Deep Reinforcement Learning for Closed-Loop Blood Glucose Control

Ian Fox IFOX@UMICH.EDU
Department of Computer Science Engineering
University of Michigan
Ann Arbor, MI, USA

Joyce Lee
Department of Pediatrics and Communicable Diseases
University of Michigan
Ann Arbor, MI, USA

Rodica Pop-Busui
Department of Internal Medicine
University of Michigan
Ann Arbor, MI, USA

Jenna Wiens WIENSJ@UMICH.EDU
Department of Computer Science Engineering
University of Michigan
Ann Arbor, MI, USA

Abstract

People 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 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 develop 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 highlight the flexibility of RL approaches, demonstrating how they can adapt to new individuals with little additional data. On over 2.1 million hours of data from 30 simulated patients, our RL approach outperforms baseline control algorithms: leading to a decrease in median glycemic risk of nearly 50% from 8.34 to 4.24 and a decrease in total time hypoglycemic of 99.8%, from 4,610 days to 6. Moreover, these approaches are able to adapt to predictable meal times (decreasing average risk by an additional 24% as meals increase in predictability). This work demonstrates the potential of deep RL to help people with T1D manage their blood glucose levels without requiring expert knowledge. All of our code is publicly available, allowing for replication and extension.[1]

1. https://gitlab.eecs.umich.edu/mld3/rl4bg

1. Introduction

Type 1 diabetes (T1D) is a chronic disease affecting 20-40 million people worldwide (You and Henneberg, 2016), and its incidence is increasing (Tuomilehto, 2013). People with T1D cannot produce insulin, a hormone that controls blood glucose levels, and must monitor their blood glucose levels and manually administer insulin doses as needed. Administering too little insulin can lead to hyperglycemia (high blood glucose levels), which results in chronic health complications (Diabetes Control and Complications Trial Research Group, 1995), while administering too much insulin results in hypoglycemia (low blood glucose levels), leading to coma or death. Getting the correct dose requires careful measurement of glucose levels and carbohydrate intake, resulting in at least 15-17 data points a day. When using a continuous glucose monitor (CGM), this can increase to over 300 data points, or a blood glucose reading every 5 minutes (Coffen and Dahlquist, 2009).

CGMs with an insulin pump, a device that delivers insulin, can be used with a closed-loop controller as an 'artificial pancreas' (AP). 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 hybrid closed-loop approaches require accurate meal announcements to maintain glucose control.

In this work, we investigate deep reinforcement learning (RL) for blood glucose control. RL is a promising solution, as it is well-suited to learning complex behavior and readily adapts to changing domains (Clavera et al., 2018). We hypothesize that deep RL, the combination of RL with a deep neural network, will be able to accurately infer latent meals to control insulin. Furthermore, as RL is a learning-based approach, we hypothesize that RL will adapt to predictable meal schedules better than baseline approaches.

The fact that RL is learning-based means it requires data to work effectively. Unlike many other health settings, there are credible simulators for blood glucose management (Visentin et al., 2014). Having a simulator alleviates many concerns of applying RL to health problems (Gottesman et al., 2018, 2019). However, that does not mean RL for blood glucose control is straightforward, and, in this paper, we identify and address several challenges. To the best of our knowledge, we present the first deep RL approach that achieves human-level performance in controlling blood glucose without requiring meal announcements.

Generalizable Insights about Machine Learning in the Context of Healthcare

Applying deep RL to blood glucose management, we encountered challenges broadly relevant for RL in healthcare. As such, we believe our solutions and insights, outlined below, are broadly relevant as well.

- The range of insulin and carbohydrate requirements across patients makes it difficult to find a single action space that balances the needs of rapid insulin administration and safety. Indeed, many health problems involve changes in action distributions across patients (*e.g.* in anesthesia dosing (Bouillon and Shafer, 1998)). To solve this problem, we present a robust patient-specific action space that naturally encourages safer policies.

- We found several pitfalls in evaluating our proposed approach that led to unrealistic performance estimates. To address this issue, we used validation data to perform

careful model selection, and used extensive test data to evaluate the quality of our models. In RL, it is typical to report performance on the final trained model (without model selection) over a handful of rollouts. Our experiments demonstrate the danger of this approach.

- Deep RL has been shown to be unstable (Irpan, 2018; Henderson et al., 2018), often achieving poor worst-case performance. This is unacceptable for safety-critical tasks, such as those in healthcare. We found that a combination of simple and widely applicable approaches stabilized performance. In particular, we used a safety-augmented reward function, realistic randomness in training data, and random restarts to train models that behaved safely over thousands of days of evaluation.
- Finally, unlike game settings where one has ability to learn from hundreds of thousands of hours of interaction, any patient-specific model must be able to achieve strong performance using a limited amount of data. We show that a simple transfer learning approach can be remarkably sample efficient and can even surpass the performance of models trained from scratch.

2. Background and Related Work

This work develops and applies techniques in reinforcement learning to the problem of blood glucose management. To frame this work, we first provide a brief introduction to RL, both in general and specifically applied to problems in healthcare. We then discuss how RL and other approaches have been used for blood glucose control and present an overview on blood glucose simulation.

2.1. Reinforcement Learning

RL is an approach to optimize sequential decision making in an environment, which is typically assumed to follow a Markov Decision Process (MDP). An MDP is characterized by a 5-tuple (S, A, P, R, γ), where $s \in S$ are the states of the environment, $a \in A$ are actions that can be taken in the environment, the transition function $P : (s, a) \to s'$ defines the dynamics of the environment, the reward function $R : (s, a) \to r \in \mathbb{R}$ defines the desirability of state-action pairs, and the discount factor $\gamma \in [0, 1]$ determines the tradeoff between the value of immediate and delayed rewards. The goal in RL is to learn a policy $\pi : s \to a$, or function mapping states to actions, that maximizes the expected cumulative reward, or:

$$\arg\max_{\pi \in \Pi} \sum_{t=1}^{\infty} E_{s_t \sim P(s_{t-1}, a_{t-1})}[\gamma^t R(s_t, \pi(s_t))], \tag{1}$$

where Π is the space of possible policies and $s_0 \in S$ is the starting state. The state value function, $V(s)$, is the expected cumulative reward where $s_0 = s$. The state-action value function $Q(s, a) = R(s, a) + E_{s' \sim P(s,a)}[\gamma V(s')]$ extends the notion of value to state-action pairs.

2.2. Reinforcement Learning in Healthcare

In recent years, researchers have started to explore RL in healthcare. Examples include matching patients to treatment in the management of sepsis (Weng et al., 2017; Komorowski et al., 2018) and mechanical ventilation (Prasad et al., 2017). In addition, RL has been explored to provide contextual suggestions for behavioral modifications (Klasnja et al., 2019). Despite its successes, RL has yet to be fully explored as a solution for a closed-loop AP system (Bothe et al., 2013).

2.3. Algorithms for Blood Glucose Control

Among recent commercial AP products, proportional-integral-derivative (PID) control is the most common backbone (Trevitt et al., 2015). The simplicity of PID controllers make them easy to use, and in practice they achieve strong results (Steil, 2013). 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). A hybrid closed-loop controller adjusts baseline insulin rates but requires human intervention to control for the effect of meals. The main weakness of PID controllers is their reactivity. As a result, they often cannot react fast enough to meals, and thus rely on meal announcements (Garg et al., 2017). Additionally, without safety modifications, PID controllers can deliver too much insulin, triggering 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 more responsive and safer policies.

Previous works have examined RL for different aspects of blood glucose control. See Tejedor et al. (2020) for a recent survey. Many of these works investigated the use of RL to adapt existing insulin treatment regimens to learn a 'human-in-the-loop' policy (Ngo et al., 2018; Oroojeni Mohammad Javad et al., 2015; Sun et al., 2018). This contrasts with our setting, where we aim to learn a fully closed-loop policy.

Like our work, Daskalaki et al. (2010) and De Paula et al. (2015) focus on the task of closed-loop glucose control. Daskalaki et al. (2010) use direct future prediction to aid PID-style control, substituting the problem of RL with prediction. De Paula et al. (2015) use a policy-iteration framework with Gaussian process approximation and Bayesian active learning. While they tackle a similar problem, these works use a simple simulator and a fully deterministic meal routine for training and testing. In our experiments, we use an FDA-approved glucose simulator and a realistic non-deterministic meal schedule, significantly increasing the challenge.

2.4. Glucose Models and Simulation

Models of the glucoregulatory system are important for the development and testing of an AP (Cobelli et al., 1982). In our experiments, we use the UVA/Padova model (Kovatchev et al., 2009). This simulator models the glucoregulatory system as a nonlinear multi-compartment system, where glucose is generated in the liver, absorbed through the gut, and controlled by external insulin. A more detailed explanation can be found in Kovatchev et al. (2009). For reproducibility, we use an open-source version of the simulator that comes with 30 virtual patients (Xie, 2018). The parameter distribution of the patient population is determined by age, and the simulator comes with 10 children, adolescents, and adults each Kovatchev et al.

(2009). We combine the simulator with a non-deterministic meal schedule (**Appendix A.1**) to realistically simulate patient behavior.

3. Methods

We present a deep RL approach well suited for blood glucose control. In framing our problem, we pay special attention to the concerns of partial observability and safety. The issue of partial observability motivates us to use a maximum entropy control algorithm, soft actor-critic, combined with a recurrent neural network. Safety concerns inspire many aspects of our experimental setup, including our choice of action space, reward function, and evaluation metrics. We also introduce several strong baselines, both with and without meal announcements, to which we compare.

3.1. Problem Setup

We frame the problem of closed-loop blood glucose control as a partially-observable Markov decision process (POMDP) consisting of the 7-tuple $(S^*, O, S, A, P, R, \gamma)$. A POMDP generalizes the MDP setting described in **Section 2.1** by assuming we do not have access to the true environment states, here denoted $s^* \in S^*$, but instead observe noisy states $s \in S$ according to the (potentially stochastic) observation function: $O : s^* \to s$. This setting applies given the noise inherent in CGM sensors (Vettoretti et al., 2019) and our assumption of unobserved meals.

In our setting, the true states $\mathbf{s}_t^* \in S^*$ are the 13-dimensional simulator states, as described in Kovatchev et al. (2009). The stochastic observation function $O : \mathbf{s}_t^* \to b_t, i_t$ maps from the simulator state to the CGM observation b_t and insulin i_t administered. To provide temporal context, we augment our observed state space $\mathbf{s}_t \in S$ to include the previous 4 hours of CGM $\mathbf{b}^t$ and insulin data $\mathbf{i}^t$ at 5-minute resolution: $\mathbf{s}_t = [\mathbf{b}^t, \mathbf{i}^t]$ where:

$$\mathbf{b}^t = [b_{t-47}, b_{t-46}, \ldots, b_t], \mathbf{i}^t = [i_{t-47}, i_{t-46}, \ldots, i_t],$$

$b_t \in \mathbb{N}_{40:400}$, $i_t \in \mathbb{R}_+$, and $t \in \mathbb{N}_{1:288}$ represents a time index for a day at 5-minute resolution. Note that in our augmented formulation, the observation function no longer directly maps to observed states, as observed states incorporate significant amounts of historical data. We chose a 4-hour window, as we empirically found it led strong performance. We use a time resolution of 5 minutes to mimic the sampling frequency of many common CGMs. Actions $a_t \in \mathbb{R}_{\geq 0}$ are real positive numbers, denoting the size of the insulin bolus in medication units.

The transition function P, our simulator, consists of two elements: i) $M : t \to c_t$ is the meal schedule, and is defined in **Appendix A.1**, and ii) $G : (a_t, c_t, \mathbf{s}_t^*) \to (b_{t+1}, i_{t+1}, \mathbf{s}_{t+1}^*)$, where $c_t \in \mathbb{R}_{\geq 0}$ is the amount of carbohydrates input at time t and G is the UVA/Padova simulator (Kovatchev et al., 2009). Note that our meal schedule is patient-specific, and includes randomness in the daily number, size, and timing of meals.

The reward function $R : (\mathbf{s}_t, a_t) \to \mathbb{R}$ is defined as negative $-risk(b_t)$ where $risk$ is the Magni risk function:

$$risk(b) = 10 * (c_0 * \log(b)^{c_1} - c_2)^2, \tag{2}$$

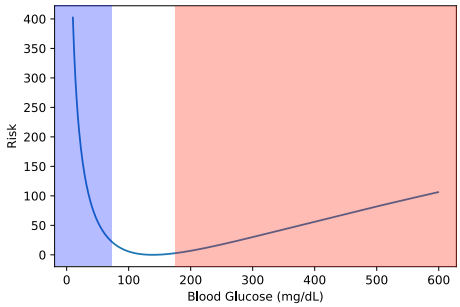


Figure 1: The risk function proposed in (Magni et al., 2007). The mapping between blood glucose values (in mg/dL, x-axis) and risk values (y-axis). Blood glucose levels corresponding to hypoglycemia are shown in the blue shaded region, the glucose range corresponding to hyperglycemia is shown in the red shaded region. This function identifies low blood glucose values as higher risk than high blood glucose values, which is sensible given the rapidity of hypoglycemia.

$c_0 = 3.35506$, $c_1 = 0.8353$, and $c_2 = 3.7932$ (Magni et al., 2007). These values are set such that $risk(70) = risk(280) = 25$, see **Figure 1**. Finally, we set $\gamma = 0.99$ for our experiments, a value we determined empirically on validation data in combination with the early termination penalty.

Considered in isolation, our reward could lead to dangerous behavior. As it is always negative, cumulative reward is maximized by ending the episode as quickly as possible, which occurs when glucose reaches unsafe levels. To avoid this, we add a termination penalty of $-1e5$ to trajectories that enter dangerous blood glucose regimes (blood glucose levels less than 10 or more than 1,000 mg/dL), countering the advantage of early termination. We investigated other reward functions, such as time in range or distance from a target blood glucose value, but found this reward worked best. It led to control schemes with less hypoglycemia, as low blood glucose is penalized more heavily than high glucose. Low glucose can occur quickly when large amounts of insulin are given without an accompanying meal. Given the lack of meal announcements and sensor noise in our setting, avoiding hypoglycemia was a significant challenge.

3.2. Soft Actor-Critic

We chose to use the soft actor-critic (SAC) algorithm to learn glucose control policies. We initially experimented with a Deep-Q Network approach (Mnih et al., 2015). However, choosing a discretized action space (as is required by Q-learning) that accounted for the range of insulin values across a day and allowed exploration proved impractical, as large doses of inappropriately timed insulin can be dangerous. Among continuous control algorithms, we selected SAC as it has been shown to be sample efficient and competitive (Haarnoja et al., 2018a). Additionally, maximum entropy policies like the ones produced by SAC can do well in partially observed settings like our own (Eysenbach and Levine, 2019).

SAC produces a stochastic policy $\pi : \mathbf{s}_t \rightarrow p(a)\ \forall a \in A$, which maps a state to a distribution over possible actions. Under SAC, the policy (or actor) is represented by

a neural network with parameters ϕ. Our network generates outputs $\mu, \log(\sigma)$ which parameterize a normal distribution $\mathcal{N}(\mu, \sigma)$. The actions are distributed according to a TanhNormal distribution, or $tanh(z), z \sim \mathcal{N}(\mu, \sigma)$. π_ϕ is trained to maximize the maximum entropy RL objective function:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t, a_t) \sim P(\mathbf{s}_{t-1}, \pi_\phi(\mathbf{s}_{t-1}))}[R(\mathbf{s}_t, a_t) + \alpha H(\pi_\phi(\cdot|\mathbf{s}_t))], \tag{3}$$

where entropy, H, is added to the expected cumulative reward to improve exploration and robustness (Haarnoja et al., 2018a). Intuitively, the return in **Equation 3** encourages a policy that can obtain a high reward under a variety of potential actions. The temperature hyperparameter α controls the tradeoff between reward and entropy. In our work, we set this using automatic temperature tuning (Haarnoja et al., 2018b). **Equation 3** is optimized by minimizing the KL divergence between the action distribution and the distribution induced by the state-action values:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\mathrm{D}_{\mathrm{KL}} \left(\pi_\phi \left(\cdot | \mathbf{s}_t \right) \| \frac{\exp\left(Q_\theta \left(\mathbf{s}_t, \cdot \right) \right)}{Z_\theta \left(\mathbf{s}_t \right)} \right) \right] \tag{4}$$

where $\mathcal{D}$ is a replay buffer containing previously seen $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ tuples, Z_θ is a partition function, and Q_θ is the state-action value function parameterized by a neural network (also called a critic). This means that our learned policy engages in probability matching, selecting an action with probability proportional to its expected value. This requires an accurate value function. To achieve this, Q_θ is trained by minimizing the temporal difference loss:

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\theta \left(\mathbf{s}_t, \mathbf{a}_t \right) - \hat{Q} \left(\mathbf{s}_t, \mathbf{a}_t \right) \right)^2 \right], \tag{5}$$

$$\hat{Q} \left(\mathbf{s}_t, \mathbf{a}_t \right) = r \left(\mathbf{s}_t, \mathbf{a}_t \right) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p} \left[V_{\overline{\psi}} \left(\mathbf{s}_{t+1} \right) \right]. \tag{6}$$

V_ψ is the soft value function parameterized by a third neural network, and $V_{\overline{\psi}}$ is the running exponential average of the weights of V_ψ over training. This is a continuous variant of the hard target network replication in (Mnih et al., 2015). V_ψ is trained to minimize:

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\frac{1}{2} \left(V_\psi \left(\mathbf{s}_t \right) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\rho} \left[Q_\theta \left(\mathbf{s}_t, \mathbf{a}_t \right) - \log \pi_\phi \left(\mathbf{a}_t | \mathbf{s}_t \right) \right] \right)^2 \right]. \tag{7}$$

In summary: we learn a policy that maps from states to a probability distribution over actions, the policy is parameterized by a neural network π_ϕ. Optimizing this network (**Equation 4**) requires an estimation of the soft state-action value function, we learn such an estimate Q_θ (**Equation 5**) together with a soft value function V_ψ (**Equation 7**). Additional details of this approach, including the gradient calculations, are given in (Haarnoja et al., 2018a). In keeping with previous work, when testing our policy we remove the sampling component, instead selecting the mean action $tanh(\mu)$. We replace the MSE temporal difference loss in **Equation 5** with the Huber loss, as we found this improved convergence.

3.2.1. Recurrent Architecture

Our network π_ϕ takes as input the past 4 hours of CGM and insulin data, requiring no human input (*i.e.*, no meal announcements). To approximate the true state $\mathbf{s}_t^*$ from the augmented state s we parameterize Q_θ, V_ψ, and π_ϕ using gated-recurrent unit (GRU) networks (Cho et al., 2014), as GRUs have been successfully used for glucose modeling previously (Fox et al., 2018; Zhu et al., 2018).

3.2.2. Patient-Specific Action Space

After the network output layer, actions are squashed using a *tanh* function. Note that this results in half the action space corresponding to negative values, which we interpret as administering no insulin. This encourages sparse insulin utilization and makes it easier for the network to learn to safely control baseline glucose levels. To ensure that the maximum amount of insulin delivered over a 5-minute interval is roughly equal to a normal meal bolus for each individual, we use the average ratio of basal to bolus insulin in a day (Kuroda et al., 2011) to calculate a scale parameter for the action space, $\omega_b = 43.2 * bas$, where bas is the default patient-specific basal insulin rate provided by Xie (2018).

3.3. Efficient Policy Transfer

One of the main disadvantages of deep RL is sample efficiency. Thus, we explored transfer learning techniques to efficiently transfer existing models to new patients. We refer to our method trained from scratch as RL-Scratch, and the transfer approach as RL-Trans. For RL-Trans, we initialize Q_θ, V_ψ and π_ϕ for each class of patients (children, adolescents, and adults) using fully trained networks from one patient of that source population (see **Appendix A.4**). We then fine-tune these networks on data collected from the target patient.

When fine-tuning, we modify the reward function by removing the termination penalty and adding a constant positive value (100) to all rewards. This avoids the negative reward issue discussed in **Section 3.1**. Removing the termination penalty increased the consistency of returns over training, allowing for a more consistent policy gradient. The additional safety provided by a termination penalty is not required as we begin with policies that are already stable. We found this simple approach for training patient-specific policies attains good performance while using far less patient-specific data.

3.4. Baselines

We examine three baseline methods for control: basal-bolus (**BB**), **PID** control, and PID with meal announcements (**PID-MA**). BB reflects an idealized human-in-the-loop control strategy, and PID reflects a common closed-loop AP algorithm. PID with meal announcements is based on current AP technology, and is a combination of the two, requiring regular human intervention. Finally, we consider an 'oracle' approach that has access to the true state s_t^*. This decouples the task of learning a policy from state inference, serving as a pseudo-upper bound on performance for our proposed approach.

3.4.1. Basal-Bolus Baseline

This baseline is designed to mimic human control and is an ideal depiction of how an individual with T1D controls their blood glucose. In this setting, we modify the state representation $\mathbf{s}_t$ to include a carbohydrate signal and a cooldown signal (explained below), and to remove the historical data, $\mathbf{s}_t = [b_t, i_t, c_t, cooldown]$. Note that the inclusion of a carbohydrate signal, or meal announcement, places the burden of providing accurate and timely estimates of meals on the person with diabetes. Each virtual patient in the simulator comes with the parameters necessary to calculate a reasonable basal insulin rate bas (the same value used in our action space definition), correction factor CF, and carbohydrate ratio CR. These three parameters, together with a glucose target b_g, define a clinician-recommended 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. The cooldown ensures that each meal is only corrected for once. Appropriate settings for these parameters can be estimated by endocrinologists, using previous glucose and insulin information (Walsh et al., 2011). The parameters for our virtual patient population, which are derived from a distribution validated by clinical trials (Kovatchev et al., 2009), are given in **Appendix A.2**.

3.4.2. PID Baseline

PID controllers are a common and robust closed-loop baseline (Steil, 2013). A PID controller operates by setting the control variable, a_t, to the weighted combination of three terms $a_t = k_P P(b_t) + k_I I(b_t) + k_D D(b_t)$ such that the process variable b_t (t is 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 $I(b_t) = \sum_{j=0}^{t}(b_j - b_g)$ corrects long-term deviations from the setpoint, and iii) the derivative term $D(b_t) = |b_t - b_{t-1}|$ uses the rate of change as a basic estimate of the future. The set point and the weights (also called gains) k_P, k_I, k_D are hyperparameters. To compare against the strongest PID controller possible, we tuned these hyperparameters using multiple iterations of grid-search with exponential refinement per-patient, minimizing risk. Our final settings are presented in **Appendix A.3**.

3.4.3. PID with Meal Announcements.

This baseline, which is similar to available hybrid closed loop AP systems (Garg et al., 2017; Ruiz et al., 2012), combines BB and PID into a control algorithm we call PID-MA. During meals, insulin boluses are calculated and applied as in the BB approach. The basal rate, instead of being fixed, is controlled by a PID algorithm, allowing for adaptation between meals. As above, we tune the gain parameters for the PID algorithm using sequential grid search to minimize risk.

3.4.4. Oracle Architecture

A deep RL approach to learning AP algorithms requires that the representation learned by the network contain sufficient information to control the system. As we are working with a simulator, we can explore the difficulty of this task in isolation, by replacing the

observed state $\mathbf{s}_t$ with the ground-truth state $\mathbf{s}_t^*$. Though unavailable in real-world settings, this representation decouples the problem of learning a policy from that of inferring the state. Here, Q_θ, V_ψ, and π_ϕ are fully connected with two hidden layers, each with 256 units. The network uses ReLU nonlinearities and BatchNorm (Ioffe and Szegedy, 2015).

3.5. Experimental Setup & Evaluation

To measure the utility of deep RL for the task of blood glucose control, we trained and tested our policies on data with different random seeds across 30 different simulated individuals.

Training and Hyperparameters. We trained our models separately for each patient. They were trained from scratch for 300 epochs for RL-Scratch, and fine-tuned for 50 epochs for RL-Trans. They were trained with batch size 256 and an epoch length of 20 days. We used an experience replay buffer of size 1e6 and a discount factor of 0.99. We found that extensive training from-scratch was required to obtain consistent performance across test runs. We also found that too small of an epoch length could lead to dangerous control policies. We optimized the parameters of Q_θ, V_ψ and π_ϕ using Adam with a learning rate of $3E-4$. All deep networks were composed of two layers of GRU cells with a hidden state size of 128, followed by a fully-connected output layer. All network hyperparameters, including number and size of layers, were optimized on training seeds on a subset of the simulated patients for computational efficiency. Our networks were initialized using PyTorch defaults.

Evaluation. We measured the performance of π_ϕ on 10 days of validation data after each training epoch. After training, we evaluated on test data using the model parameters from the best epoch as determined by the validation data. While this form of model selection is not typical for RL, we found it led to significant changes in performance (see **Section 4.2.2**). Our model selection procedure first filters out runs that could not control blood glucose within safe levels over the validation run (glucose between 30-1000 mg/dL), then selects the epoch that achieved the lowest risk. We tested each patient-specific model on 1000 days of test data, broken into 100 independent 10 day rollouts. We trained and evaluated each approach 3 times, resulting in 3000 days of evaluation per method per person.

We evaluated approaches using i) risk, the average Magni risk calculated over the 10-day test rollout, ii) % time spent euglycemic (blood glucose levels between 70-180 mg/dL), iii) % time hypo/hyperglycemic (blood glucose lower than 70 mg/dL or higher than 180 mg/dL respectively), and iv) % of rollouts that resulted in a catastrophic failure, which we define as a run that achieves a minimal blood glucose level below 5 mg/dL (at which point recovery becomes highly unlikely). Note that while catastrophic failures are a major concern, our simulation process does not consider consuming carbohydrates in reaction to low blood glucose levels. This is a common strategy to avoid dangerous hypoglycemia in real life, and thus catastrophic failures, while serious, are manageable. The random seeds controlling noise, meals, and all other forms of randomness, were different between training, validation, and test runs. We test the statistical significance of differences between methods using Mood's median test for all metrics except for catastrophic failure rate, for which we use a binomial test.

4. Experiments and Results

Our experiments are divided into two broad categories: i) experiments showing the benefits of deep RL for blood glucose control relative to baseline approaches and ii) the challenges of using deep RL in this setting, and how to overcome them.

Throughout our experiments, we consider 3 variants of RL methods: i) RL-Scratch, our approach trained from scratch on every individual, ii) RL-Trans, which fine-tunes an RL-Scratch model from an arbitrary child/adolescent/adult, and iii) RL-MA, which uses RL-Scratch trained using the automated meal boluses from BB or PID-MA. We also report results on an Oracle approach, which is trained and evaluated using the ground truth simulator state.

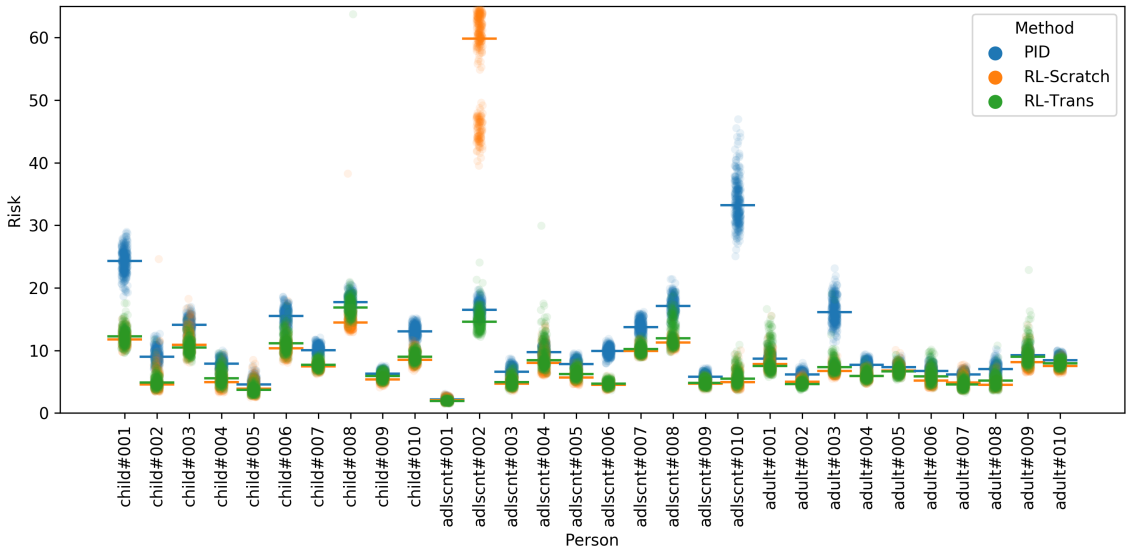


Figure 2: The risk over 10 days for different simulated patients using methods that do not require meal announcements. Each point corresponds to a different random test seed that controls the meal schedule and sensor noise, and the line indicates the median performance for each method on each patient. Results are presented across 3 random training seeds, controlling model initialization and randomness in training. We observe that, although there is a wide range in performance across and within individuals, The RL approaches tend to outperform PID.

4.1. Advantages of Deep RL

We compare our deep RL approaches to baselines with and without meal announcements across several metrics (**Section 4.1.1**). We then investigate two hypotheses for why deep RL is well suited to the problem of glucose management without meal announcements:

- the high-capacity neural network, integral to the RL approaches, is able to quickly infer when meals occur (**Section 4.1.2**), and

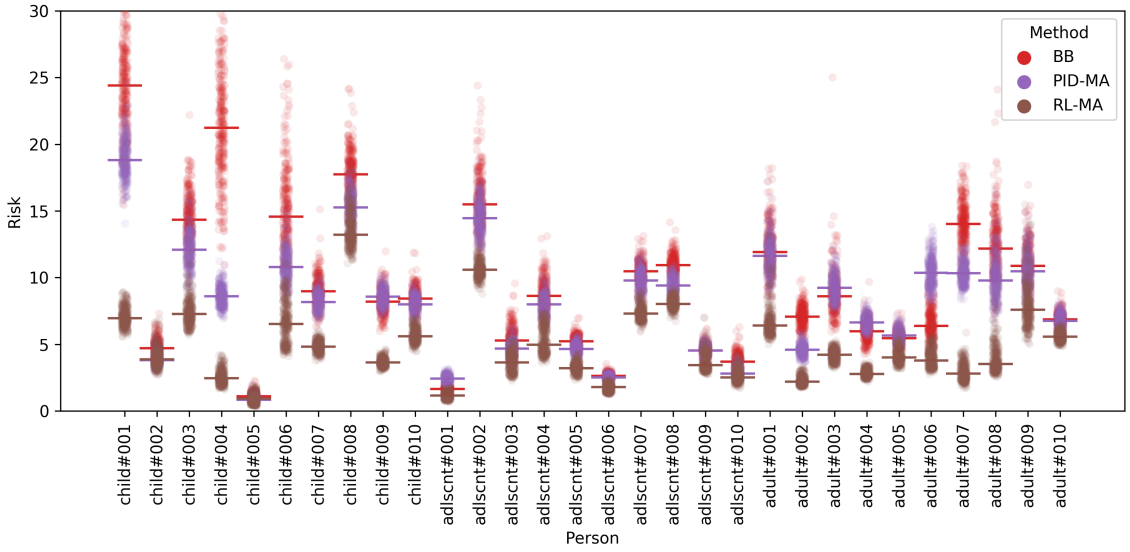


Figure 3: The risk over 10 days using methods that require meal announcements. PID-MA tends to outperform BB, and RL-MA outperforms PID-MA.

- the learning-based approach is able to adapt to predictable meal schedules better than a PID controller (**Section 4.1.3**).

4.1.1. Deep RL vs. Baseline Approaches

A comparison between the PID baseline to the RL approaches is presented in **Figure 2**. Each point represents a different test rollout by a policy. For the RL approaches, the performance of each method is reported as the combination of 3 random training restarts. Among the 3 methods that do not require meal announcements, RL-Scratch performs best across patients (average rank 1.33), followed by RL-Trans (average rank 1.7), then PID (average rank 2.97). For each individual, we rank the 3 approaches in terms of median risk. We calculate average rank by taking the mean of each approach's rankings across all 30 individuals. Note that RL-Scratch, while achieving strong performance overall, reliably performs poorly on adolescent#002. We discuss this issue in **Appendix A.5**.

One major advantage of our proposed approach is its ability to achieve strong performance without meal announcements. This does not mean that it does not benefit from meal announcements, as shown in **Figure 3**. Among the 3 methods that require meal announcements, RL-MA performs best (average rank 1.07), followed by PID-MA (average rank 2.13) then BB (average rank 2.8).

We examine additional metrics in the results presented in **Table 1**. The difference between results that are bold, or bold and underlined, and the next best non-bold result (excluding RL-Oracle) are statistically significant with $p < 0.001$. We observe that RL-MA equals or surpasses the performance of all non-oracle methods on all metrics, except for % time spent hyperglycemia. Interestingly, all RL variants achieve lower median risk than PID-MA, which requires meal announcements. This is because the RL approaches achieve

low levels of hypoglycemia, which the risk metric heavily penalizes (see **Figure 1**). Note that all methods, including PID-MA, were optimized to minimize this metric. Across patients, the RL methods achieve approximately 60-80% time euglycemic, compared with $52\% \pm 19.6\%$ observed in real human control (Ayano-Takahara et al., 2015). These results suggest that deep RL could be a valuable tool for closed-loop or hybrid closed-loop AP control.

Table 1: Median risk, percent of time Eu/Hypo/Hyperglycemic, and failure rate calculated using 1000 days of simulation broken into 100 independent 10-day rollouts for each of 3 training seeds for 30 patients, totaling 90k days of evaluation (with interquartile range). Lower Magni Risk, Hypoglycemia, and Hyperglycemia are better, higher Euglycemia is better. Hybrid and Non-closed loop approaches (requiring meal announcements) are indicated with *. Approaches requiring a fully observed simulator state are indicated with †. The non-oracle approach with the best average score is in bold and underlined, the best approach that does not require meal announcements is in bold.

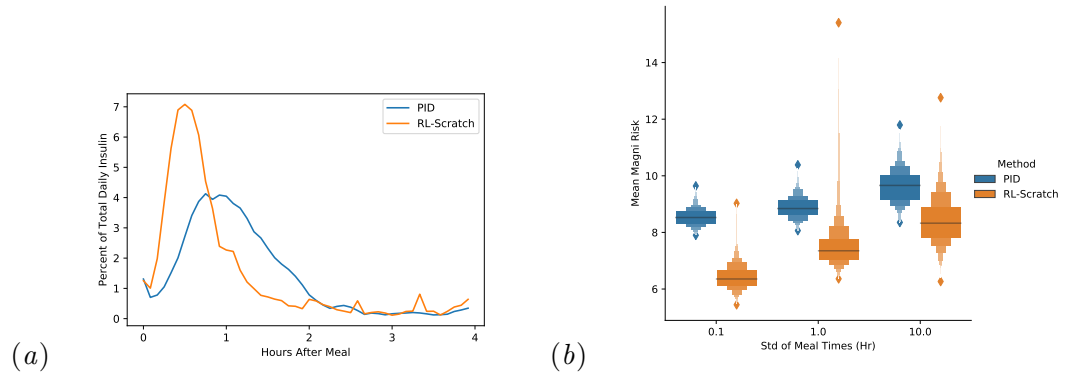
		Risk ↓	Euglycemia (%) ↑	Hypoglycemia (%) ↓	Hyperglycemia (%) ↓	Failure (%) ↓
No MA	PID	8.86 (6.8-14.3)	71.68 (65.9-75.9)	1.98 (0.3-5.5)	**24.71 (21.1-28.6)**	0.12
	RL-Scratch	**6.50 (4.8-9.3)**	**72.68 (67.7-76.2)**	**0.73 (0.0-1.8)**	26.17 (23.1-30.6)	**0.07**
	RL-Trans	6.83 (5.1-9.7)	71.91 (66.6-76.2)	1.04 (0.0-2.5)	26.60 (22.7-31.0)	0.22
MA	BB*	8.34 (5.3-12.5)	71.09 (62.2-77.9)	2.60 (0.0-7.2)	23.85 (17.0-32.2)	0.26
	PID-MA*	8.31 (4.7-10.4)	76.54 (70.5-82.0)	3.23 (0.0-8.8)	**18.74 (12.9-23.2)**	**0.00**
	RL-MA*	**4.24 (3.0-6.5)**	**77.12 (71.8-83.0)**	**0.00 (0.0-0.9)**	22.36 (16.6-27.7)	**0.00**
	RL-Oracle†	3.58 (1.9-5.4)	78.78 (73.9-84.9)	0.00 (0.0-0.0)	21.22 (15.1-26.1)	0.01

4.1.2. Detecting Latent Meals

Our approach achieves strong blood glucose control without meal announcements, but how much of this is due to the ability to react to meals? To investigate this, we looked at the total proportion of insulin delivered on average after meals for PID and RL-Scratch, shown in **Figure 4a**. A method able to infer meals should use insulin rapidly after meals, as the sooner insulin is administered the faster glycemic spikes can be controlled while avoiding hypoglycemia. We observe that RL-Scratch administers the majority of its post-meal bolus within one hour of a meal, whereas PID requires over 90 minutes, suggesting RL-Scratch can indeed better infer meals. Interestingly, when considering the percentage of total daily insulin administered in the hour after meals, RL-Scratch responds even more aggressively than BB or PID-MA (54.7% *vs.* 48.5% and 47.3% respectively). This demonstrates that our RL approach is able to readily react to latent meals shortly after they have occurred.

4.1.3. Ability to Adapt to Predictable Meal Schedules

We hypothesize that one advantage of RL is its ability to compensate for predictable variations (such as meals) in the environment, improving control as the environment becomes more predictable. To test this benefit, we changed the meal schedule generation procedure outlined in **Algorithm 1** (**Appendix A.1**) for Adult 1. We removed the small 'snack' meals, set all meal occurrence probabilities to 1, and made meal amounts constant (*i.e.*, each



(a) (b)

Figure 4: a) The average amount of insulin (in percent of total daily insulin) provided after meals for PID and RL-Scratch (note: RL-Trans, unshown, is very similar to RL-Scratch). RL-Scratch is able respond to meals more quickly than PID, with insulin peaking 30 minutes post-meal as opposed to roughly 60 minutes for PID. Additionally, the RL approach finishes delivering most post-meal insulin after 1hr, PID takes over 90 minutes. b) The distribution of average risk scores over 300 10-day rollouts for Adult 1 using meal schedules with varying amounts of predictability (meal time standard deviation). While PID performs better with more regularly spaced meals (median risk lowers from 9.66 at std=10 to 8.53 at std=0.1, a 12% decrease), RL-Scratch sees a larger proportional and absolute improvement (median risk lowers from 8.33 at std=10 to 6.36 at std=0.1, a 24% decrease).

day Adult 1 consumes an identical set of meals). We then evaluated both the PID model and RL-Scratch on 3 variations of this environment, characterized by the standard deviation of the meal times (either 0.1, 1, or 10 hours). This tests the ability of each method to take advantage of patterns in the environment. As the standard deviation decreases, the task becomes easier for two reasons: i) there are fewer instances where two meals occur in quick succession, and ii) the meals become more predictable. The results are presented in **Figure 4b**. We observe that both methods improve in performance as the standard deviation decreases, likely due to (i). However, while RL-Scratch outperforms PID under all settings, the difference increases as the standard deviation of meal times decreases, suggesting RL is better able to leverage the predictability of meals. Specifically, mean risk decreases by roughly 12% for the PID approach (from 9.65 to 8.54), whereas it decreases nearly 24% for the RL approach (from 8.40 to 6.42). This supports our hypothesis that RL is better able to take advantage of predictable meal schedules.

4.2. Challenges for Deep RL

While in the previous section we demonstrated several advantages of using deep RL for blood glucose control, here we emphasize that the application of deep RL to this task and its evaluation are non-trivial. Specifically, in this section we:

- demonstrate the importance of our action space formulation for performance (**Section 4.2.1**),

- illustrate the critical need for careful and extensive validation, both for model selection and evaluation (**Section 4.2.2**).
- show that, applied naively, deep RL leads to an unacceptable catastrophic failure rate, and present three simple approaches to improve this (**Section 4.2.3**),
- address the issue of sample complexity with simple policy transfer (**Section 4.2.4**).

4.2.1. Developing an Effective Action Space

One challenging element of blood glucose control in an RL setting is defining the action space. Insulin requirements vary significantly by person (from 16 to 60 daily units in the simulator population we use), and throughout most of the day, insulin requirements are much lower than after meals. To account for these challenges, we used a patient-specific action space, where much of the space corresponds to delivering no insulin (discussed in **Section 3.2.1**). We perform an ablation study to test the impact of these decisions. On an arbitrarily chosen patient (child#001), we shifted the *tanh* output to remove the negative insulin space. This increased the catastrophic failure rate from 0% (on this patient) to 6.6%. On a challenging subset of 4 patients (indicated in **Appendix A.2**), we looked at the effect of removing the patient-specific action scaling ω_b. This resulted in a 13% increase in median risk from 9.92 to 11.23. These results demonstrate that a patient-specific action space that encourages conservative behavior can improve performance.

4.2.2. Potential Pitfalls in Evaluation

In our experiments, we observed two key points for model evaluation: i) while often overlooked in RL, using validation data for model selection during training was key to achieving good performance, and ii) without evaluating on far more data than is typical, it was easy to underestimate the catastrophic failure rate.

Performance instability necessitates careful model selection. Off-policy RL with function approximation, particularly deep neural networks, is known to be unstable (Sutton and Barto, 2018; Gottesman et al., 2018). As a result, we found it was extremely important to be careful in selecting which network (and therefore policy) to evaluate. In **Figure 5a**, we show the fluctuation in validation performance over training for Adult#009. While performance increases on average over the course of training (at least initially), there are several points where performance degrades considerably. **Figure 5b** shows how performance averaged over all patients changes depending on the approach used to select the policy for evaluation. When we simply evaluate using the final epoch, almost half of test rollouts end in a catastrophic failure. Surprisingly, even when we select the model that minimized risk on the validation set, nearly a fifth of rollouts fail. However, by first limiting our pool of models to those that achieve a minimum blood glucose level of at least 30 mg/dL over the validation data, we reduce the catastrophic failure rate to 0.07%. As performance instability has been noted in other RL domains (Islam et al., 2017; Henderson et al., 2018), this observation is likely relevant to other applications of deep RL in healthcare.

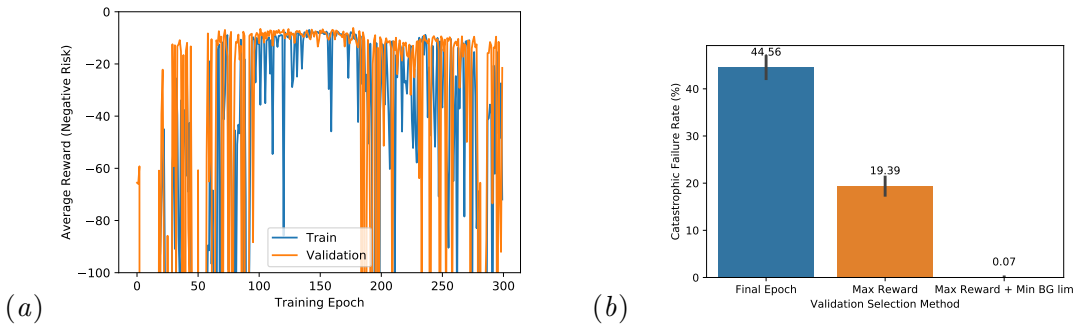


Figure 5: a) The training and validation curve (average reward) for adult#009. Note the periods of instability affect both training and validation performance. b) Catastrophic failure rate over all patients for 3 methods of model selection: i) selecting the final training epoch, ii) selecting the epoch that achieved minimal risk, and iii) selecting the minimal risk epoch that maintained blood glucose above 30 mg/dL. We see large differences in performance depending on the model selection strategy.

Extensive evaluation is necessary. Health applications are often safety critical. Thus, the susceptibility of deep RL to unexpected or unsafe behavior can pose a significant risk (Leike et al., 2017; Futoma et al., 2020). While ongoing work seeks to provide safety guarantees in control applications using deep learning (Achiam et al., 2017; Futoma et al., 2020), it is important that practitioners take every step to evaluate the safety of their approaches. While it is typical to evaluate RL algorithms on a small number of rollouts (Henderson et al., 2018), in our work we saw how easy it can be to miss unsafe behavior, even with significant testing. We examined the number of catastrophic failures that occurred using RL-Trans using different evaluation sets. Over our full evaluation set of 9,000 10-day rollouts, we observed 20 catastrophic failures (a rate of 0.22%). However, when we only evaluated using the first 5 test seeds, which is still over 12 years of data, we observed 0 catastrophic failures. Additionally, when we evaluated using 3-day test rollouts instead of 10, we only observed only 5 catastrophic failures (a rate of .05%), suggesting that longer rollouts result in a higher catastrophic failure rate. These results demonstrate that, particularly when dealing with noisy observations, it is critical to evaluate potential policies using a large number of different, lengthy rollouts.

4.2.3. Reducing Catastrophic Failures

Due to their potential danger, avoiding catastrophic failures was a significant goal of this work. The most direct approach we used was to modify the reward function, using a large termination penalty to discourage dangerous behavior. While unnecessary for fine-tuning policies, when training from scratch this technique was crucial. On a subset of 6 patients (see **Appendix A.4**), including the termination penalty reduced the catastrophic failure rate from 4.2% to 0.2%.

We found varying the training data also had a major impact on the catastrophic failure rate. During training, every time we reset the environment we used a different random seed

Table 2: Risk and percent of time Eu/Hypo/Hyperglycemic calculated for the RL approaches treating the 3 training seeds as different random restarts. The stability of the Scratch and Trans approaches improves relative to performance in Table 1.

	Risk ↓	Euglycemia (%) ↑	Hypoglycemia (%) ↓	Hyperglycemia (%) ↓	Failure Rate (%) ↓
RL-Scratch	6.39 (4.7-8.9)	72.96 (69.1-76.6)	0.62 (0.0-1.7)	25.96 (22.7-29.6)	0.00
RL-Trans	6.57 (5.0-9.3)	71.56 (67.0-75.7)	0.80 (0.0-1.9)	27.19 (23.4-31.2)	0.13
RL-MA	3.71 (2.7-6.3)	77.36 (72.7-83.2)	0.00 (0.0-0.5)	22.45 (16.7-26.9)	0.00

(which controls meal schedule and sensor noise). Note that the pool of training, validation, and test seeds were non-overlapping. On a challenging subset of 7 patients (described in **Appendix A.4**), we ran RL-Scratch with and without this strategy. The run that varied the training seeds had a catastrophic failure rate of 0.15%, the run that didn't had a 2.05% rate (largely driven by adult#009, who reliably had the highest catastrophic failure rate across experiments).

Other approaches can further improve stability. In **Table 1**, our RL results are averaged over three random restarts in training. This was done to demonstrate that our learning framework is robust to randomness in training data and model initialization. However, in a real application it would make more sense to select (using validation data) one model for use out of the random restarts. We apply this approach in **Table 2**, choosing the seed that obtained the best performance according to our model selection criteria. This improves all the RL methods. Most notably, it further reduces the catastrophic failure rate for the approaches without meal announcements (0.07% to 0% for RL-Scratch, and 0.22% to 0.13% for RL-Trans).

4.2.4. Sample Efficiency and Policy Transfer

While RL-Scratch achieves strong performance on average, it requires a large amount of patient-specific data: 16.5 years per patient. While RL-Trans reduced this amount, it still required over 2 years of patient-specific data, which for most health applications would be infeasible. Thus, we investigated how performance degrades as less data is used.

In **Figure 6**, we show the average policy performance by epoch for both RL-Scratch and RL-Trans relative to the PID controller. Note the epoch determines the maximum possible epoch for our model selection, not the actual chosen epoch. We see that far less training is required to achieve good performance with RL-Trans. In over 40% of rollouts, RL-Trans outperforms PID with no patient-specific data (median risk 10.31), and with 10 epochs of training (roughly half a year of data) RL-Trans outperforms PID in the majority of rollouts (59.6%; median risk 7.95).

Interestingly, the lack of patient-specific data does not appear to cause excessive catastrophic failures. With no patient-specific data the failure rate is 0%, after 5 epochs of training it has risen to .5%, and then declines over training to the final value of .22%. This implies two things: i) patient-specific training can increase the catastrophic failure rate, possibly by learning overly aggressive treatment policies, and ii) our current model selection procedure does not minimize the catastrophic failure rate. We do not observe this

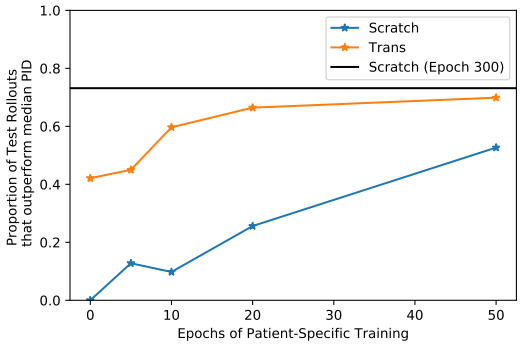


Figure 6: The proportion of test rollouts where RL-Scratch and RL-Trans outperform the median PID risk with different amounts of patient-specific training. We see that without any patient-specific data RL-Trans performs better than PID in 40% of rollouts. RL-Scratch requires a significant amount of patient-specific data before achieving comparable performance.

for RL-Scratch, where all epochs under 50 achieve a catastrophic error rate of over 17%. These results suggest that our simple transfer approach can be effective even with limited amounts of patient-specific data.

5. Conclusion

In this work, we demonstrated how deep RL can lead to significant improvements over alternative approaches to blood glucose control, with and without meal announcements (**Section 4.1.1**). We provide insight into why (**Section 4.1.2**) and when (**Section 4.1.3**) we would expect to see RL perform better. We demonstrated the importance of a robust action space in patient-specific tasks (**Section 4.2.1**), showed how careful and extensive validation is necessary for realistic evaluation of performance(**Section 4.2.2**), investigated several simple and general approaches to improving the stability of deep RL (**Section 4.2.3**), and developed a simple approach to reduce the requirements of patient-specific data (**Section 4.2.4**). While the main goal of this paper was to advance a clinical application, the challenges we encountered and the solutions they inspired are likely to be of interest to researchers applying RL to healthcare more broadly.

While our results in applying deep RL to blood glucose control are encouraging, they come with several limitations. First, our results are based on simulation. The simulator may not adequately capture variation across patients or changes in the glucoregulatory system over time. In particular, the virtual patient population the simulator comes equipped with does not differentiate individuals based on demographic information, such as gender and ethnicity. Thus, the applicability of our proposed techniques to all relevant patient populations cannot be assessed. However, as an FDA-approved substitute for animal trials (Kovatchev et al., 2009), success in using this simulator is a nontrivial accomplishment. 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, our choice of a 4-hour state space discourages learning long-term patterns or trends. In our environment, this did not reduce performance relative to a longer input history, but this could be important for managing blood glucose levels in more realistic simulators or real-world cases Visentin et al. (2018). Finally, we emphasize that blood glucose control is a safety-critical application. An incorrect dose of insulin could lead to life-threatening situations. Many of the methods we examined, even those that achieve good average performance, are susceptible to catastrophic failures. We have investigated several ways to minimize such failures, including modifying the reward function and selecting models across multiple random restarts. While the results from **Table 2** suggest these approaches mitigate catastrophic failures, the results of **Section 4.2.2** show such empirical evaluations can miss failure cases. To enable researchers to better explore and correct these limitations, we evaluated on an open-source simulator and made all the code required to reproduce our experiments publicly available.

Acknowledgments

This work was supported by JDRF. The views and conclusions in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of JDRF.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

Shiho Ayano-Takahara, Kaori Ikeda, Shimpei Fujimoto, Kanae Asai, Yasuo Oguri, Shin-ichi Harashima, Hidemi Tsuji, Kenichiro Shide, and Nobuya Inagaki. Carbohydrate intake is associated with time spent in the euglycemic range in patients with type 1 diabetes. *Journal of diabetes investigation*, 6(6):678–686, 2015.

Melanie K Bothe, Luke Dickens, Katrin Reichel, Arn Tellmann, Björn Ellger, Martin Westphal, and Ahmed A Faisal. 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.

MD Bouillon, Thomas and MD Shafer, Steven L. Does Size Matter? *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 89(3):557–560., 09 1998. ISSN 0003-3022.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP*, June 2014. arXiv: 1406.1078.

Ignasi Clavera, Anusha Nagabandi, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt: Meta-learning for model-based control. *arXiv preprint arXiv:1803.11347*, 3, 2018.

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

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

Elena Daskalaki, Luca Scarnato, Peter Diem, and Stavroula G. Mougiakakou. Preliminary results of a novel approach for glucose regulation using an actor-critic learning based controller. In *UKACC International Conference on Control 2010*, pages 1–5, 2010. doi: 10.1049/ic.2010.0287. ISSN: null.

Mariano De Paula, Gerardo G. Acosta, and Ernesto C. Martínez. On-line policy learning and adaptation for real-time personalization of an artificial pancreas. *Expert Syst. Appl.*, 42(4):2234–2255, 2015.

Diabetes Control and Complications Trial Research Group. Resource utilization and costs of care in the diabetes control and complications trial. *Diabetes Care*, 18(11):1468–1478, 1995.

Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

Ian Fox, Lynn Ang, Mamta Jaiswal, Rodica Pop-Busui, and Jenna Wiens. 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, pages 1387–1395. ACM, 2018. ISBN 978-1-4503-5552-0.

Joseph Futoma, Muhammad A Masood, and Finale Doshi-Velez. Identifying distinct, effective treatments for acute hypotension with soda-rl: Safely optimized diverse accurate reinforcement learning. *arXiv preprint arXiv:2001.03224*, 2020.

Satish K. Garg, Stuart A. Weinzimer, William V. Tamborlane, Bruce A. Buckingham, Bruce W. Bode, Timothy S. Bailey, Ronald L. Brazg, Jacob Ilany, Robert H. Slover, Stacey M. Anderson, Richard M. Bergenstal, Benyamin Grosman, Anirban Roy, Toni L. Cordero, John Shin, Scott W. Lee, and Francine R. Kaufman. 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.

Omer Gottesman, Fredrik Johansson, Joshua Meier, Jack Dent, Donghun Lee, Srivatsan Srinivasan, Linying Zhang, Yi Ding, David Wihl, Xuefeng Peng, et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1):16–18, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*, pages 1861–1870, July 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications. *arXiv:1812.05905 [cs, stat]*, December 2018b. arXiv: 1812.05905.

James Arthur Harris and Francis Gano Benedict. *A biometric study of basal metabolism in man*. Carnegie institution of Washington, 1919.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

Alex Irpan. Deep reinforcement learning doesn't work yet. https://www.alexirpan.com/2018/02/14/rl-hard.html, 2018.

Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

Predrag Klasnja, Shawna Smith, Nicholas J. Seewald, Andy Lee, Kelly Hall, Brook Luers, Eric B. Hekler, and Susan A. Murphy. Efficacy of contextually tailored suggestions for physical activity: A micro-randomized optimization trial of HeartSteps. *Annals of Behavioral Medicine*, 53(6):573–582, 2019.

Matthieu Komorowski, Leo A. Celi, Omar Badawi, Anthony C. Gordon, and A. Aldo Faisal. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, page 1, 2018.

Boris P Kovatchev, Marc Breton, Chiara Dalla Man, and Claudio Cobelli. 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.

Akio Kuroda, Hideaki Kaneto, Tetsuyuki Yasuda, Munehide Matsuhisa, Kazuyuki Miyashita, Noritaka Fujiki, Keiko Fujisawa, Tsunehiko Yamamoto, Mitsuyoshi Takahara, Fumie Sakamoto, Taka-aki Matsuoka, and Iichiro Shimomura. Basal insulin requirement is 30% of the total daily insulin dose in type 1 diabetic patients who use the insulin pump. *Diabetes Care*, 34(5):1089–1090, 2011.

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

Lalo Magni, Davide M. Raimondo, Luca Bossi, Chiara Dalla Man, Giuseppe De Nicolao, Boris Kovatchev, and Claudio Cobelli. Model predictive control of type 1 diabetes: An in silico trial. *Journal of diabetes science and technology (Online)*, 1(6):804–812, 2007.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

P. D. Ngo, S. Wei, A. Holubová, J. Muzik, and F. Godtliebsen. Reinforcement-learning optimal control for type-1 diabetes. In *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 333–336, 2018. doi: 10.1109/BHI.2018.8333436.

Mahsa Oroojeni Mohammad Javad, Stephen Agboola, Kamal Jethwani, Ibrahim Zeid, and Sagar Kamarthi. Reinforcement learning algorithm for blood glucose control in diabetic patients. In *Volume 14: Emerging Technologies; Safety Engineering and Risk Analysis; Materials: Genetics to Structures*, page V014T06A009. ASME, 2015. ISBN 978-0-7918-5757-1.

Jordan E. Pinsker, Joon Bok Lee, Eyal Dassau, Dale E. Seborg, Paige K. Bradley, Ravi Gondhalekar, Wendy C. Bevier, Lauren Huyett, Howard C. Zisser, and Francis J. Doyle. Randomized Crossover Comparison of Personalized MPC and PID Control Algorithms for the Artificial Pancreas. *Diabetes Care*, page dc152344, June 2016.

Niranjani Prasad, Li-Fang Cheng, Corey Chivers, Michael Draugelis, and Barbara E. Engelhardt. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*, 2017.

Jessica L. Ruiz, Jennifer L. Sherr, Eda Cengiz, Lori Carria, Anirban Roy, Gayane Voskanyan, William V. Tamborlane, and Stuart A. Weinzimer. Effect of Insulin Feedback on Closed-Loop Glucose Control: A Crossover Study. *Journal of Diabetes Science and Technology*, 6 (5):1123–1130, September 2012.

Garry M Steil. Algorithms for a closed-loop artificial pancreas: the case for proportional-integral-derivative control. *Journal of diabetes science and technology*, 7(6):1621–1631, 2013.

Q. Sun, M. Jankovic, J. Budzinski, B. Moore, P. Diem, C. Stettler, and S. G. Mougiakakou. A dual mode adaptive basal-bolus advisor based on reinforcement learning. *IEEE Journal of Biomedical and Health Informatics*, pages 1–1, 2018.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, second edition edition, 2018. ISBN 978-0-262-03924-6.

Miguel Tejedor, Ashenafi Zebene Woldaregay, and Fred Godtliebsen. Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial Intelligence in Medicine*, page 101836, 2020. ISSN 0933-3657. doi: 10.1016/j.artmed.2020.101836.

Sara Trevitt, Sue Simpson, and Annette Wood. Artificial pancreas device systems for the closed-loop control of type 1 diabetes. *Journal of Diabetes Science and Technology*, 10(3): 714–723, 2015.

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

Martina Vettoretti, Simone Del Favero, Giovanni Sparacino, and Andrea Facchinetti. Modeling the error of factory-calibrated continuous glucose monitoring sensors: application to dexcom g6 sensor data. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 750–753. IEEE, 2019.

Roberto Visentin, Chiara Dalla Man, Boris Kovatchev, and Claudio Cobelli. 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.

Roberto Visentin, Enrique Campos-Náñez, Michele Schiavon, Dayu Lv, Martina Vettoretti, Marc Breton, Boris P. Kovatchev, Chiara Dalla Man, and Claudio Cobelli. The UVA/Padova Type 1 Diabetes Simulator Goes From Single Meal to Single Day. *Journal of Diabetes Science and Technology*, 12(2):273–281, March 2018.

John Walsh, Ruth Roberts, and Timothy Bailey. Guidelines for optimal bolus calculator settings in adults. *Journal of Diabetes Science and Technology*, 5(1):129–135, 2011.

Wei-Hung Weng, Mingwu Gao, Ze He, Susu Yan, and Peter Szolovits. Representation and reinforcement learning for personalized glycemic control in septic patients. *NeurIPS 2017 ML4H Workshop*, 2017.

Jinyu Xie. Simglucose v0.2.1. Available: https://github.com/jxx123/simglucose, 2018.

Wen-Peng You and Maciej Henneberg. 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.

Taiyu Zhu, Kezhi Li, Pau Herrero, Jianwei Chen, and Pantelis Georgiou. A deep learning algorithm for personalized blood glucose prediction. *IJCAI Knowledge Discovery in Healthcare Data Workshop*, 2018.

Appendix A. Appendix

A.1. Harrison-Benedict Meal Generation Algorithm

See **Algorithm 1**. In short, this meal schedule calculates BMR for each simulated individual using the Harrison-Benedict equation (Harris and Benedict, 1919). This is used to estimate expected daily carbohydrate consumption, assuming individuals eat a reasonably low-carb diet where 45% of calories come from carbohydrates. Daily carbohydrates were divided between 6 potential meals: breakfast, lunch, dinner, and 3 snacks. The occurrence probability of the meal and expected size of the meal was set such that the expected number of carbs eaten per day matched the BMR-derived estimate. Note, in our experiments our implementation of of this meal schedule incorrectly estimated ages for many patients, particularly adults. This effect was fairly minor, resulting in no more than a 10% change in estimated BMR.

Algorithm 1 Generate Meal Schedule

Input: body weight w, age a, height h, number of days n
$BMR = 66.5 + (13.75 * w) + (5.003 * h) - (6.755 * a)$
$ExpectedCarbs = (BMR * 0.45)/4$ {45% of calories assumed from carbs, 4 calories per carb}
$MealOcc = [0.95, 0.3, 0.95, 0.3, 0.95, 0.3]$
$TimeLowerBounds = [5, 9, 10, 14, 16, 20] * 12$
$TimeUpperBounds = [9, 10, 14, 16, 20, 23] * 12$
$TimeMean = [7, 9.5, 12, 15, 18, 21.5] * 12$
$TimeStd = [1, .5, 1, .5, 1, .5] * 12$
$AmountMean = [0.250, 0.035, 0.295, 0.035, 0.352, 0.035] * ExpectedCarbs * 1.2$
$AmountStd = AmountMean * 0.15$
$Days = []$
for $i \in [1, \ldots, n]$ **do**
  $M = [0]_{j=1}^{288}$
  for $j \in [1, \ldots, 6]$ **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

Person	CR	CF	Age	TDI
child#001	28.62	103.02	9	17.47
child#002	27.51	99.02	9	18.18
child#003	31.21	112.35	8	16.02
child#004	25.23	90.84	12	19.82
child#005	12.21	43.97	10	40.93
child#006	24.72	89.00	8	20.22
child#007	13.81	49.71	9	36.21
child#008	23.26	83.74	10	21.49
child#009	28.75	103.48	7	17.39
child#010	24.21	87.16	12	20.65
adolescent#001	13.61	49.00	18	36.73
adolescent#002	8.06	29.02	19	62.03
adolescent#003	20.62	74.25	15	24.24
adolescent#004	14.18	51.06	17	35.25
adolescent#005	14.70	52.93	16	34.00
adolescent#006	10.08	36.30	14	49.58
adolescent#007	11.46	41.25	16	43.64
adolescent#008	7.89	28.40	14	63.39
adolescent#009	20.77	74.76	19	24.08
adolescent#010	15.07	54.26	17	33.17
adult#001	9.92	35.70	61	50.42
adult#002	8.64	31.10	65	57.87
adult#003	8.86	31.90	27	56.43
adult#004	14.79	53.24	66	33.81
adult#005	7.32	26.35	52	68.32
adult#006	8.14	29.32	26	61.39
adult#007	11.90	42.85	35	42.01
adult#008	11.69	42.08	48	42.78
adult#009	7.44	26.78	68	67.21
adult#010	7.76	27.93	68	64.45

Table 3: Basal-Bolus Parameters

A.2. BB Parameters

The parameters are presented in **Table 3**. The simulator provides age and TDI for each individual. To calculate CR and CF we use equations $CR = 500/TDI$ and $CF = 1800/TDI$, which were provided to us via a clinical consultation. The resulting CR and CF we used differed from those provided in the baseline Simglucose download from Xie (2018), in practice we found our values led to better performance.

A.3. PID and PID-MA parameters

	k_p	k_i	k_d
child#001	-3.49E-05	-1.00E-07	-1.00E-03
child#002	-3.98E-05	-2.87E-08	-3.98E-03
child#003	-6.31E-05	-1.74E-08	-1.00E-03
child#004	-6.31E-05	-1.00E-07	-1.00E-03
child#005	-1.00E-04	-2.87E-08	-6.31E-03
child#006	-3.49E-05	-1.00E-07	-1.00E-03
child#007	-3.98E-05	-6.07E-08	-2.51E-03
child#008	-3.49E-05	-3.68E-08	-1.00E-03
child#009	-3.49E-05	-1.00E-07	-1.00E-03
child#010	-4.54E-06	-3.68E-08	-2.51E-03
adolescent#001	-1.74E-04	-1.00E-07	-1.00E-02
adolescent#002	-1.00E-04	-1.00E-07	-6.31E-03
adolescent#003	-1.00E-04	-1.00E-07	-3.98E-03
adolescent#004	-1.00E-04	-1.00E-07	-4.79E-03
adolescent#005	-6.31E-05	-1.00E-07	-6.31E-03
adolescent#006	-4.54E-10	-1.58E-11	-1.00E-02
adolescent#007	-1.07E-07	-6.07E-08	-6.31E-03
adolescent#008	-4.54E-10	-4.54E-12	-1.00E-02
adolescent#009	-6.31E-05	-1.00E-07	-3.98E-03
adolescent#010	-4.54E-10	-4.54E-12	-1.00E-02
adult#001	-1.58E-04	-1.00E-07	-1.00E-02
adult#002	-3.98E-04	-1.00E-07	-1.00E-02
adult#003	-4.54E-10	-1.00E-07	-1.00E-02
adult#004	-1.00E-04	-1.00E-07	-3.98E-03
adult#005	-3.02E-04	-1.00E-07	-1.00E-02
adult#006	-2.51E-04	-2.51E-07	-1.00E-02
adult#007	-1.22E-04	-3.49E-07	-2.87E-03
adult#008	-1.00E-04	-1.00E-07	-1.00E-02
adult#009	-1.00E-04	-1.00E-07	-1.00E-02
adult#010	-1.00E-04	-1.00E-07	-1.00E-02

Table 4: PID parameters

	k_p	k_i	k_d
child#001	-5.53E-09	-1.00E-07	-3.49E-04
child#002	-1.00E-04	-2.87E-08	-1.00E-03
child#003	-1.00E-05	-2.87E-08	-1.00E-03
child#004	-6.31E-05	-1.00E-07	-1.00E-03
child#005	-1.00E-04	-1.00E-07	-3.31E-03
child#006	-1.00E-05	-3.68E-08	-1.00E-03
child#007	-2.35E-07	-1.00E-07	-1.00E-03
child#008	-4.72E-06	-2.87E-08	-1.00E-03
child#009	-1.00E-05	-1.00E-07	-3.49E-04
child#010	-3.49E-05	-4.72E-08	-1.00E-03
adolescent#001	-1.00E-04	-4.72E-08	-6.31E-03
adolescent#002	-1.00E-05	-1.00E-07	-3.49E-03
adolescent#003	-6.31E-05	-1.00E-07	-2.09E-03
adolescent#004	-6.31E-05	-1.00E-07	-2.51E-03
adolescent#005	-4.79E-05	-1.00E-07	-3.98E-03
adolescent#006	-1.00E-04	-1.00E-07	-2.75E-03
adolescent#007	-1.00E-05	-1.00E-07	-3.02E-03
adolescent#008	-1.58E-09	-1.00E-07	-2.75E-03
adolescent#009	-3.98E-05	-1.00E-07	-1.91E-03
adolescent#010	-1.00E-04	-1.00E-07	-4.37E-03
adult#001	-1.07E-07	-1.00E-07	-4.37E-03
adult#002	-1.58E-04	-1.00E-07	-4.37E-03
adult#003	-7.59E-05	-1.00E-07	-2.51E-03
adult#004	-1.00E-04	-1.00E-07	-1.00E-03
adult#005	-1.07E-07	-1.00E-07	-6.31E-03
adult#006	-1.00E-04	-1.00E-07	-1.00E-02
adult#007	-1.58E-04	-2.51E-07	-3.02E-03
adult#008	-1.58E-05	-1.00E-07	-3.98E-03
adult#009	-4.54E-10	-1.00E-07	-6.31E-03
adult#010	-3.98E-05	-1.00E-07	-4.37E-03

Table 5: PID-MA parameters

A.4. Relevant Patient Subgroups

For tuning our models and selecting hyperparameters, we focused on one challenging but representative individual from each category. In particular, we used child#001, adolescent#004, and adult#001.

For our action space ablation, we examined the subset of 4 individuals who were most prone to catastrophic failures: child#006, child#008, adolescent#002, and adult#009

For the termination penalty experiment, we included child#001, child#003, adolescent#002, adolescent#008, adult#008, and adult#009. We used these patients as they contained a mix of regular and hard to control patients.

For seed progression experiments, we focused on a subset of patients we found reliably challenging for different models to control (in terms of risk and catastrophic failure rate). We included child#001, child#006, child#008, adolescent#002, adolescent#003, adult#001, and adult#009 in this analysis.

A.5. RL-Scratch on Adolescent#002

RL-Scratch has a distinct form of degenerate behavior that occurs only in adolescent#002, drastically lowering performance (see **Figure 2**). We reliably observe that models trained on adolescent#002 do not administer *any* insulin, and thus achieve chronic hyperglycemia. This is because, unlike any other virtual patient, adolescent#002 does not require any insulin to achieve blood glucose levels below 1000 mg/dL (the threshold at which we apply the hyperglycemic termination penalty) over a 10-day period. As a result of the large disparity between average rewards and the termination penalty, the network quickly learns to never administer insulin and the learned exploration rate collapses to 0. This approach can be easily avoided by warm-starting using an environment from another individual, as then the network has already learned to administer generally safe amounts of insulin. Adolescent#002 spends, on average, 97% of their time hyperglycemic under RL-Scratch, and only 39.2% of time hyperglycemic under RL-Trans.