adolescent6, where it achieved a total reward of 231.97 compared to 209.93 in PPO.

The standard deviation of the total reward of the evaluation simulations were reduced in 18 out of the 20 subjects (except for Adult5 and Adult6) (Table 4); 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 (Figs. 5 and Appendix C for detailed results). 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 (Figs. 5 and Appendix C for detailed results). However, the planning phase of G2P2C was automatically initiated at approximately 200,000 learning steps once the learned glucose prediction module (M^{Π}) achieved the predefined accuracy threshold as described in Section 3.2. By analysing 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) (Figs. 5).

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 5. 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. G2P2C, PPO, 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

Table 4
Comparison of total reward for all subjects based on evaluation simulations for PPO and G2P2C.

Subject	PPO	G2P2C	Significance (PPO - G2P2C)
Adult0*	270.71 ± 12.74	**275.03 ± 2.94**	$P < .001$, $r = 0.34$
Adult1*	275.60 ± 9.77	**277.36 ± 2.85**	$P < .001$, $r = 0.14$
Adult2*	251.52 ± 42.69	**264.73 ± 18.80**	$P < .001$, $r = 0.44$
Adult3*	255.48 ± 34.93	**261.05 ± 28.40**	$P < .001$, $r = 0.21$
Adult4*	266.10 ± 26.53	**272.81 ± 14.48**	$P < .001$, $r = 0.51$
Adult5*	253.94 ± 27.34	**253.98 ± 42.61**	$P < .001$, $r = 0.30$
Adult6*	265.15 ± 21.62	**265.87 ± 28.54**	$P < .001$, $r = 0.22$
Adult7*	272.93 ± 9.76	**275.23 ± 5.11**	$P < .001$, $r = 0.22$
Adult8*	258.68 ± 18.04	**262.61 ± 4.07**	$P < .001$, $r = 0.19$
Adult9*	266.28 ± 16.86	**267.42 ± 3.10**	$P < .001$, $r = 0.06$
Adolescent0	280.79 ± 14.79	**282.20 ± 1.49**	$P = .686$, $r = 0.01$
Adolescent1*	229.18 ± 19.20	**235.09 ± 16.04**	$P < .001$, $r = 0.30$
Adolescent2*	258.27 ± 14.79	**259.24 ± 9.92**	$P = .01$, $r = 0.05$
Adolescent3*	251.84 ± 21.35	**259.44 ± 12.72**	$P < .001$, $r = 0.42$
Adolescent4	253.39 ± 21.53	**253.93 ± 10.86**	$P = .831$, $r = 0.00$
Adolescent5*	259.45 ± 32.03	**262.36 ± 26.89**	$P < .001$, $r = 0.21$
Adolescent6*	209.93 ± 36.20	**231.97 ± 21.85**	$P < .001$, $r = 0.44$
Adolescent7*	231.91 ± 26.24	**238.09 ± 9.08**	$P < .001$, $r = 0.17$
Adolescent8*	267.80 ± 12.48	**268.13 ± 5.94**	$P < .001$, $r = 0.09$
Adolescent9*	249.40 ± 27.14	**252.89 ± 24.62**	$P < .001$, $r = 0.18$

* Statistical significance ($P < 0.05$). 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.

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 [57]) and a low-risk profile in terms of LGBI(1.1≤LGBI≤2.5 [57]). 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.

The insulin administration strategies learned by the RL-based algorithms were much more complex compared to the BB strategies (Appendix D). This highlights the importance of focussing on the explainability of RL-based APSs. Example simulated glucose control trajectories for the candidate algorithms focusing on Adolescent0 are presented in Appendix D. An online demonstration tool named CAPSML (https://capsml.com/), released as part of this study, can be used to run custom simulations for the *in-silico* T1D subjects to visualize the performance of the candidate algorithms on custom simulations.

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 with standard treatment methods commonly used in clinical practice and state-of-the-art RL algorithms.

Principal Findings. G2P2C outperformed all the state-of-the-art RL algorithms in terms of TIR, failure, and reward metrics (Table 3), while PPO and G2P2C were the best performing RL algorithms (Fig. 5). However, G2P2C was able to improve safety by reducing catastrophic failures compared to PPO while also improving reward performance. G2P2C improved the reward performance of all 20 subjects while achieving statistically significant ($P < 0.05$) improvements for 18 subjects. 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, upon which both BB methods rely. The improvement was statistically significant ($P < 0.05$) compared to BBI ($r = 0.09$) and BBHE ($r = 0.14$). For the challenging adolescent cohort, it achieved a TIR of 64.33%, which can be further improved compared to the gold standard benchmark. However, the performance comparison should account for G2P2C receiving no prior information about upcoming meals. We provided a comprehensive clinical analysis and example glucose simulation graphs in Table 5 and Appendix D.

Optimization Objective. A2C, PPO, and SAC were designed based on an long horizon standard RL objective to reflect the clinical goal in T1D to improve the TIR. However, we observed that optimizing only for the long horizon was insufficient as they resulted in catastrophic failures in the short-term. 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 learned policy for the short horizon. The fine-tuning of the policy to the short-term improved the performance in G2P2C while reducing the variability in performance (Fig. 5). We designed G2P2C using an average reward RL objective compared to the popular discounted RL setting. This was due to the high glucose variability observed among subjects (Adolescent0 was the easiest, while Adolescent6 was the hardest to control (Table 4)), which required personalized tuning of the discounting rate to learn satisfactory control policies.

Characteristics of G2P2C. G2P2C improved the sample efficiency by re-using collected experiences and facilitating a simulation-based exploration of the glucose state space using the learned model. This should be beneficial in real-world online learning applications where real experience is limited to tune the policy. The learned glucose dynamics model in G2P2C could also be valuable in feature distillation between the actor and critic networks, and in designing safety modules for an APS (e.g., to predict future high/low blood glucose events).

Comparison with Previous Work. Research on glucose control often benchmarks the performance of algorithms with 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 (please refer to Table 8 in Appendix E for further details), and the unavailability of the algorithm and simulator codebases/software. 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 previous work in terms of TIR (Table 6). In our comparison, we consider RL-based fully autonomous systems as well as the study [64], 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 type of population used for the algorithmic assessment. Some previous studies have only considered handpicked individual subjects [60] or adult-only cohorts [67]. [66] has considered child, adult, and adolescent cohorts, however, reports the overall median TIR values. [64] has used the adolescent and adult cohorts and presented 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 [60,66] while others have focused on more challenging meal protocols with 180 g of CHO daily [64,67]. 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 [88]. However, previous studies have used

Table 5
Clinical performance comparison for candidate algorithms. The median, inter-quartile range followed by the mean (standard deviation) are presented.

Method	Failure (%)	Severe Hypo. (%)	Hypo. (%)	Normo. (TIR) (%)	Hyper. (%)	Severe Hyper. (%)	RI	LBGI	HBGI
Adult									
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)
A2C	9.11	0.00	0.00	58.68	23.61	14.24	12.07	0.54	10.50
		0.00–0.00	0.00–1.04	48.82–68.75	17.36–29.51	6.25–22.92	9.15–15.90	0.03–2.64	7.41-13.95
		0.84(2.93)	1.48(3.56)	59.06(14.31)	23.12(9.95)	15.49(11.95)	13.15(5.77)	2.14(3.43)	11.00(5.44)
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)
SAC	59.49	2.64	2.86	66.67	12.85	1.77	14.69	7.25	6.01
		0.00–8.45	0.00–7.83	45.83–78.95	0.00–22.92	0.00–27.08	10.90–21.82	1.27–12.07	1.44-16.06
		5.31(6.86)	5.11(6.17)	61.76(21.01)	13.13(12.25)	14.69(19.70)	17.39(9.11)	7.27(5.69)	10.12(10.55)
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)
A2C	14.41	0.00	0.00	54.86	16.32	26.04	17.55	0.49	16.07
		0.00–0.00	0.00–0.69	46.88–62.38	11.46–22.22	17.36–34.72	13.33–23.22	0.01–2.32	11.90-21.67
		0.83(3.04)	1.55(3.84)	56.03(14.40)	16.57(8.30)	25.03(14.35)	18.03(7.12)	1.94(3.16)	16.09(7.60)
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)
SAC	82.06	4.70	4.94	71.43	3.12	0.00	14.83	9.13	3.06
		1.01–11.54	1.04–9.38	56.25–80.25	0.00–17.50	0.00–17.78	11.62–21.47	4.19–12.62	1.01-13.85
		6.85(7.00)	6.26(6.15)	65.62(20.16)	9.99(12.56)	11.27(16.79)	16.95(7.49)	8.46(5.35)	8.49(9.78)
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). Acronyms: A2C: Advantage Actor Critic, 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, SAC: Soft Actor Critic, TIR: Time in Range.

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 [89] 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, [64,67] have used independent customized platforms based on the UVA/PADOVA simulator while [66] has adopted an open-source version. In our work, we used the open-source version to ensure the reproducibility of our work. [64,67] are the most comparable to this study due to the similar challenging meal protocol used. Compared to [64], 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 [67]. The evaluation trials conducted were different in these studies, where [64] used a 10-day simulation for each subject without multiple repetitions, while [67] 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 [64,67] studies, while [66] reported a FR of 0%. However, they defined failures as simulations where glucose levels are $\leq$ 5 mg/dL. In contrast, in our work, we defined a much stricter FR where glucose levels $\leq$ 40 mg/dL or $\geq$ 600 mg/dL were considered failures. The lack of appropriate benchmarks in this field of research is a hurdle towards meaningful comparisons.

Limitations and Future Work. 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 analysed 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 min), in contrast to the one-step (5 min) 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 capacity of the algorithm to learn a more personalized 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

Table 6
Comparison of G2P2C with previous work on RL-based APSs.

Study	System	Performance†		Meal Protocol‡	Simulator
		Adults TIR	Adolescents TIR	Average daily CHO (*g*)	
Fox and Wiens [60][a]					
SAC	FA	Full cohorts not considered		Low CHO Meals (values not provided)	Simglucose 2018 (UVA/PADOVA 2008)
Fox et al. [66][b]					
SAC (RL-MA)	Hybrid*	77.12		Low CHO Meals (values not provided)	Simglucose 2018 (UVA/PADOVA 2008)
SAC (RL-Scratch)	FA	72.68			
Lee et al. [67]					
PPO	FA	89.30 ± 4.19	–	180	Custom platform based on UVA/PADOVA 2013
Lim et al. [64][c]					
SAC	Hybrid*	65.93 ± 17.29	62.20 ± 19.99	180 (157.5-202.5)	Custom platform based on UVA/PADOVA 2013
Our Work					
PPO	FA	69.12 ± 10.53	63.73 ± 13.95	180	Simglucose 2018 (UVA/PADOVA 2008)
G2P2C	FA	72.69 ± 9.53	64.33 ± 13.18		

*Hybrid system (require at least meal announcement).
†**Metrics**: [a]Experiments were conducted only on three handpicked subjects. The TIR results are presented as mean±std for Lee et al. [67], Lim et al. [64], and our work. The median TIR is presented for Fox et al. [66]. [b]Results presented as the median TIR for all cohorts. This is the only RL-based study with the Child cohort.
‡**Meal Protocol**: [a,b] Protocols with low CHO meals were considered (values not provided), [c]Meal intake information required, scenario with multiple small meals is used ($40g$, $50g$, $20g$, $50g$, $20g$ ±12.5%).
Acronyms: APS: Artificial Pancreas Systems, CHO: Carbohydrate, FA: Fully Autonomous, G2P2C: Glucose Control by Glucose Prediction and Planning, MA: Meal Announcement, PPO: Proximal Policy Optimization, RL: Reinforcement Learning, SAC: Soft Actor Critic.

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. Furthermore, this study was limited to the adolescent and adult cohorts of the simulator due to compute resource limitations. In our future work, we aim to extend the proposed method to the child cohort.

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 (https://capsml.com/, Appendix F) of G2P2C for users to experiment with the algorithm and compare its performance with 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. We empirically demonstrated that G2P2C improves TIR performance and safety compared to the benchmarked state-of-the-art RL algorithms while showing clinically promising results. To facilitate the development of RL-based APSs, we open-source the codebase of G2P2C (https://github.com/chirathyh/G2P2C) and provide an online demonstration tool for G2P2C (https://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 APSs.

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: https://capsml.com/

CRediT authorship contribution statement

Chirath Hettiarachchi: Methodology, Investigation, Software, Visualization, Validation, Writing – original draft, Writing – review & editing. **Nicolo Malagutti:** Methodology, Writing – original draft, Writing – review & editing. **Christopher J. Nolan:** Funding acquisition, Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Hanna Suominen:** Funding acquisition, Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Elena Daskalaki:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors report no competing interest.

Data availability

The code base of the research and an online demonstration tool is publicly release under the MIT license.

Acknowledgements

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 personalized health technologies and solutions in collaboration with patients, clinicians, and healthcare providers. This work was also supported by computational resources provided by the Australian Government through the National Computational Infrastructure (NCI Australia) under the ANU Merit Allocation Scheme. 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.

Appendix A. Neural network architectures

G2P2C. The feature extractor modules E^{Π} and E^{V} each consisted of a single-layer Long Short-Term Memory (LSTM) network [76] with 16 hidden units. 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 [78] and optimized using the Adam optimizer [90]. ReLU activation functions were used in the two networks, while the final layer of the π module used *tanh* and sigmoid activation functions to predict the mean and standard deviation of the policy. The final action was clipped to $[-1, 1]$. The M^{Π}, and M^{V} modules used *tanh* and softplus activation functions to predict the mean and standard deviation of the glucose prediction respectively.

The architecture of **A2C & PPO** algorithms were similar to G2P2C, with the only difference of having no M^{Π}, and M^{V} modules present. The **SAC** algorithm used a similar Actor Network (Π_θ). However the gaussian policy implemented was unbounded and an invertible squashing function used as presented in the original work of Haarnoja et al. [36]. The Critic-Network (V_ϕ) was modified to have two Q-value modules and two target modules. Each of these modules had dense layers similar to the v modules (see Fig. 6).

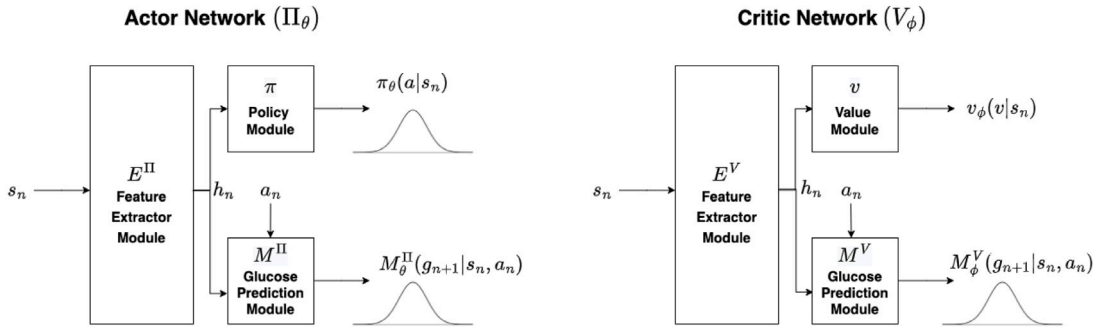


Fig. 6. Schematic diagram of actor and critic networks.

Appendix B. Hyperparameters

The PPO and A2C algorithms used an appropriate subset of the hyperparameters used for G2P2C (see Table 7).

Table 7
Hyperparameters for candidate RL algorithms.

Hyperparameter	Symbol	Value
Shared		
Sample time		5 min
Glucose sensor		Guardian RT
Insulin pump		Insulet
Augmented state history	k	12 (1-h)
Total number of steps	I_{total}	800,000
Optimizer		Adam [90]
G2P2C		
Batch size of policy and value update		1,024
Batch size of model learning and planning update		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 x 10^{-4}
PPO clip range	ϵ	0.1
Target Kullback–Leibler divergence (d_{KL}) threshold	d_{target}	0.01
Target glucose prediction error threshold	e_{target}	15 mg/dL
Planning trajectories	m	50 (per state)
Planning horizon	n_{plan}	6 (30-min)
SAC		
Replay buffer size		100,000
Discount		0.997
Batch size		256
Learning Rate		3 x 10^{-4}
Target smoothing coefficient		0.005
Target update interval		1
Gradient steps		1
Initial entropy coefficient		0.1

Appendix C. Training results of RL algorithms

See Figs. 7 and 8.

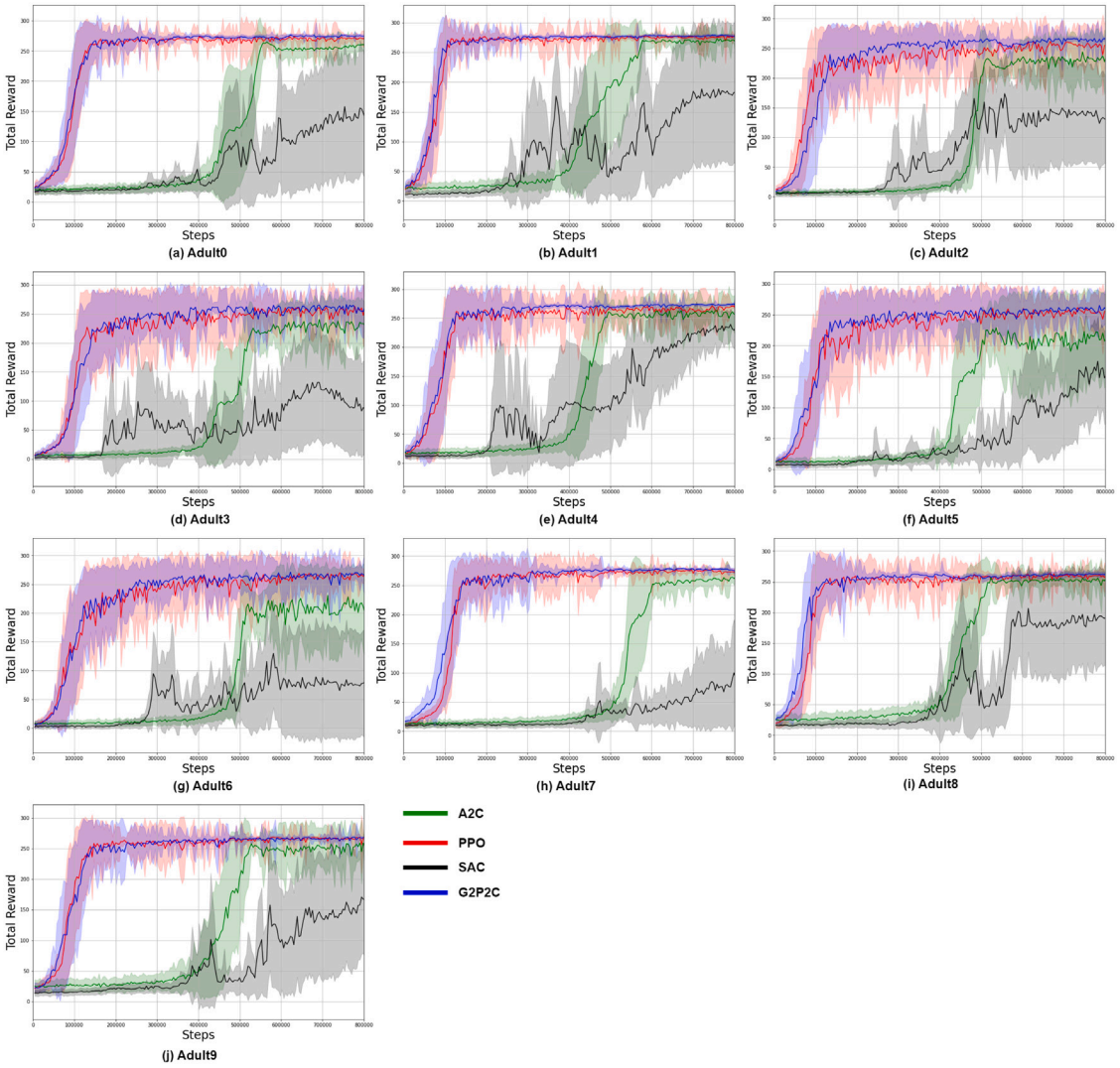


Fig. 7. Training curves for the *in-silico* Adult cohort. Mean and standard deviation of the total reward achieved against environment steps for evaluation simulations are presented.

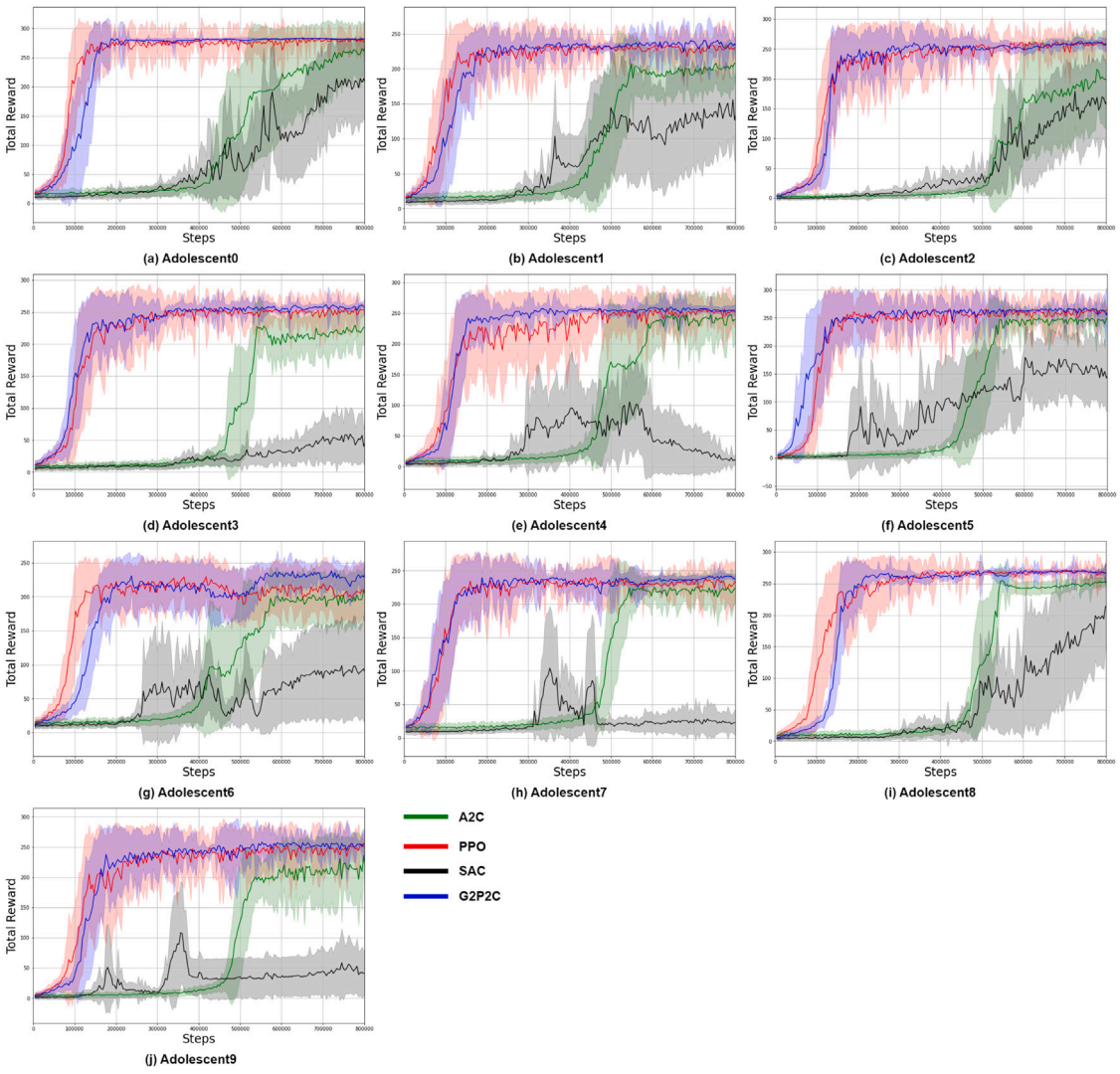


Fig. 8. Training curves for the *in-silico* Adolescent cohort. Mean and standard deviation of the total reward achieved against environment steps for evaluation simulations are presented.

Appendix D. Example glucose control simulations for candidate algorithms

The Fig. 9 represent simulations for a period of 24 h using candidate glucose control algorithms for subject Adolescent0. The glucose sensor measurements (blue) and meal events (red) are presented in the top, while the insulin action (green) of the candidate algorithm is illustrated by the bar chart below. The clinical objective is to improve the time spent in the normoglycemic range (Time in Range — TIR) highlighted in light green while minimizing the time spent in hypoglycemic/hyperglycemic ranges and avoiding the severe-hypoglycemic/severe-hyperglycemic ranges.

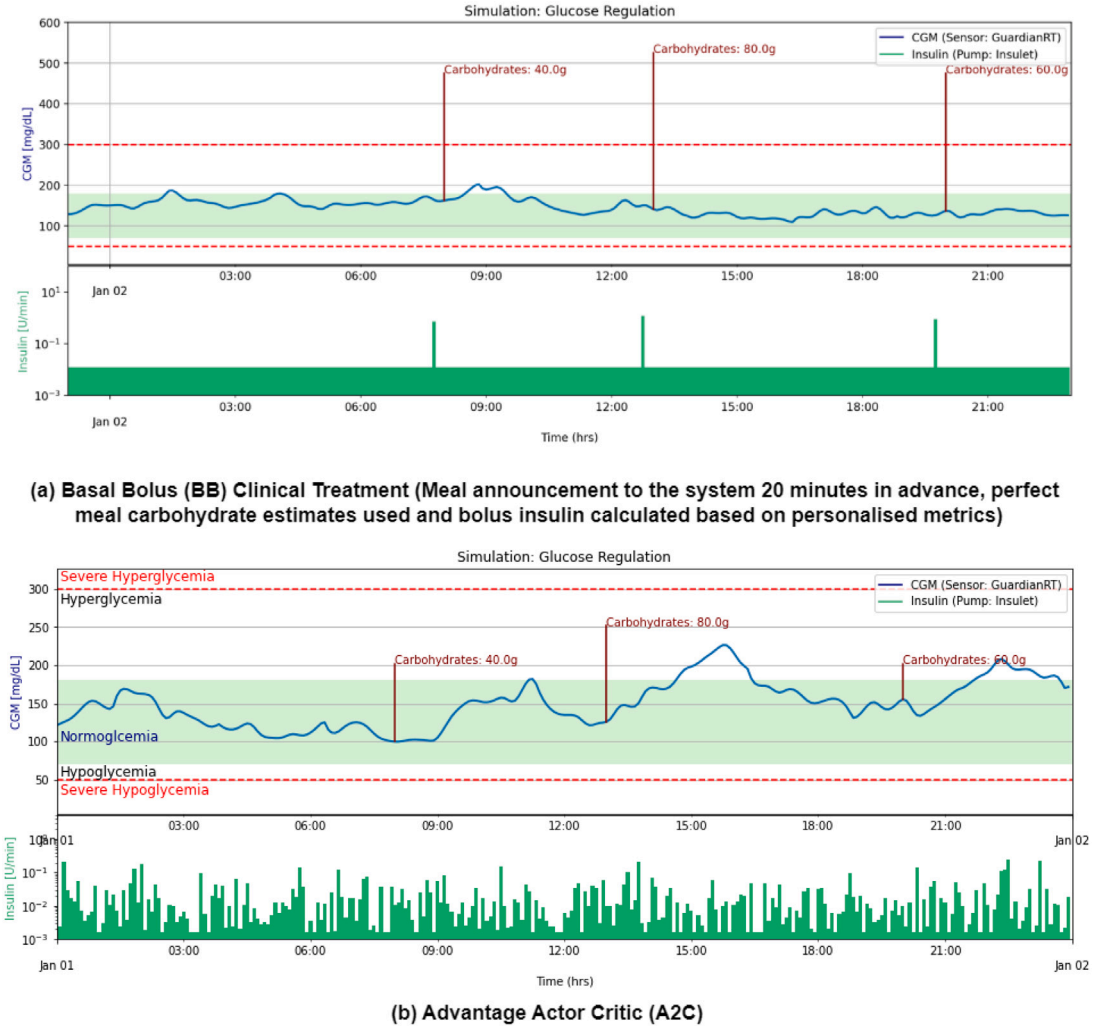
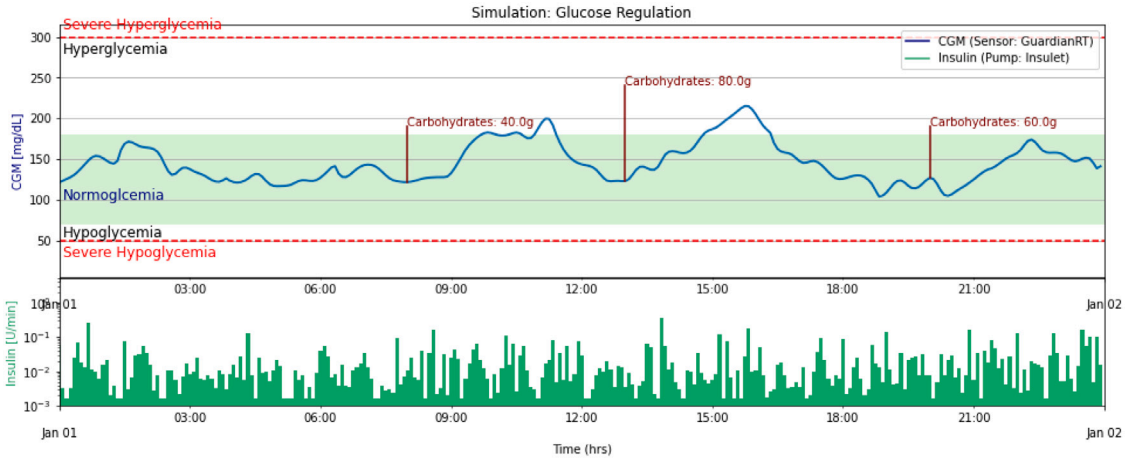
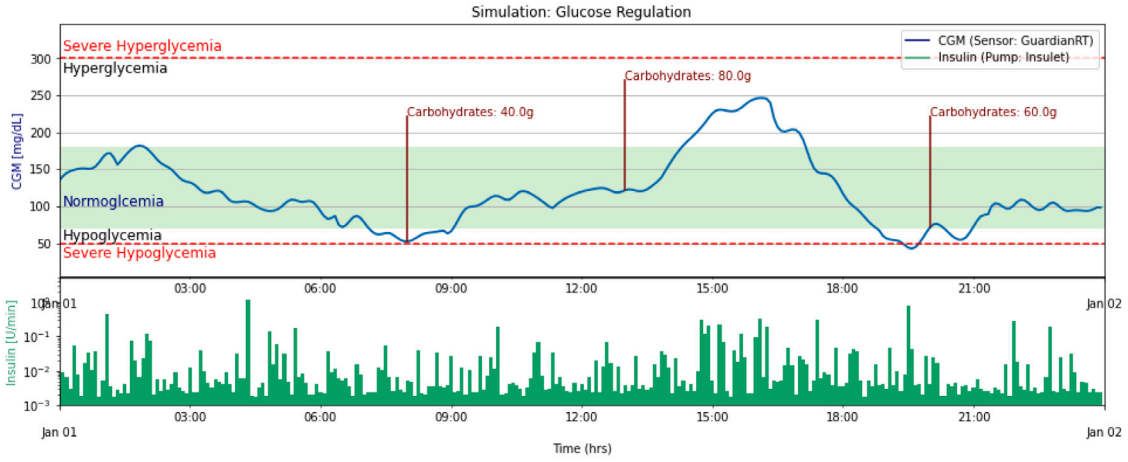


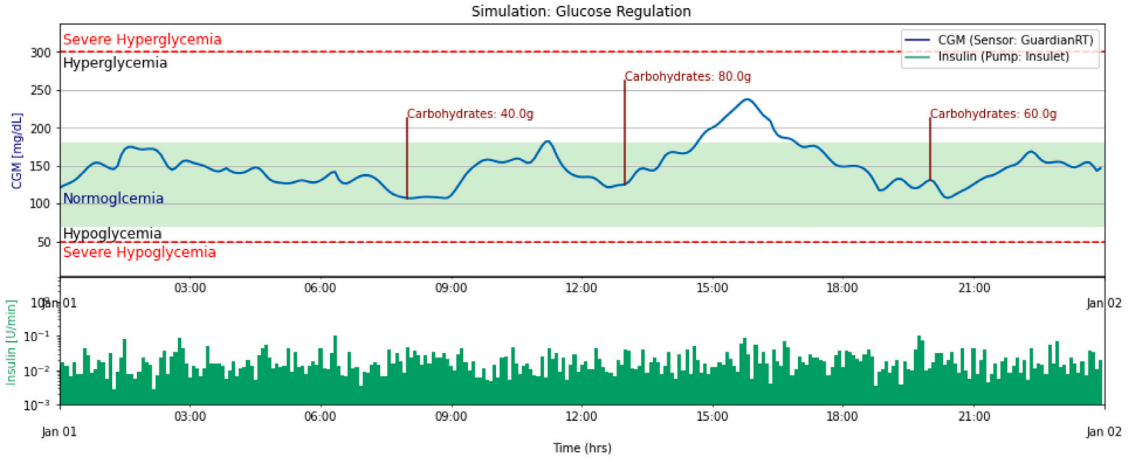
Fig. 9. Example glucose control simulations for candidate algorithms.



(c) Proximal Policy Optimisation (PPO)



(d) Soft Actor Critic (SAC)



(e) Glucose Control by Glucose Prediction & Planning (G2P2C)

Fig. 9. (*continued*).

Appendix E. Type 1 Diabetes simulators/models

See Table 8.

Table 8
A comparison of simulators available for glucose regulation.

Simulator (Study Reference)	FDA-Approved	Availability	Platform and ML Integration
UVA/PADOVA 2008 [12], 2013 [57], 2018 [91]	Yes	Commercial	MATLAB Limited ML Integration
Simglucose (2018) [56]	Implementation of FDA-approved UVA/PADOVA(2008).	Open-source	Python Leverage existing ML frameworks
Horvorka's Model (2010) [92]	No	Equations available for implementation	Not Applicable
T1D VPP (2019) [93]	No	Open-source	MATLAB Limited ML Integration
mGIPsim (2019) [94]	No	Private	Unknown

Acronyms: FDA: Food and Drug Administration, T1D VPP: Type 1 Diabetes Virtual Patient Population.

Appendix F. Tutorial: Understanding and evaluating G2P2C using CAPSML

CAPSML (https://capsml.com/) is a tool where you can define a custom meal protocol and try out different control algorithms. You can explore RL-based Glucose control algorithms and compare their performance with basal-bolus clinical treatment methods. A demonstration video of CAPSML can be accessed at https://youtu.be/JO5MkPCuqCw. The simulations can be conducted by following the steps highlighted below:

Step 1: The Simulator. The simulations are based on the UVA/PADOVA model, which is currently the only FDA-approved simulator. We use an opensource simulator Simglucose which uses the UVA/PADOVA 2008 model. The simulator includes models of commercially available insulin pumps and glucose sensors and allowed for the definition of different meal protocols and selection of T1D subjects for simulations (Note: The simulations are configured to use the Insulet pump and the GuardianRT glucose sensor, with a sample rate of 5 min) (see Fig. 10).

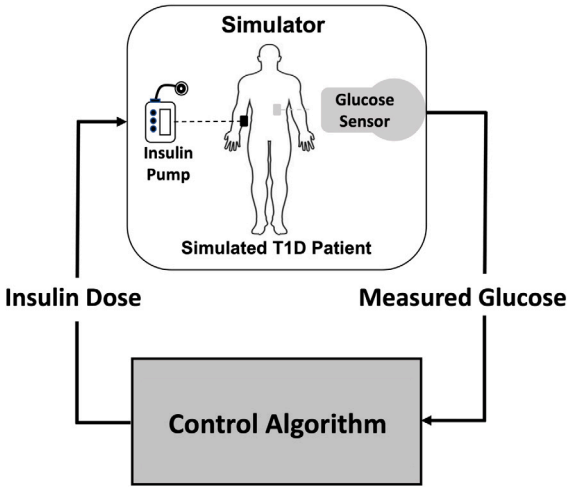


Fig. 10. Simulator Setup.

Step 2: Select an in-silico T1D Subject. The simulator comprises of a cohort of 30 in-silico subjects of three age categories (adults, adolescents, and children). The cohorts represents the patient variability found in a real T1D population, making meaningful statistical results available for the evaluation. You can select in-silico subjects from the Adult and Adolescent cohorts.

Step 3: Select a Control Algorithm. We have implemented an ideal basal — bolus control strategy with perfect meal information and 20 min ahead meal announcement named BBI and a basal bolus strategy with human error (BBHE). These strategies are based on patient-specific characteristics provided by the simulator (e.g., Total Daily Insulin, Carbohydrate Ratio). Basal insulin is continuously infused using the pump and correction and meal bolus doses are calculated automatically. We have implemented state-of-the-art reinforcement learning algorithms: A2C, PPO, and SAC. We also provide our novel algorithm named G2P2C (Glucose Control by Glucose Prediction and Planning). The RL-based algorithms (A2C, PPO, SAC, and, G2P2C) does not require meal announcements or carbohydrate estimation information.

Step 4: Setup a Meal Protocol. You can setup a custom meal protocol by selecting the meal time (24 h time format) and the carbohydrate content of the meals (grams). Currently the simulations support 3 meals (breakfast, lunch, and dinner).

Step 5: Run the Simulation. Once the parameters for the simulation is set, please press run. After the simulation concludes the simulated glucose trajectory will be displayed (see Fig. 11).

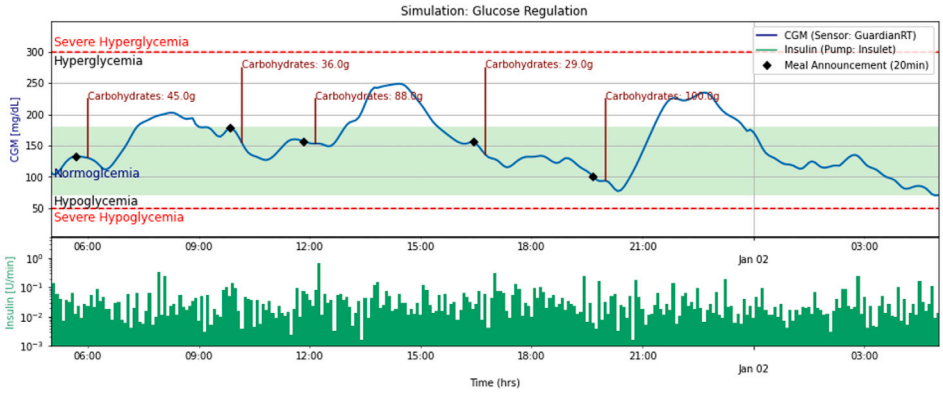


Fig. 11. Sample Glucose Trajectory.

References

[1] L.A. DiMeglio, C. Evans-Molina, R.A. Oram, Type 1 diabetes, Lancet 391 (10138) (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 Technol. 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, Exp. Rev. Med. Dev. 16 (10) (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 Obes. Metab. 23 (3) (2021) 655–660.

[5] G.M. Steil, Algorithms for a closed-loop artificial pancreas: The case for proportional-integral-derivative control, J. Diabetes Sci. Technol. 7 (6) (2013) 1621–1631.

[6] B.W. Bequette, Algorithms for a closed-loop artificial pancreas: the case for model predictive control, J. Diabetes Sci. Technol. 7 (6) (2013) 1632–1643.

[7] R.B. Shah, M. Patel, D.M. Maahs, V.N. Shah, Insulin delivery methods: Past, present and future, Int. J. Pharmaceut. Investig. 6 (1) (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, N. Engl. J. Med. 329 (14) (1993) 977–986.

[9] D. Slattery, S. Amiel, P. Choudhary, Optimal prandial timing of bolus insulin in diabetes management: A review, Diabetic Med. 35 (3) (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 Res. Clin. Pract. 99 (1) (2013) 19–23.

[11] C. Cobelli, E. Renard, B. Kovatchev, Artificial pancreas: past, present, future, Diabetes 60 (11) (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, J. Diabetes Sci. Technol. 3 (1) (2009) 44–55.

[13] A. Cinar, Multivariable adaptive artificial pancreas system in type 1 diabetes, Curr. Diabetes Rep. 17 (10) (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 (4) (2019) 800.

[15] J. Vliebergh, E. Lefever, C. Mathieu, Advances in newer basal and bolus insulins: Impact on type 1 diabetes, Curr. Opin. Endocrinol., Diabetes Obes. 28 (1) (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, Exp. Rev. Med. Dev. 10 (5) (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, Comput. Methods Programs Biomed. 109 (2) (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, 2018, arXiv preprint arXiv:1802.10440.

[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 (7839) (2020) 604–609.

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

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

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

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

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

[27] 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.

[28] N. Brew-Sam, A. Parkinson, M. Chhabra, A. Henschke, E. Brown, L. Pedley, E. Pedley, K. Hannan, K. Brown, K. Wright, et al., Toward diabetes device development that is mindful to the needs of Young people living with type 1 diabetes: A data-and theory-driven qualitative study, JMIR Diabetes 8 (1) (2023) e43377.

[29] 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.

[30] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, Y. Tassa, Safe exploration in continuous action spaces, 2018, arXiv preprint arXiv:1801.08757.

[31] J. Achiam, D. Held, A. Tamar, P. Abbeel, Constrained policy optimization, in: International Conference on Machine Learning, PMLR, 2017, pp. 22–31.

[32] W. Saunders, G. Sastry, A. Stuhlmueller, O. Evans, Trial without error: Towards safe reinforcement learning via human intervention, 2017, arXiv preprint arXiv:1707.05173.

[33] FDA, The content of investigational device exemption (IDE) and premarket approval (PMA) applications for artificial pancreas device systems, Silver Spring (2012).

[34] D. Nathan, S. Genuth, J. Lachin, P. Cleary, O. Crofford, M. Davis, L. Rand, C. Siebert, et al., The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus, New Engl. J. Med. 329 (14) (1993) 977–986.

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

[36] 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, 2018, arXiv preprint arXiv:1812.05905.

[37] 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.

[38] 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, 2019, arXiv preprint arXiv:1910.07113.

[39] 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.

[40] 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.

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

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

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

[44] R.S. Sutton, Dyna, an integrated architecture for learning, planning, and reacting, ACM Sigart Bull. 2 (4) (1991) 160–163.

[45] 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.

[46] M. Andrychowicz, A. Raichuk, P. Stanczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, O. Bachem, What matters for on-policy deep actor-critic methods? A large-scale study, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021, URL https://openreview.net/forum?id=nIAxjsniDzg.

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

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

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

[50] 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.

[51] A.X. Lee, A. Nagabandi, P. Abbeel, S. Levine, Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model, Adv. Neural Inf. Process. Syst. 33 (2020) 741–752.

[52] T. Anthony, Z. Tian, D. Barber, Thinking fast and slow with deep learning and tree search, Adv. Neural Inf. Process. Syst. 30 (2017).

[53] 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.

[54] 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 J. Biomed. Health Inf. 23 (6) (2018) 2633–2641.

[55] 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.

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

[57] C.D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, C. Cobelli, The UVA/PADOVA type 1 diabetes simulator: New features, J. Diabetes Sci. Technol. 8 (1) (2014) 26–34.

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

[59] T. Zhu, K. Li, P. Herrero, P. Georgiou, Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation, IEEE J. Biomed. Health Inf. 25 (4) (2020) 1223–1232.

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

[61] 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.

[62] 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, Comput. Math. Methods Med. 2018 (2018).

[63] 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.

[64] 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.

[65] M. Tejedor, A.Z. Woldaregay, F. Godtliebsen, Reinforcement learning application in diabetes blood glucose control: A systematic review, Artif. Intell. Med. 104 (2020) 101836.

[66] 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.

[67] 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 J. Biomed. Health Inf. 25 (2) (2020) 536–546.

[68] H. Emerson, M. Guy, R. McConville, Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes, J. Biomed. Inform. 142 (2023) 104376.

[69] X. Sun, M. Rashid, N. Hobbs, R. Brandt, M.R. Askari, A. Cinar, Incorporating prior information in adaptive model predictive control for multivariable artificial pancreas systems, J. Diabetes Sci. Technol. 16 (1) (2022) 19–28.

[70] M.R. Askari, I. Hajizadeh, M. Rashid, N. Hobbs, V.M. Zavala, A. Cinar, Adaptive-learning model predictive control for complex physiological systems: Automated insulin delivery in diabetes, Annu. Rev. Control 50 (2020) 1–12.

[71] K.-L.A. Yau, Y.-W. Chong, X. Fan, C. Wu, Y. Saleem, P.-C. Lim, Reinforcement learning models and algorithms for diabetes management, IEEE Access 11 (2023) 28391–28415.

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

[73] 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. 557–570.

[74] A. Mackey, E. Furey, Artificial pancreas control for diabetes using TD3 deep reinforcement learning, in: 2022 33rd Irish Signals and Systems Conference (ISSC), IEEE, 2022, pp. 1–6.

[75] 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 Technol. Therapeut. 7 (6) (2005) 849–862.

[76] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[77] L. Ljung, C. Andersson, K. Tiels, T.B. Schön, Deep learning and system identification, IFAC-PapersOnLine 53 (2) (2020) 1175–1181.

[78] 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, Adv. Neural Inf. Process. Syst. 32 (2019) 8026–8037.

[79] 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, N. Engl. J. Med. 363 (4) (2010) 311–320.

[80] C. Roversi, M. Vettoretti, S. Del Favero, A. Facchinetti, G. Sparacino, H.-R. Consortium, Modeling carbohydrate counting error in type 1 diabetes management, Diabetes Technol. Therapeut. 22 (10) (2020) 749–759.

[81] J. Walsh, R. Roberts, T. Bailey, Guidelines for optimal bolus calculator settings in adults, J. Diabetes Sci. Technol. 5 (1) (2011) 129–135.

[82] T. Battelino, T. Danne, R.M. Bergenstal, S.A. Amiel, R. Beck, T. Biester, E. Bosi, B.A. Buckingham, W.T. Cefalu, K.L. Close, et al., Clinical targets for continuous glucose monitoring data interpretation: Recommendations from the international consensus on time in range, Diabetes Care 42 (8) (2019) 1593–1603, http://dx.doi.org/10.2337/dci19-0028.

[83] S. Kalra, J.J. Mukherjee, S. Venkataraman, G. Bantwal, S. Shaikh, B. Saboo, A.K. Das, A. Ramachandran, Hypoglycemia: The neglected complication, Indian J. Endocrinol. Metabol. 17 (5) (2013) 819, http://dx.doi.org/10.4103/2230-8210.117219.

[84] G.D. Stoner, Hyperosmolar hyperglycemic state, Am. Family Phys. 71 (9) (2005) 1723, URL https://www.aafp.org/pubs/afp/issues/2017/1201/p729.html.

[85] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (3/4) (1965) 591–611.

[86] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60.

[87] C.O. Fritz, P.E. Morris, J.J. Richler, Effect size estimates: current use, calculations, and interpretation, J. Exp. Psychol. [Gen.] 141 (1) (2012) 2.

[88] L. Li, MATLAB User Manual, Matlab, Natick, MA, USA, 2001.

[89] G. vanRossum, F.L. Drake, Python Reference Manual, Python Software Foundation, Amsterdam, Netherlands, 2010.

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

[91] R. Visentin, E. Campos-Náñez, M. Schiavon, D. Lv, M. Vettoretti, M. Breton, B.P. Kovatchev, C. Dalla Man, C. Cobelli, The UVA/Padova type 1 diabetes simulator goes from single meal to single day, J. Diabetes Sci. Technol. 12 (2) (2018) 273–281, http://dx.doi.org/10.1177/1932296818757747.

[92] M.E. Wilinska, L.J. Chassin, C.L. Acerini, J.M. Allen, D.B. Dunger, R. Hovorka, Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes, J. Diabetes Sci. Technol. 4 (1) (2010) 132–144, http://dx.doi.org/10.1177/193229681000400117.

[93] N. Resalat, J. El Youssef, N. Tyler, J. Castle, P.G. Jacobs, A statistical virtual patient population for the glucoregulatory system in type 1 diabetes with integrated exercise model, in: P. Palumbo (Ed.), PLOS ONE 14 (7) (2019) e0217301, http://dx.doi.org/10.1371/journal.pone.0217301, URL http://dx.plos.org/10.1371/journal.pone.0217301.

[94] M. Rashid, S. Samadi, M. Sevil, I. Hajizadeh, P. Kolodziej, N. Hobbs, Z. Maloney, R. Brandt, J. Feng, M. Park, L. Quinn, A. Cinar, Simulation software for assessment of nonlinear and adaptive multivariable control algorithms: Glucose–Insulin dynamics in type 1 diabetes, Comput. Chem. Eng. 130 (2019) http://dx.doi.org/10.1016/j.compchemeng.2019.106565.