# Reinforcement Learning for Blood Glucose Control: Challenges and Opportunities

**Ian Fox** [1] **Jenna Wiens** [1]

## Abstract

Individuals with type 1 diabetes (T1D) lack the ability to produce the insulin their bodies need. As a result, they must continually make decisions about how much insulin to self-administer in order to adequately control their blood glucose levels. Longitudinal data streams captured from wearables, like continuous glucose monitors, can help these individuals manage their health, but currently the majority of the decision burden remains on the user. To relieve this burden, researchers are working on closed-loop solutions that combine a continuous glucose monitor and an insulin pump with a control algorithm in an 'artificial pancreas.' Such systems aim to estimate and deliver the appropriate amount of insulin. Here, we investigate the utility of reinforcement learning (RL) techniques for automated blood glucose control. Through a series of experiments, we compare the performance of different deep RL approaches to non-RL approaches. We find that the RL approaches are competitive with the baselines (achieving an average risk across three patients of 8.56 *vs.* the baseline 8.48) and are better able to handle latent behavioral patterns (improving risk in one patient to 9.26 *vs.* the baseline 11.80). These preliminary results suggest that RL could be useful for improving blood glucose control algorithms.

## 1. Introduction

Type 1 diabetes (T1D) is a chronic disease affecting 20-40 million people worldwide (You & Henneberg, 2016), and its rate of occurrence is increasing (Tuomilehto, 2013). People with T1D cannot produce insulin, a hormone that signals cells to uptake glucose in the bloodstream. Without insulin, the body must metabolize energy in other ways that, when relied on repeatedly, can lead to life-threatening conditions (Kerl, 2001). Moreover, the accumulation of glucose in the blood stream can lead to hyperglycemia, and this in turn can lead to nerve, eye, heart, and kidney damage (DeFronzo et al., 2015). Tight blood glucose control with insulin injections can help (Control et al., 1995), but intensive control can increase risk of hypoglycemia (i.e., low blood sugar), which in turn can lead to increased risk of heart disease, seizures, and sudden death.

To safely and appropriately control blood glucose levels, individuals with T1D must continually make decisions about how much insulin to self-administer. This requires careful measurement of glucose levels and carbohydrate intake, resulting in at least 15-17 data points a day. If the individual uses a continuous glucose monitor (CGM), this can increase to over 300 data points, or a blood glucose reading every 5 minutes (Coffen & Dahlquist, 2009).

Combined with an insulin pump, a wearable device that automates the delivery of insulin, CGMs present an opportunity for closed-loop control. Such a system, known as an 'artificial pancreas' (AP), automatically anticipates the amount of required insulin and delivers the appropriate dose. This would be life-changing for many individuals, since it would relieve the decision burden placed on those with T1D. For many years, researchers have worked towards the creation of an AP for blood glucose control (Kadish, 1964; Bequette, 2005; Bothe et al., 2013). Though the technology behind CGMs and insulin pumps has advanced, there remains significant room for improvement when it comes to the control algorithms (Bothe et al., 2013; Pinsker et al., 2016). Current approaches cannot leverage latent behavioral patterns, nor can they easily incorporate additional useful signals (*e.g.*, physical activity).

In this work, we investigate the utility of a reinforcement learning (RL) based approach for blood glucose control (Bothe et al., 2013). RL is particularly well-suited for this task because it: i) can readily incorporate additional data streams (as part of the state representation), ii) makes minimal assumptions about the structure of the underlying process, allowing the same system to adapt to different individuals or to changes in individuals over time, and iii) can learn

[1]Computer Science Engineering, University of Michigan, Michigan, USA. Correspondence to: Ian Fox <ifox@umich.edu>.

*Reinforcement Learning for Real Life (RL4RealLife) Workshop in the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, USA, 2019. 

to leverage patterns such as regular meal times. Finally, it can take advantage of existing FDA-approved simulators for model training. Given that RL is a viable adaptive control method across a variety of tasks (Silver et al., 2018; Rajeswaran et al., 2018), we hypothesize that it can be used to learn high-performance blood glucose control algorithms.

To test this hypothesis, we evaluate model-free RL algorithms for blood glucose control. In our experiments, we leverage an FDA-approved simulator for the glucoregulatory system that simulates 30 different patients (10 children, 10 adolescents, and 10 adults). We present empirical results from three different RL-based approaches. In addition, we compare to several non-RL baselines including a 'basal-bolus' (BB) controller and a proportional-integral-derivative (PID) control algorithm. Preliminary results suggest that at least in some settings, RL can lead to better policies compared to non-RL baselines (average risk 9.26 *vs.* 11.80). Going forward, in the context of diabetes, RL could be used to more effectively adapt to latent behavioral patterns.

## 2. Background and Related Works

In recent years, RL (and in particular deep RL) has had a number of successes (e.g., Alpha Go (Silver et al., 2018) and Atari (Mnih et al.)). Within an RL framework, one seeks a mapping from some set of observations describing the current state (e.g., current blood glucose and historical insulin levels) to an action (e.g., bolus of insulin) that maximizes some notion of reward (e.g., precise glucose control). Within healthcare and medicine, researchers have started to explore the RL framework as a solution for matching patients to treatment, since it naturally reframes the problem from a diagnosis-based problem to an action-based problem (Komorowski et al., 2018).

Despite its success in other domains, RL has yet to be fully explored as a solution for a closed-loop AP system (Bothe et al., 2013). RL is a promising approach to this task, as it is well-suited to learning complex behavior that readily adapts to changing domains (Clavera et al., 2018), but can be limited by the amount of data required to learn effective policies. However, unlike many other disease settings, there exist credible simulators for the glucoregulatory system (Visentin et al., 2014). These simulators have been used before for learning problems. Specifically, researchers have investigated the use of off-policy evaluation for discovering good open-loop control parameters in diabetes simulations (Thomas & Brunskill, 2017). But, we are unaware of work using these simulators to learn blood glucose control policies with deep RL.

### 2.1. Current AP algorithms

There have been three main branches of techniques used to create APs: PID control (Steil, 2013), model predictive control (MPC) (Bequette), and fuzzy logic (FL) (Atlas et al., 2010). PID controllers are by far the most common (Steil, 2013), and as a result we focus on them here. The simplicity of PID controllers make them easy to use, and in practice they achieve strong results. For example, the Medtronic Hybrid Closed-Loop system, one of the few commercially available, is built on a PID controller (Garg et al., 2017; Ruiz et al., 2012). The main weakness of a PID controller, in the setting of blood glucose control, is their reactivity. As they only respond to current glucose values (including a derivative), they cannot respond fast enough to meals to satisfactorily control postprandial excursions without meal announcements (Garg et al., 2017), and without additional safety modifications can overcorrect for these spikes, triggering postprandial hypoglycemia (Ruiz et al., 2012). In contrast, we hypothesize that an RL approach will be able to leverage patterns associated with meal times, resulting in better policies that can anticipate meals.

### 2.2. Glucose Models and Simulation

Models of the blood glucose system have long been seen as an important component for the development and testing of an AP (Cobelli et al., 1982). Models can be used as simulation environments for testing the efficacy of control systems (Kovatchev et al., 2009), as controllers for administering insulin (Bequette), or as tools to gain a deeper understanding of the physiological processes at work (Bergman, 1989). Current models are built using a combination of rigorous experimentation and expert knowledge of the underlying physiological phenomena. Typical models are built on an underlying multi-compartment model, with various sources and sinks corresponding to physiological phenomena, involving often dozens of patient-specific parameters. One such simulator, the one we use in our experiments, is the UVA/Padova model (Kovatchev et al., 2009). We explain this simulator in greater detail in **Section 3.1**.

## 3. Methods

In this work, we examine the use of RL and non-RL control algorithms for blood glucose control. We begin by formalizing the problem. We then present two baselines: an analogue to human-control in the form of a basal-bolus controller and a PID controller. Finally, we describe three deep Q-network (DQN) implementations that vary in terms of architecture and state representation.

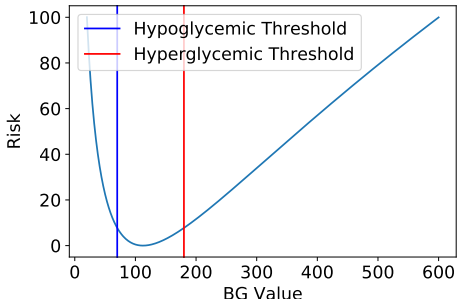


*Figure 1.* The risk function proposed in (Clarke & Kovatchev). This shows the mapping between blood glucose values (in mg/dL, x-axis) and Risk values (y-axis). The hypo- and hyperglycemic thresholds are shown in vertical lines, they correspond to a risk value of 7.75.

### 3.1. Problem Setup

We frame the problem of blood glucose control as a Markov decision process (MDP) consisting of the 4-tuple $(S, A, P, R)$. Our precise formulation of this problem varies depending on the method and setting. Here, we describe the standard formulation, and explain further differences as they arise. States $s_t \in S$ consist of the previous 24 hours of blood glucose and insulin data at the resolution of 5-minute intervals: $s_t = [\mathbf{b}^t, \mathbf{i}^t]$ where:

$$\mathbf{b}^t = [b_{t-287}, b_{t-286}, \dots b_t], \mathbf{i}^t = [i_{t-287}, i_{t-286}, \dots i_t]$$

and $b_t \in \mathcal{N}_{40:400}$, $i_t \in \mathcal{R}_+$, $t \in \mathcal{N}_{0:288}$ and represents a time index for a day at 5-minute resolution.

Actions $a_t \in A$ are real positive numbers, denoting the size of the insulin bolus in medication units. The transition function $P$ consists of two elements: i) $G : (a_t, c_t) \rightarrow (b_{t+1}, i_{t+1})$, where $c_t \in \mathcal{R}_+$ is the amount of carbohydrates input at time $t$, $G$ is a model of the glucoregulatory system, its behavior is defined in accordance with the UVA/Padova simulator (Kovatchev et al., 2009), ii) $M : t \rightarrow c_t$ is the meal schedule, and is defined according to **Algorithm 1**. Note the specific numbers are largely derived from the implementation of (Xie, 2018). The reward function $R$ is defined according to $risk(b_{t+1}) - risk(b_t)$ where $risk$ is the asymmetric blood glucose risk function defined as:

$$risk(b) = 10 * (1.509 * \log(b)^{1.084} - 5.381)$$

shown in **Figure 1** (Clarke & Kovatchev). Note that our implementation of the UVA/Padova simulator builds off the open-source implementation of (Xie, 2018).

Briefly, the simulator models the glucoregulatory system as a nonlinear multi-compartment system, where glucose is generated through the liver and absorbed through the gut and is controlled by externally administered insulin. A more detailed explanation can be found in (Kovatchev et al., 2009). The version of the UVA/Padova simulator we use comes with 30 virtual patients, each of which consists of several dozen parameters fully specifying the glucoregulatory system. The patients are divided into three classes: children, adolescents, and adults, each category with 10 patients.

**Algorithm 1** Generate Meal Schedule

**Input:** body weight $w$, number of days $n$
$MealOcc = [0.95, 0.3, 0.95, 0.3, 0.95, 0.3]$
$TimeLowerBounds = [5, 9, 10, 14, 16, 20] * 12$
$TimeUpperBounds = [9, 10, 14, 16, 20, 23] * 12$
$TimeMean = [7, 9.5, 12, 15, 18, 21.5] * 12$
$TimeStd = [1, .5, 1, .5, 1, .5] * 12$
$AmountMean = [0.7, 0.15, 1.1, 0.15, 1.25, 0.15] * w$
$AmountStd = AmountMean * 0.15$
$Days = []$
**for** $i \in [1, \dots, n]$ **do**
  $M = [0]_{j=1}^{288}$
  **for** $j \in [1, \dots, 6]$ **do**
    $m \sim Binomial(MealOcc[j])$
    $lb = TimeLowerBounds[j]$
    $ub = TimeUpperBounds[j]$
    $\mu_t = TimeMean[j]$
    $\sigma_t = TimeStd[j]$
    $\mu_a = AmountMean[j]$
    $\sigma_a = AmountStd[j]$
    **if** $m$ **then**
      $t \sim Round(TruncNormal(\mu_t, \sigma_t, lb, ub))$
      $c \sim Round(max(0, Normal(\mu_a, \sigma_a)))$
      $M[t] = c$
    **end if**
  **end for**
  $Days.append(M)$
**end for**

### 3.2. Basal-Bolus Baseline

This baseline is designed to mimic human control and is typical of how an individual with T1D currently controls their blood glucose. In this setting, we modify the standard state representation to include a carbohydrate signal and a cooldown signal (explained below), and to remove all non-current measurements $s_t = [b_t, i_t, c_t, cooldown]$. Note that this inclusion means this is not a 'closed-loop' control scheme, as the burden to provide information on carbohydrates falls on the individual. Each virtual patient in the simulator comes with the parameters necessary to calculate optimal basal insulin rates $bas$, a correction factor $CF$, and carbohydrate ratio $CR$. These three parameters, together with a glucose target $b_g$ define a policy

$\pi(s_t) = bas + (c_t > 0) * (\frac{c_t}{CR} + cooldown * \frac{b_t - b_g}{CF})$ where $cooldown$ is 1 if there have been no meals in the past three hours, otherwise it is 0. This ensures that each meal is only corrected for once, otherwise meals close in time could lead to over-correction and hypoglycemia.

### 3.3. PID Baseline

Variants of PID controllers are already used in commercial AP applications (Garg et al., 2017). A PID controller operates by setting the control variable, here $a_t$, to the weighted combination of three terms $a_t = k_P P(b) + k_I I(b) + k_D D(b)$ such that the process variable $b_t$ (where $t$ is again the time index) remains close to a specified setpoint $b_g$. The terms are calculated as follows: i) the proportional term $P(b_t) = \max(0, b_t - b_g)$ increases the control variable proportionally to the distance from the setpoint, ii) the integral term $I(b_t) = \sum_{j=0}^{t}(b_j - b_g)$ acts to correct long-term deviations from the setpoint, and iii) the derivative term $D(b_t) = |b_t - b_{t-1}|$ acts to control a basic estimate of the future, here approximated by the rate of change. The set point and the weights (also called gains) $k_P, k_D, k_I$ are hyperparameters. In our experiments, we set these hyperparameters using multiple iterations of grid-search (using training seeds for the environment) with exponential refinement between iterations.

### 3.4. Q Learning

#### 3.4.1. General Formulation

Within a Q-learning framework (Watkins & Dayan, 1992), the state-action value function $Q(s, a)$ is learned through temporal-difference updates: $Q(s_t, a_t) = R(s_t, a_t) + (1 - \gamma) \max_{a^* \in A} Q(s_{t+1}, a^*)$ where $R$ is the reward function described above. From this Q-function, one can extract the optimal policy as $\pi^*(s_t) = \text{argmax}_{a \in A} Q(s_t, a)$. Note that this formulation requires discrete action bins. Here, we used an action bin formulation based on the per-patient basal rate used for the basal-bolus controller, $bas$. We discritized the action space into three bins: $\{0, bas, 5 * bas\}$. This could disadvantage the Q-learning based approaches relative to the baselines, which use a continuous action space. We explored the use of policy-gradient methods which allow for continuous control, but failed to achieve comparable performance.

### 3.5. Oracle Q Learning

A deep RL approach to learning AP algorithms necessitates two functioning parts: i) the representation learned by the network must contain sufficient information to control the system, and ii) an appropriate control algorithm must be learned from interaction. As we are working with a simulator, we first explore the difficulty of task (ii) in isolation, by replacing the state $s_t$ with the ground-truth state of the simulator at time $t$, which is a 13-dimensional vector with real valued elements representing glucose, carbohydrate, and insulin values in different compartments of the body. Though this representation is not available for real applications, it approximates an upper limit of performance with our current learning framework. The Q network is a fully-connected network with two hidden layers, each with 256 units.

### 3.6. Deep Q Learning

In our full Q-learning approach, we use our original definition of state, which includes the past 24 hours of CGM and insulin data (note: no carbohydrate information). This setup is both plausible for real-world applications, and allows for fully closed-loop operation. We investigate using a 1d-CNN and GRU for our deep Q network, as these types of architectures have successfully been applied to blood glucose data in the past (Fox et al.; Zhu et al., 2018).

### 3.7. Experimental Setup & Evaluation

To measure the utility of deep RL for the task of blood glucose control, we learned policies using the approaches described above, and tested these policies on simulated data with different random seeds across several different individuals.

We trained our models for 800 epochs (batch size 128) with an experience replay buffer of size 30k and a discount factor of 0.99. We trained our RL models using $\epsilon$-greedy exploration with $\epsilon = 0.05$. We optimized the Huber loss of our temporal difference predictions using Adam with a learning rate of $10^{-3}$ for the CNN and $10^{-5}$ for the GRU (chosen using performance on training seeds). Our networks were initialized using PyTorch defaults.

Our network architectures were as follows:

**Oracle-Q Network:** A 2 layer fully connected network with 256 units, batch norm, and ReLU nonlinearities.

**CNN-Q:** A 1-d CNN with two blocks, each consisting of 2 iterations of alternating width-3 32-channel convolutional layers and batch norm/ReLU, followed by width-2 max pooling. This is followed by a fully connected layer with 512 units with batch norm, ReLU, and dropout (with $p = 0.2$), and finally a fully connected output layer.

**GRU-Q:** Our GRU has 2 recurrent layers of size 128 followed by a fully connected output layer.

We evaluated policies on 10 continuous simulation days using average risk (see **Figure 1**). Due to computational costs, we were unable to learn CNN-Q and GRU-Q networks for all patients, thus in those experiments, we focused on a subset of patients (one child, one adult, and one adolescent).

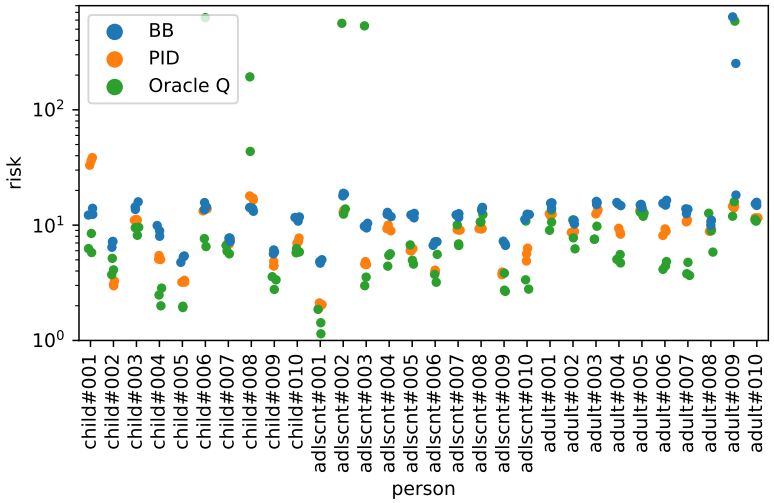


*Figure 2.* The average risk over 10 days from different methods on different simulated patients. Each point corresponds to a different random seed, that controls initialization, the meal schedule, and randomness in training. On average, the Oracle Q method does the best, though it occasionally fails catastrophically by providing far too much insulin.

## 4. Results

We investigate the performance of different policies in several stages. In the first stage, we compare the performance of the 'basal-bolus' controller, the PID, and the oracle DQN across all thirty patients. In the second stage, we compare to the CNN-Q and GRU-Q on a subset of patients. Finally, we show how the uncertainty of the meal schedule can affect the performance of controllers in different ways.

### 4.1. Baseline Models vs. Oracle DQN

Results comparing the basal-bolus and PID baselines to the oracle Q network are given in **Figure 2**. Each point represents a different policy, resulting from a different initialization. Despite the variation across initializations, a clear tread emerges: closed-loop control algorithms that can deliver frequent small doses of insulin can significantly outperform a 'basal-bolus' controller (oracle Q outperforms in 82/90 runs). This suggests that, in addition to relieving decision burden, AP systems could lead to overall better blood glucose control. In addition, the policy learned using the oracle Q outperformed the simple PID controller, reducing risk in 68/90 runs across the thirty patients. Recall, that the Oracle Q network has access to the ground truth state, but does not have access to the future (i.e., does not know when a meal is coming until it sees it), so we do not expect perfect control. The average risk for most individuals is above the risk threshold for hyper/hypoglycemia of 7.75. This is far from the optimal level of control. However, it is not the case that all time is spent hypo/hyperglycemic. Across patients, approximately 60-80% of time is spent euglycemic. If insulin is not given well in advance of meals, glucose can increase significantly for a brief period of time, leading to elevated average/mean risk. This skews the distribution of risk towards hyperglycemia and therefore increased risk.

### 4.2. Baseline Models vs. DQN

To explore the ability of the DQN to learn the necessary representations, we next compare the two non-oracle architectures CNN-Q and GRU-Q to the oracle DQN in addition to the other baselines. Again, we limit our evaluation to three simulated patients 'Adolescent 1', 'Adult 1', and 'Child 3.' We selected Child 3 over Child 1, as a previous iteration of the PID controller was unable to achieve stable performance on Child 1 (this has since been corrected).

We present our results in terms of average risk across 10 heldout days of simulation in **Table 1**. Without additional information, we do not observe that either non-oracle DQN consistently outperforms the PID across the three patients. The DQN using a GRU outperformed the CNN, slightly outperforming the PID for Adolescent 1 (mean average risk 1.59 *vs.* 2.02) and performing slightly worse on Child 3 and Adult 1 (respectively, mean average risk 11.53 *vs.* 11.11, and mean average risk 12.55 *vs.* 12.30).

### 4.3. Ability to Adapt to Meals

One of the main potential advantages of RL is its ability to adapt to underlying behavioral patterns. To investigate this potential benefit, we explored changing the meal schedule generation procedure outlined in **Algorithm 1** for Adult 1. We removed the 'snack' meals (those with occurrence prob-

*Table 1.* Average risk over 10 days of simulation. Each entry contains the result from three random seeds (sorted low to high). The approach with the best average score is underlined, the second best is bolded. Across all patients the Oracle-Q network performs the best. Among the realistic methods, the PID has the best average performance in Child 3 and Adult 1, and the GRU-Q has the best average performance in Adolescent 1.

| Person | Child#003 | Adolescent#001 | Adult#001 |
|---|---|---|---|
| BB | 13.69, 14.38, 15.93 | 4.68, 4.83, 5.01 | 14.00, 15.51 , 15.61 |
| PID | **11.02, 11.08, 11.23** | 1.90, 2.04, 2.12 | **12.00, 12.39, 12.53** |
| Oracle-Q | 7.79, 7.81, 8.74 | 0.98, 1.01, 1.15 | 8.67, 8.94, 9.10 |
| CNN-Q | 11.45, 12.78, 13.32 | 2.08, 3.45, 4.38 | 12.20, 12.45, 15.95 |
| GRU-Q | 9.83, 11.41, 13.38 | **1.46, 1.62, 1.70** | 12.20, 12.56, 12.88 |

abilities of 0.3) and set all meal occurrence probabilities to 1 and meal amount standard deviations to 0, leaving us with 3 fixed sized meals occurring at variable times throughout the day. We then evaluated both the PID model and the CNN-based DQN model on 3 variations of this environment, characterized by the standard deviation of the meal times (either 0.1, 1, or 10 hours). The results are presented in **Figure 3**. We observe that the PID outperforms the DQN at baseline, but it is unable to leverage the information contained in the more regular meal schedule. The DQN is able to do this, and as a result outperforms the PID when meal time standard deviation is 0.1 hours. Note that for these additional experiments, the DQN was trained to fewer epochs (200) than presented in **Table 1**.

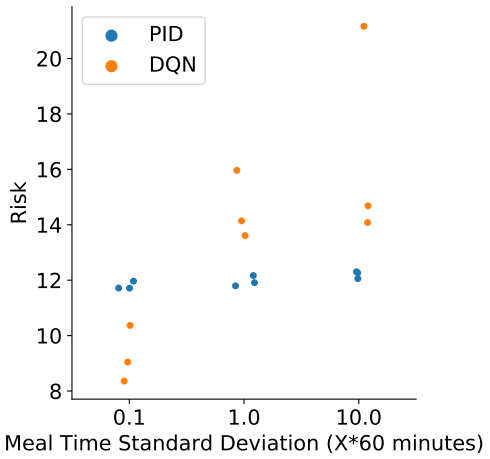


*Figure 3.* Average risk over 10 days for Adult 1 using different meal schedules. The x-axis is meal time standard deviation in multiples of 60 minutes, at lower values meals occur at more predictable times.

## 5. Discussion and Conclusion

In this work, we examined the use of deep model-free RL for learning automated blood glucose control algorithms. When given information about the ground truth state a deep Q network (DQN) was able to outperform baseline approaches. Furthermore, even without access to the ground-truth state, a DQN outperformed the baselines, when there were clear patterns in the meal schedule that could be exploited. Though promising, the DQN approaches did not consistently outperform the PID controller across all settings.

There are several factors that could be contributing to the poor performance including: the discretization of the action space, how we define the reward, and finally potential noise in the input signal. First, Q-learning requires a discretized action space; this could result in the over administration (or under administration) of insulin in some cases. In the future, we plan to investigate policy gradient methods that can take advantage of a continuous action space. Second, we define a reward function based on risk. Though, optimizing this risk function should lead to tight glucose control, it could lead to excess insulin utilization (as its use is unpenalized). Future work could consider resource-aware variants of this reward. Third, it should be noted that blood glucose data collected by CGMs are only noisy approximations of actual blood glucose levels. Recent advances in CGM technology has helped to reduce this noise (Shah et al., 2018), but it is still a concern. Our simulation includes CGM sensor noise, though in practice, we do not find that it drastically affects performance. Nonetheless, additional preprocessing, applied before the data are used as input to the algorithm, could improve performance.

Beyond the performance of the learned policies, across our experiments, we found that well over a thousand days of simulation data were required when training our deep approaches. While this is not an issue in the simulated environment, it would be infeasible to apply such approaches to real individuals. Model-based RL could be explored as a more sample efficient alternative to model-free RL. This is particularly promising given the existence of reasonable blood glucose models to serve as a starting point.

Finally, we emphasize that blood glucose control is a safety-critical application. An incorrect dose of insulin could lead to life-threatening situations. Importantly, the proposed approach, though promising, is not ready for deployment.

As shown by the worst-case performance of the Oracle Q method in **Figure 2**, our current approach can fail catastrophically. Going forward, there are several approaches that could be investigated to guarantee acceptable worst-case performance. Using the notion of 'shielding' from (Alshiekh et al., 2018), hard limits on insulin informed by blood glucose levels could prevent catastrophic hypoglycemia. Though this, in turn, could limit controller effectiveness in response to rapidly increasing glucose levels. Additionally, approaches that incrementally modify existing safe policies can limit worst-case performance and lead to safer control (Berkenkamp et al., 2017).

## References


Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., and Topcu, U. Safe reinforcement learning via shielding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Atlas, E., Nimri, R., Miller, S., Gurmberg, E. A., and Phillip, M. MD-Logic Artificial Pancreas System: A Pilot Study in Adults with Type 1 Diabetes Mellitus. *Diabetes Care*, February 2010. ISSN 0149-5992, 1935-5548. doi: 10.2337/dc09-1830. URL http://care.diabetesjournals.org/content/early/2010/01/29/dc09-1830.

Bequette, B. W. Algorithms for a closed-loop artificial pancreas: The case for model predictive control. 7(6):1632–1643. ISSN 1932-2968, 1932-2968. doi: 10.1177/193229681300700624. URL http://journals.sagepub.com/doi/10.1177/193229681300700624.

Bequette, B. W. A Critical Assessment of Algorithms and Challenges in the Development of a Closed-Loop Artificial Pancreas. *Diabetes Technology & Therapeutics*, 7(1):28–47, February 2005. ISSN 1520-9156, 1557-8593. doi: 10.1089/dia.2005.7.28. URL http://www.liebertpub.com/doi/10.1089/dia.2005.7.28.

Bergman, R. N. Toward physiological understanding of glucose tolerance: minimal-model approach. *Diabetes*, 38(12):1512–1527, 1989.

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pp. 908–918, 2017.

Bothe, M. K., Dickens, L., Reichel, K., Tellmann, A., Ellger, B., Westphal, M., and Faisal, A. A. The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas. *Expert Review of Medical Devices*, 10(5):661–673, September 2013. ISSN 1743-4440, 1745-2422. doi: 10.1586/17434440.2013.827515. URL http://www.tandfonline.com/doi/full/10.1586/17434440.2013.827515.

Clarke, W. and Kovatchev, B. Statistical tools to analyze continuous glucose monitor data. 11. ISSN 1520-9156, 1557-8593. doi: 10.1089/dia.2008.0138. URL http://www.liebertpub.com/doi/10.1089/dia.2008.0138.

Clavera, I., Nagabandi, A., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt: Meta-learning for model-based control. *arXiv preprint arXiv:1803.11347*, 3, 2018.

Cobelli, C., Federspil, G., Pacini, G., Salvan, A., and Scandellari, C. An integrated mathematical model of the dynamics of blood glucose and its hormonal control. *Mathematical Biosciences*, 58(1):27–60, 1982.

Coffen, R. D. and Dahlquist, L. M. Magnitude of type 1 diabetes self-management in youth health care needs diabetes educators. *The Diabetes Educator*, 35(2):302–308, 2009.

Control, D., Group, C. T. R., et al. Resource utilization and costs of care in the diabetes control and complications trial. *Diabetes Care*, 18(11):1468–1478, 1995.

DeFronzo, R. A., Ferrannini, E., Alberti, K. G. M. M., Zimmet, P., and Alberti, G. *International Textbook of Diabetes Mellitus, 2 Volume Set*. John Wiley & Sons, May 2015. ISBN 978-0-470-65861-1. Google-Books-ID: h5WDBgAAQBAJ.

Fox, I., Ang, L., Jaiswal, M., Pop-Busui, R., and Wiens, J. Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1387–1395. ACM. ISBN 978-1-4503-5552-0. doi: 10.1145/3219819.3220102. URL http://doi.acm.org/10.1145/3219819.3220102.

Garg, S. K., Weinzimer, S. A., Tamborlane, W. V., Buckingham, B. A., Bode, B. W., Bailey, T. S., Brazg, R. L., Ilany, J., Slover, R. H., Anderson, S. M., Bergenstal, R. M., Grosman, B., Roy, A., Cordero, T. L., Shin, J., Lee, S. W., and Kaufman, F. R. Glucose Outcomes with the In-Home Use of a Hybrid Closed-Loop Insulin Delivery System in Adolescents and Adults with Type 1 Diabetes. *Diabetes Technology & Therapeutics*, 19(3):155–163, January 2017. ISSN 1520-9156. doi: 10.1089/dia.2016.0421. URL https://www.liebertpub.com/doi/full/10.1089/dia.2016.0421.

Kadish, A. H. Automation Control of Blood Sugar. I. a Servomechanism for Glucose Monitoring and Control. *The American journal of medical electronics*, 3:82–86, 1964.

Kerl, M. E. Diabetic ketoacidosis: pathophysiology and clinical and laboratory presentation. *Compendium*, 23(3): 220–228, 2001.

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., and Faisal, A. A. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, pp. 1, 2018.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. 518(7540):529–533. ISSN 0028-0836. doi: 10.1038/nature14236. URL http://www.nature.com/nature/journal/v518/n7540/abs/nature14236.html.

Pinsker, J. E., Lee, J. B., Dassau, E., Seborg, D. E., Bradley, P. K., Gondhalekar, R., Bevier, W. C., Huyett, L., Zisser, H. C., and Doyle, F. J. Randomized Crossover Comparison of Personalized MPC and PID Control Algorithms for the Artificial Pancreas. *Diabetes Care*, pp. dc152344, June 2016. ISSN 0149-5992, 1935-5548. doi: 10.2337/dc15-2344. URL http://care.diabetesjournals.org/content/early/2016/06/10/dc15-2344.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Ruiz, J. L., Sherr, J. L., Cengiz, E., Carria, L., Roy, A., Voskanyan, G., Tamborlane, W. V., and Weinzimer, S. A. Effect of Insulin Feedback on Closed-Loop Glucose Control: A Crossover Study. *Journal of Diabetes Science and Technology*, 6(5):1123–1130, September 2012. ISSN 1932-2968. doi: 10.1177/193229681200600517. URL https://doi.org/10.1177/193229681200600517.

Shah, V. N., Laffel, L. M., Wadwa, R. P., and Garg, S. K. Performance of a factory-calibrated real-time continuous glucose monitoring system utilizing an automated sensor applicator. *Diabetes Technology & Therapeutics*, 20(6): 428–433, 2018.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Steil, G. M. Algorithms for a Closed-Loop Artificial Pancreas: The Case for Proportional-Integral-Derivative Control. *Journal of Diabetes Science and Technology*, 7(6):1621–1631, November 2013. ISSN 1932-2968, 1932-2968. doi: 10.1177/193229681300700623. URL http://journals.sagepub.com/doi/10.1177/193229681300700623.

Thomas, P. S. and Brunskill, E. Importance sampling with unequal support. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Tuomilehto, J. The emerging global epidemic of type 1 diabetes. *Current diabetes reports*, 13(6):795–804, 2013.

Visentin, R., Dalla Man, C., Kovatchev, B., and Cobelli, C. The university of virginia/padova type 1 diabetes simulator matches the glucose traces of a clinical trial. *Diabetes technology & therapeutics*, 16(7):428–434, 2014.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Xie, J. Simglucose, 2018. URL https://github.com/jxx123/simglucose.

You, W.-P. and Henneberg, M. Type 1 diabetes prevalence increasing globally and regionally: the role of natural selection and life expectancy at birth. *BMJ Open Diabetes Research and Care*, 4(1):e000161, 2016.

Zhu, T., Li, K., Herrero, P., Chen, J., and Georgiou, P. A deep learning algorithm for personalized blood glucose prediction. *IJCAI Knowledge Discovery in Healthcare Data Workshop*, 2018.