RL-based Automated Glucose Control for People with Type 1 Diabetes

Hayley Hawkins (535005132), Jessica Rumsey (436000653)

Abstract

Automated insulin delivery remains a central topic in the management of Type 1 Diabetes. Manual systems introduce the possibility of human error, and implementing an automated system has become much more popular for insulin-dosing. In this work, we explore reinforcement learning for fully automated closed-loop glucose regulation that does not depend on meal announcement. We build on the well-known RL4T1D framework developed by Hettiarachchi et al. (2024) to simulate glucose-insulin dynamics using the FDA-approved virtual Simglucose environment. Our approach models blood glucose control as a Partially Observable Markov Decision Process, and our RL agent learns policies by using Proximal Policy Optimization (PPO). However, due to environmental setup issues our results were less-than-ideal, revealing unstable learning, elevated risk, and early episode termination.

1 Introduction

Although automated insulin delivery has evolved for Type 1 Diabetes care, most commercial systems rely on manual meal announcements and patient approximate insulin needs. However, manual input is not only a cognitive burden for patients with Type 1 Diabetes, but introduces human error and limits the effectiveness of glucose control. This paper investigates a reinforcement learning approach to closed-loop insulin dosing that removes the need for meal announcements and aims to achieve fully automated glucose regulation. Our project is inspired by the "RL for Automatic Treatment in Type 1 Diabetes" (RL4T1D) codebase that trains RL agents to estimate insulin doses and maintain healthy glucose levels for people with this disease (Hettiarachchi et al., 2024).

The objective of our project is to implement the environment simulation, agent architecture, and training loops provided in the aforementioned codebase. We work to properly evaluate the success of the Proximal Policy Optimization (PPO) reinforcement learning algorithm in the closed-loop blood glucose control problem.

2 Background

Artificial pancreatic systems automate insulin delivery and stabilize blood glucose levels to mimic the role of a pancreas for people with Type 1 Diabetes. An artificial pancreas has three main components: (1) continuous glucose monitoring (CGM), (2) an insulin pump to deliver insulin, and (3) a control algorithm. As described by Bothe et al. (2014), because the success of an artificial pancreas is dependent on the needs of an individual, the control algorithm must be an adaptive algorithm.

Reinforcement learning can be applied to decisions that rely on an observed state so can be adapted to an artificial pancreas system where the agent must continuously observe glucose concentration and determine the amount of insulin to provide and the time of dosage. Work towards a fully closed-loop model, where insulin dosages are exclusively determined by parameters measured in a person's body, and the necessity of an adaptive algorithm motivates our use of reinforcement learning in this optimization problem.

The RL4T1D codebase adapted for our project is an in silico analysis of the closed loop glucose control problem towards an autonomous artificial pancreas system and relies on Simglucose, an open-source Python simulation environment for Type 1 Diabetes. Simglucose provides a "population" of virtual patients for research experiments thus removing patient risk associated with experiments of closed-loop insulin delivery. It supports reinforcement-learning style interaction: at each time step, the environment returns an observation, allows an action, computes the next state and rewards.

The simulator provides continuous glucose monitor (CGM)-style readings and records recent insulin actions and meal events, allowing agents to construct state representations from past glucose and insulin histories. At each time step, the RL agent issues a continuous or discretized insulin dose, which the simulator applies to update glucose levels according to physiological dynamics. Typical simulations span one or multiple days with meals and disturbances, offering a validated and flexible environment for developing and evaluating automated glucose-control algorithms before human trials.

Proximal Policy Optimization (PPO) formulated by Schulman et al. (2017) is an algorithm to rival the data efficiency and performance of Trust Region Policy Optimization (TRPO) while only using first-order optimization. Our project implements PPO for the blood glucose control optimization problem related to Type 1 Diabetes treatment.

TRPO proposed by Schulman et al. (2017) combats instability in traditional policy gradient algorithms by constraining how much a policy changes per update, ensuring each update is meaningful (improves performance) and safe (does not destabilize what is already working). A local approximation of the objective function constrained by an upper bound on the KL divergence between the current and old policy achieves this. Adapting TRPO for first-order optimization, the PPO algorithm follows:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\Big[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\Big]$$

The rt() term denotes the importance sampling ratio, and the second term clips this importance sampling ratio to discourage drastic changes to the policy. To reduce, with the goal to entirely eliminate, cognitive burden on people with Type 1 Diabetes, Hettiarachchi et. al (2022) suggested PPO as a control algorithm for the ongoing problem of optimizing blood glucose levels optimization problem. They use PPO to train a RL agent and experiments are conducted in silico on the 10 subject, adult cohort in the Simglucose simulator. Conducting 1,000 trials for each subject, the mean total time spent between 70 - 180 mg/dL of glucose, called Time in Range (TIR), was 65%, a result comparable to other proposed methods to this control problem. We attempt to replicate these results.

3 Related Works

Understanding how other researchers have approached glucose control and medical decision-making provides valuable context for our own work. A recent study by Emerson et al. (2023) explores offline reinforcement learning for blood glucose regulation, emphasizing safety and stability in insulin dosing decisions. Unlike our project, which trains agents online through simulated interaction in Simglucose, their work focuses on leveraging historical patient data to learn policies without additional exploration risk. This complementary approach highlights the growing emphasis on balancing learning performance with clinical safety in RL-based diabetes management.

Another study by Fang et al. (2023) introduces Offline Inverse Constrained Reinforcement Learning (ICRL) as a framework for safe-critical decision making in healthcare. Their approach extends traditional RL by explicitly incorporating safety constraints and inverse learning objectives, ensuring that learned policies adhere to clinical safety limits derived from expert behavior. While our project focuses on direct policy optimization through the PPO algorithm in a simulated environment, Fang et al.'s method demonstrates how constraint-aware learning can reduce the risk of unsafe exploration.

Only recently has the PPO algorithm for Type 1 Diabetes evolved to Dual PPO implemented by Marchetti et al. (2025). Dual PPO employs two PPO agents each with a unique constraint on the maximum rate of insulin they can deliver. These unique constraints ensure only one agent is active

at any given time and transition between the two agents depends on readings taken from the CGM. By a predetermined patient-specific transition threshold, the first agent operates above the transition threshold to handle hyperglycemia while the second agent operates in the target blood glucose range between a set safety threshold and the transition threshold. Insulin delivery automatically halts when CGM readings are below the safety threshold to target hypoglycemic cases. Dual PPO significantly improves TIR compared to results from the implementation of single PPO. We examine this paper as evidence of future directions for our project and the evolving nature of solving the blood glucose levels optimization problem.

4 Methods

Problem Definition

We implement Hettiarchchi's et al. (2022) PPO algorithm for closed loop glucose control. The environment is modeled by a Partially Observable Markov Decision Process outlined by the 6-tuple (S*, O, S, A, P, R) (true states, observation function, noisy states, action, probability and reward function). The state space is defined by the stochastic observation function, O, which maps the true states of the Simglucose simulator to the glucose sensor observations, administered insulin, and meal announcements at time step t and their historical measurements. Actions are defined on a continuous action space and are the insulin dose administered at each time step t.

The reward function formalized by Hettiarchchi et al. (2022) utilizes the blood glucose Risk Index (RI) to quantify the risk associated with low and high blood glucose levels. RI is the sum of the Low Blood Glucose Risk Index (LGBI) and the High Blood Glucose Index (HGBI) which are measures of hypoglycemia and hyperglycemia. Glucose values below 39 mg/dL are highly penalized (-15,000) because they are outside the glucose sensor's detectable range while all other glucose readings are rewarded proportional to themselves and the negative RI. Any episodes with simulated glucose values outside the glucose sensor's full detectable range from 39 - 600 mg/dL are terminated.

Success is observed by measuring Time in Range (TIR), the total time a patient spends between 70 - 180 mg/dL of glucose, and minimizing RI which is equivalent to minimizing LGBI and HGBI. Suitable hypo- and hyperglycemic risk profiles are $2.5 < \text{LGBI} < 5$ and $10 < \text{HGBI} < 15$ (Hettiarchchi 2022).

Implementation Details

We worked to properly execute the PPO algorithm for one patient and one day. The Guardian RT glucose sensor and the Insulet pump from Simglucose are used to simulate glucose dynamics, and they do so at 5-minute intervals. Therefore, 288 timesteps is one full simulated day. We adjusted hyperparameters such as max_epi_length, the maximum number of insulin actions per episode, and total_interactions, the total number of insulin actions taken every 5 minutes, to accommodate our desired execution. Notably, total_interactions / max_epi_length is approximately equal to the number of days training is performed. Since the environment is stochastic, each day, that is each episode, the meal times, sizes, and probability of occurring vary as well as the patient's starting glucose state. However, because we trained on only one patient, the PPO agent was limited to one metabolic profile and encountered the same insulin sensitivity, carb ratio, and endogenous glucose production each day, to name a few.

We focused our efforts on adolescent 0 (patient_id = 0). Maintaining the agent's default configuration given by Hettiarchchi et al. but adjusting max_epi_length to 288 timesteps and patient_id, we ran our first trial of the PPO algorithm. We compared the product of this minimally adjusted experiment to the baseline results for the PPO algorithm provided in the RL4T1D GitHub. Major discrepancies between our results and the provided baseline motivated our adjustments of hyperparameters.

The PPO algorithm has a Long Short-Term Memory (LTSM) architecture which typically means sequences of experiences generated by the agent are flattened for loss. If the code flattens inconsistently, value network outputs and target returns can have mismatched shapes, leading to amplified value loss. Our training experienced value losses greater than 3000, so we added reassurances in the code to ensure the value network and target return would have identical shapes.

For quicker results, we decreased total interactions from the default value to 100,000. Though this decreases learning opportunities for the agent, we suspected this value was still suitable for our

simple one patient, one day experiment. Encountering abnormalities with the agent's insulin administration, we increased the entropy coefficient to 0.05 to encourage the agent to try different insulin dosages earlier. We also decreased both value and policy network learning rates to accommodate our limited data for a one patient, one day experiment and avoid overfitting or a degenerate policy. We did set the value network (critic) slightly higher (2e-4) than the policy network (actor) learning rate (1e-4)to encourage the critic to learn fast and the actor to learn cautiously. Additionally, we did not use parallel workers.

5 Experiments

We attempted to conduct three experiments on a single virtual patient using the Simglucose environment and RL4T1D codebase: (1) the clinical treatment algorithm (Basal-Bolus) as a baseline, (2) the PPO agent with default hyperparameters, and (3) a modified PPO agent with adjusted hyperparameters. Each experiment reports Time in Range (normo), time spent in hypoglycemia (hypo), time spent in hyperglycemia (hyper), and the LGBI and HGBI used to compute the overall RI. Higher RI indicates worse glucose control, whereas higher normoglycemic percentage indicates better outcome.

In the summary metrics table produced at the end of each experiment, normo, hypo, and hyper represent the percentage of timesteps the patient's blood glucose levels fall within each range. The ranges are 70 - 180 mg/dL, < 70 mg/ dL, and > 180 mg/dL. However, the percentage is not consistently calculated using 288 total timesteps. The percentages are based on the number of timesteps each episode experiences, whether the episode terminates early or not. Therefore, the percentages across episodes are not comparable. An important note about our experiment is that every episode would terminate immediately once the agent received a reward of -15, indicating a glucose drop of below 39 mg/dL.

Clinical Baseline

Clinical Baseline (Basal-Bolus) Results Table

Statistic	Normo (%)	Hypo (%)	Hyper (%)	LGBI	HGBI	RI	Fail (%)
Mean	100.00	0.00	0.00	0.223	2.480	2.704	0.00
Min	100.00	0.00	0.00	0.149	2.029	2.401	0.00
Max	100.00	0.00	0.00	0.372	2.706	2.856	0.00

The clinical treatment algorithm produced near-ideal glucose outcomes for this patient, with approximately 100% time in range, 0% failures, and a low overall RI ≈ 2.7. Both LGBI and HGBI remained low, suggesting low hypoglycemic and hyperglycemic exposure. These values provided a useful lower bound for which RL agents can be compared and confirm that the clinical baseline exhibits safe and stable glucose regulation under controlled conditions. Compared to PPO, the clinical controller is substantially safer but less adaptable, highlighting the tradeoff between safety and optimal glycemic performance.

PPO 1

PPO 1: 1 Patient 1 Day Summary Statistics Table (800,000 Interactions)

Statistic	Normo (%)	Hypo (%)	Hyper (%)	S-Hypo	S-Hyper	LGBI	HGBI	RI	Reward	Fail (%)
Mean	73.44	9.53	0.00	17.03	0.00	14.11	0.64	14.75	−1.51	100.00
Std	7.18	4.59	0.00	5.30	0.00	2.45	0.40	2.60	3.57	100.00
Min	50.00	0.00	0.00	5.88	0.00	6.33	0.00	6.49	−26.19	100.00
Max	88.24	34.62	0.00	41.18	0.00	24.26	2.20	25.11	4.24	100.00

The default PPO configuration with 800,000 interactions maintains glucose in the target range roughly 73% of the time but shows a considerable amount of hypoglycemia (mean 9.53%) which exceeds clinically acceptable limits. The wide variability in severe hypoglycemia (5.88 to 41.18) suggests an unstable policy that sometimes overdoses insulin. These results indicate that the default PPO configuration, at least within our environment set up, is not sufficient for closed-loop insulin delivery.

PPO 2

PPO 2: 1 Patient 1 Day Summary Statistics Table (100,000 Interactions)

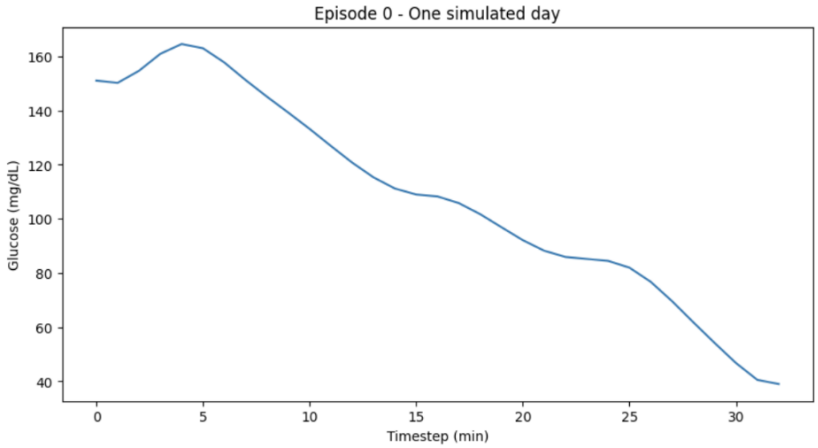
Statistic	Normo (%)	Hypo (%)	Hyper (%)	S-Hypo	S-Hyper	LGBI	HGBI	RI	Reward	Fail (%)
Mean	81.54	5.75	0.00	12.71	0.00	10.85	1.16	12.01	6.42	100.00
Std	8.43	1.70	0.00	7.47	0.00	3.61	0.18	3.73	3.86	100.00
Min	70.15	2.94	0.00	5.88	0.00	6.77	0.85	7.88	1.89	100.00
Max	91.18	7.50	0.00	23.88	0.00	13.75	1.28	15.02	12.18	100.00

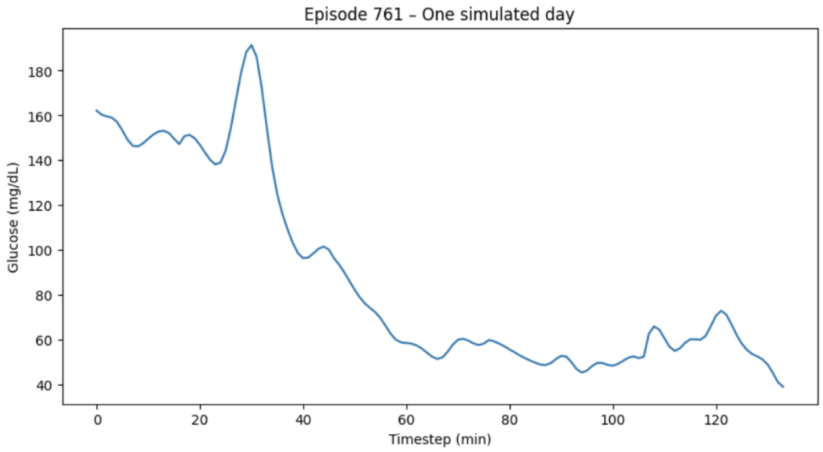
After adjusting total interactions, entropy, and learning rates, our second PPO experiment demonstrates higher time-in-range (mean 81.54%) and reduced hypoglycemia compared with the first experiment, indicating modest improvements. However, hypoglycemia still remains high and LGBI values are higher than what would be expected for a safe clinical system. Although the mean reward becomes more positive, the RI is still higher than the clinical baseline. Looking at both tables indicates that tuning hyperparameters can improve performance, but learning remains unstable.

6 Discussion

Our experimental results emphasize the need for further examination and trials of our implementation of the PPO algorithm from the RL4T1D codebase. With repeated failures across experiments despite numerous hyperparameter adjustments and debugging steps, we hypothesize the environment set up, rather than the PPO agent, is the cause of our faulty training.

Simglucose is supposed to simulate meals according to probabilities determined in the environment configuration. For our experiments, there was a 95% chance a patient ate breakfast, lunch, and dinner. However, our episodes rarely included meals. The readings from the continuous glucose monitor illustrated that without meals, blood glucose levels rapidly decreased, and all of our episodes terminated early because the patient's blood glucose fell below 39 mg/dL before the full 24-hour duration.





We plot glucose (mg/dL) vs. timestep (min) for episode 0 and episode 761 to highlight the difference between episodes that did and did not receive a meal. The patient did not eat a meal in episode 0, and thus their blood glucose levels continuously decreased, and the episode terminated before three hours. The graph of episode 761 shows the effect of a single meal on glucose levels. The corresponding carbohydrate intake temporarily elevates the patient's blood glucose before it rapidly declines. This was one of our longest episodes which terminated at 11 hours.

Incorrect meal administration also affected the LGBI and HGBI values. In particular, LGBI was extremely high, indicating a greater risk of frequent or extreme instances of low blood sugar. Skipping meals often induces hypoglycemia which fits with the context of our problem. Instances of hyperglycemia did occur but they were sparse and thus represented by 0 in the summary metrics table. The LGBI and HGBI values presented by the clinical baseline further affirm our numbers are outside of their appropriate risk profiles.

Due to frequent early episode terminations, we had to carefully interpret the TIR percentage in the summary metrics table. While an average TIR of $\tilde{8}0\%$ appears to exceed the baseline results for PPO provided in the original RL4T1D GitHub, we learned to interpret this as 80% of the timesteps the episode actually experienced. Therefore, for our episodes that terminated before hour 3, of approximately 30 timesteps, 24 of them were in the normo range. If blood glucose levels are never interrupted by meals, they continuously decrease. So, this interpretation illustrates how rapidly blood glucose levels drop as it only took 6 steps outside of the normo range for the patient's glucose levels to fall below the sensor's threshold.

Most obviously, our PPO agent never administered insulin regardless of encouraging early exploration and encountering hypo- and hyperglycemic states. Despite adjusting hyperparameters, we were never able to resolve this issue. Given the incorrectly simulated meals, incorrect LGBI and HGBI values, early episode termination, and no insulin administration, we strongly believe issues with the environment set up inhibited proper training of the PPO agent. Most likely, communication between Simglucose and our agent was barred so neither received proper information to accommodate the other.

7 Conclusion

This project investigated and attempted to replicate fully automated, closed-loop insulin control using PPO within the RL4T1D framework. Although we tried to follow the original design as closely as possible and included hyperparameter adjustments, our experimental results showed consistently elevated hypoglycemic risk, unstable learning, and early episode termination. After looking through our results and plotting multiple episodes for visual aid, we believe the issue likely stems from our environment configuration. Specifically, incorrect meal administration, misreported glucose events, and premature terminations. These less-than-ideal findings highlight that reliable RL for automated insulin delivery depends critically on accurate simulation-agent-environment communication. If

this project were to be redone, we would include a heavier focus on validating the Simglucose setup before trying to troubleshoot through hyperparameter adjustments.

8 References

Bothe, M. K., Dickens, L., Reichel, K., Tellmann, A., Ellger, B., Westphal, M., & Faisal, A. A. (2013). The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas. Expert review of medical devices, 10(5), 661-673.

Emerson, H., Guy, M., & McConville, R. (2023). Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes. Journal of Biomedical Informatics, 142, 104376.

Fang, N., Liu, G., & Gong, W. (2025). Offline inverse constrained reinforcement learning for safe-critical decision making in healthcare. IEEE Transactions on Artificial Intelligence.

Hettiarachchi, C., Malagutti, N., Nolan, C., Daskalaki, E., & Suominen, H. (2022, July). A reinforcement learning based system for blood glucose control without carbohydrate estimation in type 1 diabetes: In silico validation. In 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 950-956). IEEE.

Hettiarachchi, C., Malagutti, N., Nolan, C. J., Suominen, H., & Daskalaki, E. (2024). G2P2C—A modular reinforcement learning algorithm for glucose control by glucose prediction and planning in Type 1 Diabetes. Biomedical Signal Processing and Control, 90, 105839.

Man, C. D., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., & Cobelli, C. (2014). The UVA/PADOVA type 1 diabetes simulator: new features. Journal of diabetes science and technology, 8(1), 26-34.

Marchetti, A., Sasso, D., D'Antoni, F., Morandin, F., Parton, M., Matarrese, M. A. G., & Merone, M. (2025). Deep reinforcement learning for Type 1 Diabetes: Dual PPO controller for personalized insulin management. Computers in Biology and Medicine, 191, 110147.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In International conference on machine learning (pp. 1889-1897). PMLR.

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

Viroonluecha, P., Egea-Lopez, E., & Santa, J. (2022). Evaluation of blood glucose level control in type 1 diabetic patients using deep reinforcement learning. Plos one, 17(9), e0274608.

Code References:

- Google Colab for glucose graphs
- Github Codebase for RL training