A Quantitative Comparison of Deep Reinforcement Learning Algorithms for Type 1 Diabetes Control

Federico BALDISSERI[1,*], Mohab M. H. ATANASIOUS[1], Valentina BECCHETTI[1], Antonio DI PAOLA[1,2], Giada LOPS[2], Danilo MENEGATTI[1], Andrea WRONA[1], Saverio MASCOLO[1], and Francesco DELLI PRISCOLI[1]

***Abstract*— Type 1 diabetes is a growing global health challenge. Standard clinical practice often relies on manual insulin injections, which can lead to suboptimal glucose regulation. Recent advancements have shifted focus towards Artificial Pancreas systems, integrating continuous glucose monitoring with automated insulin delivery. This work presents a quantitative comparison of four Deep Reinforcement Learning algorithms for autonomous glycemic regulation via insulin injection: DDPG, PPO, SAC, and TD3. The validation is conducted using the Hovorka model, in presence of uncertainties on number, time and amount of meals. Results show that all four controllers are able to maintain blood glucose levels within the target range. The TD3 algorithm outperforms the others in terms of several key performance indicators such as time in range, time in hypo/hyperglycemia and total insulin usage, while also exhibiting fewer hyperglycemic episodes compared to prior works in academic literature.**

***Index Terms*— Type 1 Diabetes, Insulin, Deep Reinforcement Learning, DDPG, PPO, TD3, SAC.**

I. INTRODUCTION

Diabetes is currently one of the major concerns in healthcare, since it affects approximately 420 million people in the world, and this number is expected to double before 2030 [1]. Type 1 diabetes (T1D) is a chronic disease, characterized by the destruction of pancreatic beta cells due to an autoimmune response [2]. This leads to complete dependency on endogenous insulin injection in order to prevent hyperglycemia, that leads to cardiovascular diseases and blindness, and hypoglycemia, that leads to coma and ultimately death. A safe and effective glycemic regulation keeps blood glucose concentration in the healthy range $70 - 180$ mg/dl. Patients still need to try and perform such regulation by estimating and administrating insulin by themselves: this inherently yields risks, due to over/under estimation of meal intake, and unpredicted physiological variations, among others. An attempt to overcome these risks led to the development of the Artificial Pancreas (AP) [3], that is a medical device composed of a continuous glucose monitoring system, an insulin pump and a closed-loop control algorithm which, based on glycemic readings, adjusts the insulin infusion in order to maintain blood glucose concentration in normoglycemia. Several control approaches have been proposed for AP systems. The Proportional-Integral-Derivative (PID) algorithm calculates the insulin dosage based on the weighted sum of its PID terms [4]; however, it requires precise parameter tuning to manage individual variability effectively and the algorithm often overestimates insulin needs during large post-prandial peaks, increasing the risk of hypoglycemia [4]. Model Predictive Control (MPC) optimizes the insulin dosage via a receding horizon approach based on a known or estimated system model to predict the future system outputs (i.e., glycemic levels) over a finite prediction horizon. At each step, MPC selects the optimal sequence of control actions (i.e., the insulin dosages) over the horizon, by applying only the first action. The process is then repeated, shifting the prediction horizon forward [5]. Both linear [6] and nonlinear [7] approaches have been used in a number of studies to investigate MPC-based blood glucose regulation, proving the latter to operate reliably even when there are small random disturbances like exercise, stress, and exhaustion [8]; in both cases a reliable model of the system is required together with a high computational burden caused by real-time optimization [9]. Valuable results [10] have also been achieved through Fuzzy-logic based AP, which rely on expert-developed rule-based policies that explicitly state when and how much insulin to infuse as needed. This approach holds great potential for supervising other training-based control algorithms and enhancing safety, but struggles handling unexpected emergency conditions.

Recently, Reinforcement Learning (RL) is gaining prominence in various healthcare applications, including diabetes management. It has been proposed as a valid framework for fully automated closed-loop insulin delivery systems [11]–[14]. RL overcomes the problem of the need of a reliable model of the system, whose precision is crucial especially in a safety-critic context like the one of T1D control.

To the best of the authors' knowledge, this is the first work performing a quantitative comparison of several different Deep Reinforcement Learning (DRL) algorithms in the context of AP.

The main contributions of this work are summarized as follows:

[1]Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy.

[2]Department of Electrical and Information Engineering, Polytechnic University of Bari, Italy.

*Corresponding author. Email: baldisseri@diag.uniroma1.it

This work has been partially co-funded by the EU and the Swiss State Secretariat for Education, Research and Innovation, in the frame of the SHIELD (grant Nº 101156751) project. Moreover, it has been partially carried out in the framework of the research project "Automatic Control Algorithms and AI for the Treatment of Type 1 Diabetes" (n. AR124190738184E0), supported by Sapienza University of Rome.

- Four continuous controllers based on DRL are developed for the problem of autonomous insulin regulation for patients suffering from T1D;
- The proposed controllers are validated through several in-silico simulations on the Hovorka model, incorporating noisy meal disturbances to realistically represent variability of meal intake;
- The performances of the proposed controllers are compared in terms of key diabetes control metrics and training time.

The remainder of the manuscript is organized as follows: Section II introduces the proposed control methodologies; Section III describes the Hovorka dynamical model of a patient suffering from T1D; Section IV presents and discusses the in-silico simulations performed in order to validate the proposed controllers; finally, Section V draws conclusions and outlines future research directions.

II. Recalls on Markov Decision Processes and Deep Reinforcement Learning

Reinforcement Learning (RL) is a subfield of Machine Learning focusing on training intelligent agents to perform actions over a dynamical environment, based on observations and rewards [15]. The environment is formalized in the form of a Markov Decision Process (MDP), a model for sequential decision making when outcomes are uncertain. Said model satisfies the Markov property, i.e., its future evolution is independent of its history.

Formally, an MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where:

- $\mathcal{S}$ is the set of states.
- $\mathcal{A}$ is the set of actions.
- $P = P(s'|s,a)$ is the probability that the system transitions from state s to state s' upon selecting action a.
- $R(s,a,s')$ is the instantaneous reward received by the agent after transitioning from state s to s' as a result of executing action a.
- $\gamma \in [0,1)$ is the discount factor, which determines the agent's preference between immediate rewards ($\gamma \approx 0$) and future rewards ($\gamma \approx 1$).

The goal in an MDP is to learn an optimal policy $\pi : \mathcal{S} \to \mathcal{A}$ to maximize the expected discounted sum of instantaneous rewards over a potentially infinite horizon, known as the state-action value function $Q(s,a)$:

$$Q(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s, a_0 = a\right]. \tag{1}$$

RL algorithms can be exploited to solve MDP-related problems, without assuming the complete knowledge of P. DRL extends traditional RL by leveraging deep neural networks to efficiently approximate complex mappings between high-dimensional states, actions, and rewards. This enables more effective learning of optimal control policies, particularly in environments with intricate and large-scale state spaces [16]. In the following, some recalls of the four DRL procedures involved in this study are provided.

A. DDPG

Deep Deterministic Policy Gradient (DDPG) is an off-policy reinforcement learning algorithm designed for continuous state and action spaces. Introduced in [17], DDPG combines elements of Deep Q Networks (which approximate the Q function in (1) using neural networks) and Policy Gradient Methods (which directly optimize the control policy). Unlike stochastic approaches, DDPG exhibits a deterministic nature after training, as no randomness is involved in action selection.

The algorithm employs an Actor-Critic framework, where the actor μ_θ selects actions, and the critic Q_ϕ evaluates them. To enhance training stability, it utilizes also a target actor $\mu_{\theta'}$ and target critic $Q_{\phi'}$ networks. Learning is carried out using past experiences stored in a Replay Buffer $\mathcal{D}$, minimizing the loss functions of the critic network

$$L = \frac{1}{N}\sum_i \Big[Q_\phi(s_i,a_i) - \\ + \Big(r_i + \gamma Q_{\phi'}\big(s_i', \mu_{\theta'}(s_i')\big)\Big)\Big]^2, \tag{2}$$

and updating the actor policy using the sampled policy gradient

$$\nabla_\theta J \approx \frac{1}{N}\sum_i \nabla_a Q_\phi \nabla_\theta \mu_\theta \tag{3}$$

where θ, θ', ϕ and ϕ' represent the weights of the actor, target actor, critic, and target critic networks, respectively. By leveraging stored experience instead of directly relying on the current policy, DDPG improves learning efficiency and stability.

B. PPO

Proximal Policy Optimization (PPO) is a policy gradient method designed for stable and efficient reinforcement learning [18]. It employs a clipped surrogate objective function to constrain policy updates, preventing drastic parameter changes and improving stability. PPO estimates the advantage function, defined as $A_t = Q_t - V_t$, to guide policy updates. The optimization problem maximizes:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right], \tag{4}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and ϵ is a hyperparameter controlling the clipping range $[1-\epsilon, 1+\epsilon]$ to avoid large policy updates. PPO adopts an Actor-Critic framework, where the actor updates the policy and the critic estimates the state-value function. The final objective function includes an entropy term for exploration and a squared-error loss for value estimation:

$$L_{\text{PPO}} = L_{\text{CLIP}} - c_1 L_{\text{VF}} + c_2 S, \tag{5}$$

where L_{VF} is the value function loss, S is the entropy bonus, and c_1 and c_2 are hyperparameters that balance the importance of the value function loss and entropy bonus.

C. TD3

The Twin Delayed Deep-Deterministic Policy Gradient (TD3) algorithm was firstly introduced in [19], to overcome one of the main disadvantages of DDPG, i.e., the learned ability to overestimate Q values, leading to significant errors in the Q function, until policy breaking. TD3 learns two critic functions, Q_{ϕ_1} and Q_{ϕ_2} – from this the term "twin" – setting the minimum Q-value of these two as a target for the Bellman error loss function, i.e.:

$$y = r + \gamma \min_{i=1,2} Q_{\phi'_i}(s', a'(s')), \tag{6}$$

where $a'(s')$ is the target action generated with smoothing and $Q_{\phi'_i}$ is one of the two target critic networks. Moreover, this method, by delaying the update of policy networks, allows the Q-functions to stabilize first and to reduce the likelihood of high variance and instability in the policy. Another relevant failure mode regards DDPG target policy which exploits sharp peaks in the Q-function that may arise due to approximation errors; instead, in the TD3 framework, a Gaussian clipped noise is added to the target actions, while computing the Q-value targets, resulting in a smoother Q-function.

D. SAC

Soft Actor-Critic (SAC) is an off-policy RL algorithm designed for continuous control tasks [20]. It extends discrete RL by incorporating entropy maximization. The actor network outputs a squashed Gaussian policy:

$$\pi_\theta(s) = \tanh(\mu_\theta(s) + \sigma_\theta(s) \cdot \epsilon), \quad \epsilon \sim \mathcal{N}(0,1) \tag{7}$$

which bounds actions between $[-1, 1]$. The critic network learns the Q-values by minimizing the following loss:

$$L_Q = \mathbb{E}\left[(Q_\phi(s,a) - y)^2\right], \tag{8}$$

where the target is given by:

$$y = r + \gamma(\min_{i=1,2} Q_{\phi'_i}(s', a') + \alpha H(\pi)). \tag{9}$$

$H(\pi)$ represents the so-called entropy bonus, that encourages exploration, while the parameter α controls the exploration-exploitation trade-off.

SAC's instability with deep networks is linked to gradient explosions when the actor updates through the critic. Spectral normalization stabilizes training, enabling larger networks and improving performance [20].

III. System Mathematical Modeling

The Hovorka model effectively describes the blood glucose-insulin regulatory system [21].

The glucose absorption subsystem is given by:

$$\begin{aligned} \dot{D}_1 &= A_G D - \frac{D_1}{\tau_G}, \\ \dot{D}_2 &= \frac{D_1}{\tau_G} - U_G, \end{aligned} \tag{10}$$

where $U_G = D_2/\tau_G$.

TABLE I
Variables of the Hovorka Model

Variable	Description	Unit
Q_1	Amount of glucose in main blood stream	mmol
Q_2	Amount of glucose in peripheral tissues	mmol
S_1	Amount of insulin in compartment 1	mU
S_2	Amount of insulin in compartment 2	mU
I	Plasma insulin concentration	mU/L
x_1	Remote effect of insulin on glucose distribution	–
x_2	Remote effect of insulin on glucose disposal	–
x_3	Remote effect of insulin on endogenous glucose production	–
D	CHO eating rate (disturbance)	[mmol/min]
u	Insulin injection (control action)	U/min
G	Blood glucose conc. (measurable output)	mg/dL

The glucose subsystem is given by:

$$\begin{aligned} \dot{Q}_1 &= -(F_{01c} + F_R) - x_1 Q_1 + k_{12} Q_2 + U_G \\ &\quad + \max(\text{EGP}_0(1 - x_3), 0), \\ \dot{Q}_2 &= x_1 Q_1 - (k_{12} + x_2) Q_2, \end{aligned} \tag{11}$$

where the measurable output of the system, namely the blood glucose concentration, is given by:

$$y = G = \frac{Q_1}{V_G}, \tag{12}$$

and $F_{01c} = \frac{F_{01}}{0.85} \frac{G}{G+1}$, and

$$F_R = \begin{cases} \rho(G - G_\tau) V_G, & \text{if } G \geq G_\tau, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The insulin subsystem is given by:

$$\begin{aligned} \dot{S}_1 &= u - \frac{S_1}{\tau_I}, \\ \dot{S}_2 &= \frac{S_1}{\tau_I} - U_I, \\ \dot{S}_3 &= \frac{U_I}{V_I} - k_e I, \end{aligned} \tag{14}$$

where $U_I = S_2/\tau_I$.

The insulin action subsystem is given by:

$$\begin{aligned} \dot{x}_1 &= k_{b1} I - k_{a1} x_1, \\ \dot{x}_2 &= k_{b2} I - k_{a2} x_2, \\ \dot{x}_3 &= k_{b3} I - k_{a3} x_3. \end{aligned} \tag{15}$$

The variables and parameters of the model are illustrated in Table I and Table II, respectively.

Note that the mathematical model is unknown to the DRL agents, since it is only used as the (black-box) environment that generates outputs based on the given inputs.

This system translates to the MDP framework in the following way: the state space is $\mathcal{S} = \{G, \delta\}$, where δ is

TABLE II
PARAMETERS OF THE HOVORKA MODEL

Parameter	Description	Unit
G_b	Basal glucose plasma conc.	$\mathrm{mg/dl}$
G_τ	Renal glucose clearance threshold	$\mathrm{mg/dl}$
I_b	Basal insulin plasma conc.	$\mu\mathrm{U/ml}$
p_1	Indep. glucose disappearance rate	min^{-1}
p_2	Spontaneous glucose uptake	min^{-1}
p_3	Insulin-dep. glucose uptake	$\frac{\mathrm{ml}}{\mu\mathrm{U\,min}^2}$
δ	Pancreatic β-cellsrelease	$\frac{\mu\mathrm{U\,dl}}{\mathrm{ml\,mg\,min}^2}$
n	Insulin infusion	min^{-1}
V_I	Insulin distr. vol.	L
V_G	Glucose distr. vol.	L
F_{01}	Non-insulin-dependent glucose flux	$\mathrm{mmol/min}$
EGP_0	EGP to zero insulin conc.	$\mathrm{mmol/min}$
ρ	Renal glucose clearance capacity	–
S_{IT}	Insulin sensitivity of transport	$\mathrm{L/min \cdot mU}$
S_{ID}	Insulin sensitivity of disposal	$\mathrm{L/min \cdot mU}$
S_{IE}	Insulin sensitivity of EGP	$\mathrm{L/mU}$
τ_G	Time-to-maximum CHO absorption	min
τ_I	Time-to-maximum insulinabsorption	min
A_G	CHO bioavailability	–
k_{12}	Transfer rate	min^{-1}
k_{a1}	Deactivation insulin transport rate	min^{-1}
k_{b1}	Activation insulin transport rate	$\mathrm{L/(mU \cdot min)}$
k_{a2}	Deactivation insulin disposal rate	min^{-1}
k_{b2}	Activation insulin disposal rate	$\mathrm{L/(mU \cdot min)}$
k_{a3}	Deactivation insulin EGP rate	min^{-1}
k_{b3}	Activation insulin EGP rate	$\mathrm{L/(mU \cdot min)}$
k_e	Insulin elimination from plasma	min^{-1}

a binary term defined at step t as follows:

$$\delta_t = \begin{cases} 0, & \text{if } \big(G_t - G_{t-1} \ < 0.05 \ \wedge \ \delta_{t-1} = 0\big) \\ & \vee \ \big(G_t - G_{t-1} < 0\big), \\ 1, & \text{otherwise.} \end{cases} \tag{16}$$

The first condition encompasses two cases: either the blood glucose is decreasing, or it is slightly increasing (due to endogenous glucose production represented by the term EGP_0, not due to a meal). The second condition indicates that the blood glucose is increasing.

This rationale helps the RL agent differentiate between two cases: if blood glucose concentration is rising sharply, it means that the patient is eating; on the other hand, if the blood glucose concentration is rising slowly, it means that the increase is due to endogenous glucose production.

The action space is $\mathcal{A} = \{u | u \in [0, 0.3] \ \mathrm{U/min}\}$, and the reward function is defined as follows:

$$R(G, \delta) = \begin{cases} -2, & \text{if } \delta = 1 \ \wedge \ G \geq 180, \\ 0.5, & \text{if } \delta = 0 \ \wedge \ G \geq 180, \\ -3, & \text{if } \delta = 0 \ \wedge \ G \leq 70, \\ 0.5, & \text{if } \delta = 1 \ \wedge \ G \leq 70, \\ 1, & \text{otherwise.} \end{cases} \tag{17}$$

Note that low positive rewards are given when blood glucose concentration is returning to normoglycemic values. Moreover, hypoglycemia is penalized slightly more than hyperglycemia, since it yields worse clinical consequences.

IV. SIMULATIONS AND RESULTS

In order to evaluate the performances of the proposed controllers, simulations of their effectiveness were performed on the Hovorka glucoregulatory model presented in Section III, integrated using the Dormand-Prince method [22], in a Python 3.10 environment using Tensorflow (V2.17.0) and Keras (V3.5.0), and a NVIDIA Tesla T4 GPU. The numerical values of parameters in Table II have been taken from [21].

A. Simulation Scenario

Meal disturbance is simulated as in [23]: it is assumed that throughout $34\mathrm{h}$ of simulation starting at 6:00 in the morning, five meals are consumed, containing respectively 50, 60 and 80 grams of carbohydrates in the first day, and 60 and 60 grams of carbohydrates in the second day. Initial conditions are randomly selected in the normoglycemic range $70 - 180 \ \mathrm{mg/dl}$, and to simulate variability in meal consumption, a noise with uniform distribution is introduced into various aspects of the eating schedule as follows:

- Number of meals: the possibility of skipping meals or having snacks is modeled as follows; the probabilities of each meal (breakfast, snack, lunch, afternoon snack, dinner) occurring are $[0.9, 0.15, 0.8, 0.25, 0.75]$, and each meal is associated with a binary variable, α, where $\alpha = 1$ if the meal occurs and $\alpha = 0$ if it does not occur. In addition, a constraint is introduced such that the sum of α corresponding to breakfast, lunch, and dinner must be ≥ 2; this means that no more than one meal can be skipped among breakfast, lunch, and dinner.
- Meal time: the time at which each meal occurs is subject to random variation, which is differentiated for each meal as follows: breakfast: $[-60, +60]$ minutes; snack: $[0, +60]$ minutes; lunch: $[0, +120]$ minutes; afternoon snack: $[0, +60]$ minutes; dinner: $[0, +120]$ minutes
- Carbohydrate amount in a meal: the amount of carbohydrates in each meal is subject to variability, which is differentiated for each meal as follows: breakfast: $[-15, +15]$ grams; snack: $[-5, +5]$ grams; lunch: $[-20, +20]$ grams; afternoon snack: $[-5, +5]$ grams; dinner: $[-20, +20]$ grams.

B. Quantitative Comparison of Performances

Each episode within the training phase of the four DRL algorithms counts for $T = 1440$ steps. The other training hyperparameters are selected as follows:

- DDPG: discount factor $\gamma = 0.89$, actor learning rate $\alpha = 1 \times 10^{-5}$, critic learning rate $\beta = 5 \times 10^{-5}$, batch size $B = 64$, number of training episodes $E = 150$. Both the actor and critic share the same neural architecture composed of three layers, the first two of 64 and 32 neurons with ReLU activation function, and the last one of one neuron only, with hyperbolic tangent activation function for the actor.
- PPO: discount factor $\gamma = 0.99$, learning rate for both networks $\alpha = 1 \times 10^{-4}$, batch size $B = 256$, clip range $\epsilon = 0.3$, rollout $= 2048$, number of training

episodes $E = 1000$. The architecture of both the actor and the critic network consists of two hidden layers, each comprising 256 neurons. The hidden layers employ a Tanh activation function. The actor's output layer uses a linear activation function, adjusted to the action space by a clipping function.

- TD3: discount factor $\gamma = 0.89$, actor learning rate $\alpha = 1 \times 10^{-4}$, critic learning rate $\beta = 3 \times 10^{-4}$, batch size $B = 1024$, *polyak* averaging coefficient used for updating the target network $\tau = 0.005$, policy delay set to 2, number of training episodes $E = 100$. The architecture of both the actor and the two critic networks consists of two hidden layers, each comprising 64 neurons. The hidden layers employ a ReLU activation function, whereas the output layer presents a tanh activation function.
- SAC: discount factor $\gamma = 0.99$, actor learning rate $\alpha = 1 \times 10^{-4}$, learning rate for both critics and target critics $\beta = 2 \times 10^{-4}$, *polyak* averaging coefficient for updating target Q-networks $\tau = 5 \times 10^{-4}$, fixed entropy regularization coefficient $\alpha_{er} = 0.2$, batch size $B = 512$, number of training episodes $E = 100$. The five networks share the same architecture with two hidden layers composed of 256 neurons and ReLU activation functions. The actor network's outputs, μ and σ, are produced using a linear activation function. The standard deviation σ is then clamped between a minimum value of e^{-20} and a maximum value of e^{2}.

Figure 1 depicts the evaluation of the four trained controllers on a random test scenario that was not involved in the training phase, showing the evolution over 34h of glycemia, control action, and meal disturbance. Note that all of them achieve satisfactory glycemic regulation. In particular, DPPG and TD3 agents yield less hypoglycemic episodes with respect to SAC and PPO agents.

Figure 2 shows the evaluation of the four trained controllers on 30 scenarios that are not involved in the training phase. The DDPG and TD3 agents still exhibit the best preservation of hypoglycemia.

The performance comparison of the proposed controllers is summarized in Table III. The TD3 controller achieves the most satisfactory glycemic regulation, yielding the highest value of the time in normoglycemic range and the lowest value of the time in hypoglycemia, while also reaching the highest minimum value in hypoglycemia. This is desirable, since hypoglycemia causes worse clinical conditions. Such results show lower overall hyperglycemic episodes with respect to previous works in literature [11], [12].

V. Conclusions

This work presented a quantitative comparison of four DRL agents, namely DDPG, PPO, SAC, and TD3 for automated T1D management. Such controllers were validated through in-silico simulations on the Hovorka model. A quantitative comparison of their performances highlighted TD3 as the one yielding the most satisfactory glycemic regulation, with respect to average time in range, average time in hypoglycemia, and average minimum glycemia.

Future works will tackle the limitations of this study: the controllers shall be assessed on various parametrizations of the given mathematical model, and even on a simulator, such as the Uva/Padova [24], in order to achieve robustness to model uncertainties and patient personalization. Moreover, the second component of the state may be investigated by using the actual numerical value of the derivative of the blood glucose concentration, thus highlighting the behavior of the glycemic regulation system of each specific patient.

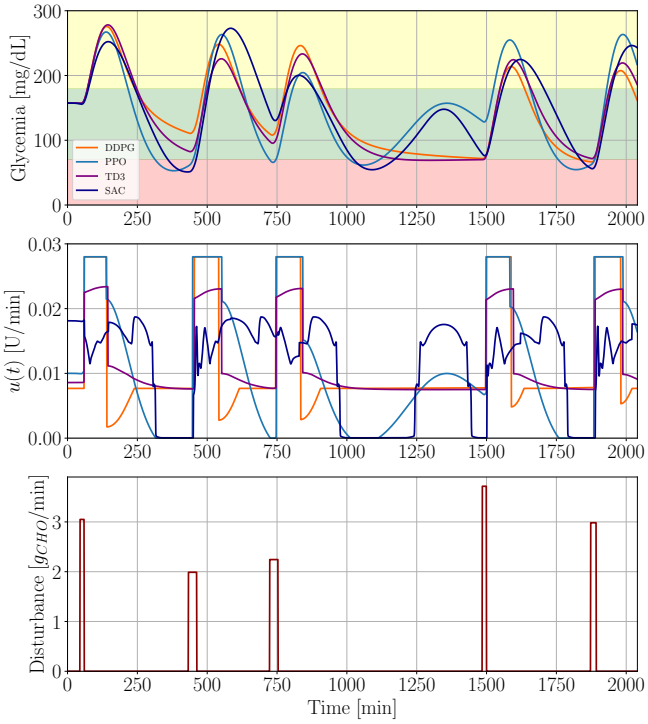

Fig. 1. Evaluation of the four controllers on a random test scenario over 34 hours. The figure includes: blood glucose concentration (first plot), control action (second plot), and meal disturbance (third plot).

TABLE III
KPIs comparison of the used control logics

KPI	DDPG	PPO	TD3	SAC
Avg time in range [%min]	63.84	53.91	**65.08**	55.72
Avg time in hyperglycemia [%min]	23.09	**17.70**	26.38	30.26
Avg time in hypoglycemia [%min]	13.07	28.39	**8.54**	14.02
Avg max glycemia [mg/dl]	257	**243**	265	267
Avg min glycemia [mg/dl]	66	38	**69**	53
Avg total injected insulin [U]	24.27	23.97	23.03	**22.32**
Training time [min]	**23.25**	58.23	24.96	32.97

References

[1] World Health Organization, "Diabetes." https://www.who.int/en/news-room/fact-sheets/detail/diabetes, 2023. Accessed: January 12th, 2025.

[2] M. A. Atkinson, G. S. Eisenbarth, and A. W. Michels, "Type 1 diabetes," *The Lancet*, vol. 383, no. 9911, pp. 69–82, 2014.

[3] E. Bekiari, K. Kitsios, H. Thabit, M. Tauschmann, E. Athanasiadou, T. Karagiannis, A.-B. Haidich, R. Hovorka, and A. Tsapas, "Artificial pancreas treatment for outpatients with type 1 diabetes: systematic review and meta-analysis," *bmj*, vol. 361, 2018.

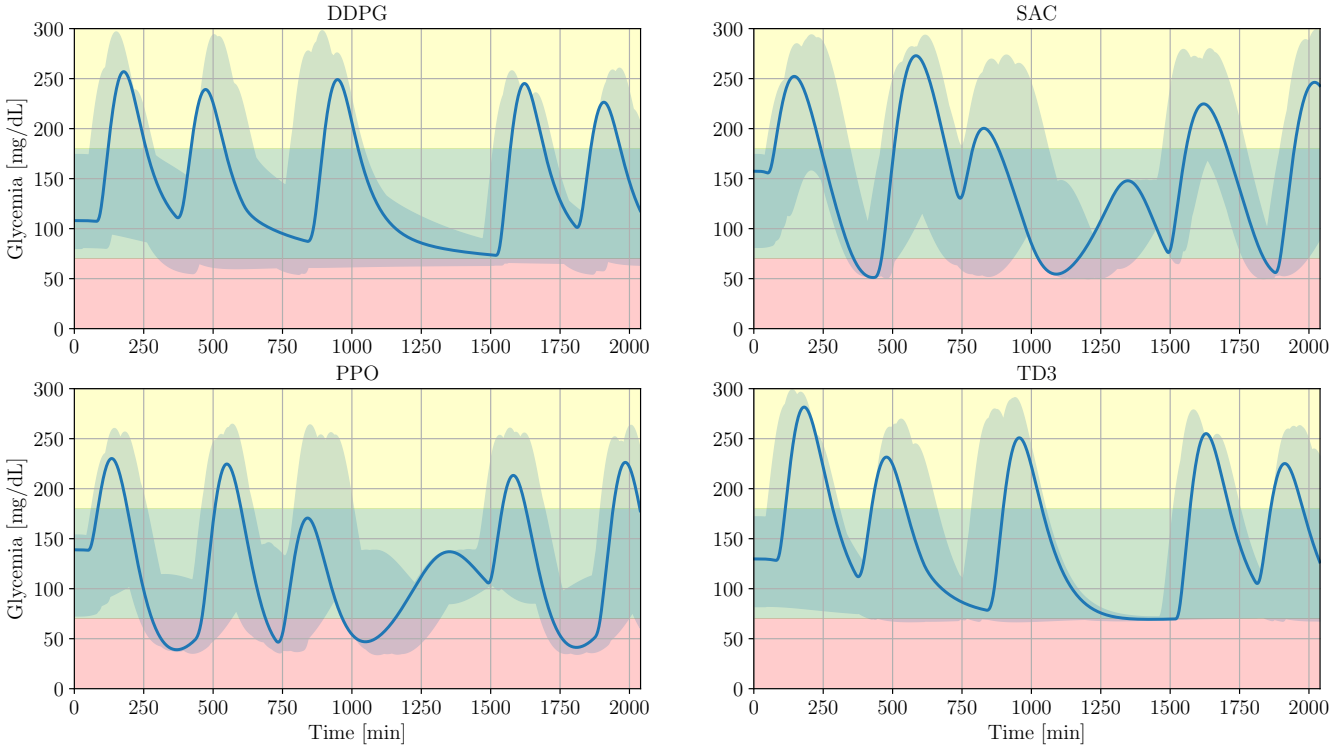

Fig. 2. Evaluation of the trained controllers on 30 scenarios lasting 34 hours. The blue solid lines represent the first evaluation scenario, while shaded blue regions represent the other scenarios. The green, yellow and red background areas represent normoglycemic, hyperglycemic and hypoglycemic regions, respectively.

[4] G. M. Steil, K. Rebrin, C. Darwin, F. Hariri, and M. F. Saad, "Feasibility of automating insulin delivery for the treatment of type 1 diabetes," *Diabetes*, vol. 55, no. 12, pp. 3344–3350, 2006.

[5] W. Ming, X. Guo, G. Zhang, Y. Liu, Y. Wang, H. Zhang, H. Liang, and Y. Yang, "Recent advances in the precision control strategy of artificial pancreas," *Medical & Biological Engineering & Computing*, vol. 62, pp. 1615–1638, June 2024.

[6] V. Becchetti, M. M. H. Atanasious, D. Menegatti, F. Baldisseri, and A. Giuseppi, "Dynamic mode decomposition for individualized model predictive control with application to type 1 diabetes," in *2024 32nd Mediterranean Conference on Control and Automation (MED)*, pp. 239–244, IEEE, 2024.

[7] A. Mirzaee, M. Dehghani, and M. Mohammadi, "A nonlinear mpc approach for blood glucose regulation in diabetic patients," in *Proceedings of the 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*, (Tabriz, Iran), pp. 1–5, IEEE, 2021.

[8] S. Parihar, P. Shah, R. Sekhar, and J. Lagoo, "Model predictive control and its role in biomedical therapeutic automation: A brief review," *Applied System Innovation*, vol. 5, no. 6, 2022.

[9] A. Cinar, "Automated insulin delivery algorithms," *Diabetes Spectrum*, vol. 32, pp. 209–214, 08 2019.

[10] E. Atlas, R. Nimri, S. Miller, E. A. Grunberg, and M. Phillip, "Md-logic artificial pancreas system: a pilot study in adults with type 1 diabetes," *Diabetes Care*, vol. 33, no. 5, pp. 1072–1076, 2010.

[11] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1223–1232, 2020.

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

[13] F. Baldisseri, D. Menegatti, and A. Wrona, "Deep deterministic policy gradient control of type 1 diabetes," in *2024 European Control Conference (ECC)*, pp. 868–873, IEEE, 2024.

[14] M. M. Atanasious, V. Becchetti, F. Baldisseri, D. Menegatti, and A. Wrona, "Deep reinforcement learning control of type-1 diabetes with cross-patient generalization," in *2024 32nd Mediterranean Conference on Control and Automation (MED)*, pp. 221–226, IEEE, 2024.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2019.

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

[19] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, PMLR, 10–15 Jul 2018.

[21] R. Hovorka, V. Canonico, L. J. Chassin, U. Haueter, M. Massi-Benedetti, M. O. Federici, T. R. Pieber, H. C. Schaller, L. Schaupp, T. Vering, *et al.*, "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes," *Physiological Measurement*, vol. 25, no. 4, p. 905, 2004.

[22] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[23] M. Messori, G. P. Incremona, C. Cobelli, and L. Magni, "Individualized model predictive control for the artificial pancreas: In silico evaluation of closed-loop glucose control," *IEEE Control Systems Magazine*, vol. 38, no. 1, pp. 86–104, 2018.

[24] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli, "The uva/padova type 1 diabetes simulator: new features," *Journal of diabetes science and technology*, vol. 8, no. 1, pp. 26–34, 2014.