MPC-guided Imitation Learning of Bayesian Neural Network Policies for the Artificial Pancreas

Hongkai Chen[†], Nicola Paoletti[‡], Scott A. Smolka[⋆] and Shan Lin[†]

***Abstract*— Although Model Predictive Control (MPC) is one of the main algorithms that has been proposed for insulin control in the context of artificial pancreas (AP), it typically requires complex online optimization, which is infeasible for resource-constrained medical devices. MPC also usually relies on state estimation, an error-prone process. In this paper, we introduce a novel approach to insulin control for the AP that uses Imitation Learning to synthesize neural-network policies from MPC-computed demonstrations. Such policies are computationally efficient and, by instrumenting MPC at training time with full state information, they can directly map measurements into optimal therapy decisions, thus bypassing state estimation. We apply Bayesian inference via Monte Carlo Dropout to learn policies, which allows us to quantify prediction uncertainty and thereby derive safer therapy decisions. We show that our control policies trained under specific patient models readily generalize (in terms of model parameters and disturbance distributions) to patient cohorts, consistently outperforming traditional MPC with state estimation.**

I. Introduction

The *artificial pancreas* (AP) is a system for the automated delivery of insulin therapy for Type 1 diabetes (T1D), a disease in which patients produce little or no insulin to regulate their blood glucose (BG) levels and maintain adequate glucose uptake in muscle and adipose tissue. The AP consists of an insulin infusion pump and a subcutaneous Continuous Glucose Monitor (CGM) for sensing glucose levels. CGM readings are transmitted to a control algorithm that computes the appropriate insulin dose. Such control should maintain a fine balance. Lack of insulin leads to hyperglycemia (i.e., high BG), which if untreated can cause complications such as stroke, kidney failure, and blindness. Excessive insulin can lead to hypoglycemia (low BG), a critical state that can result in unconsciousness and death.

Driven by advances in the mathematical modeling of T1D physiology [1], [2], *Model Predictive Control* (MPC) has become the preferable AP algorithm due to demonstrated performance improvements over other approaches, both in in-silico and clinical trials [3], [4]. MPC works by determining the insulin therapy that optimizes the future BG profile, predicted via physiological models. It has, however, two important limitations.

[†]Hongkai Chen and Shan Lin are with the Electrical and Computer Engineering Department, Stony Brook University, Stony Brook, NY, 11794, USA {hongkai.chen, shan.x.lin}@stonybrook.edu

[‡]Nicola Paoletti is with Department of Computer Science, Royal Holloway, University of London, UK Nicola.Paoletti@rhul.ac.uk

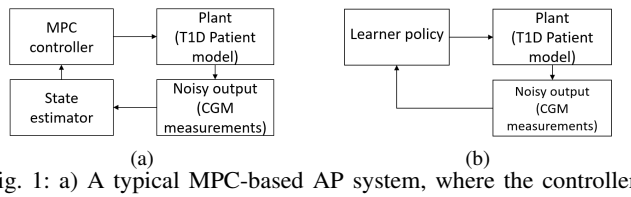
[⋆]Scott Smolka is with the Department of Computer Science, Stony Brook University, Stony Brook, NY, 11794, USA sas@cs.stonybrook.edu

First, MPC requires complex (often nonlinear and non-convex) online optimization, which is infeasible when the algorithm is deployed in resource-constrained medical devices. This is why commercial AP systems use simplistic linear models with MPC (e.g., a linearized version of the nonlinear model [2] in the Typezero® insulin delivery algorithm [5]) or favor simpler control algorithms (e.g., PID control in the Medtronic™ Minimed 670G [6], and fuzzy control in the Glucositter by DreaMed Diabetes [7]). Second, and more crucial, MPC requires *state estimation* (SE) to recover the most recent patient state from CGM measurements [8], [9]. Besides its computational cost, SE is error-prone, as it relies strictly on CGM readings, which are an approximate, delayed, and noisy proxy of the target control variable, the BG. Incorrect state estimates might compromise the safety of the insulin therapy.

Our Contributions. We present a novel method to derive *end-to-end insulin control policies*, i.e., policies that subsume state estimation and control, directly mapping CGM measurements into patient-optimal insulin dosages. To capture the complex logic of MPC and SE, we consider policies represented as deep recurrent neural networks. Such an approach addresses the main issues surrounding the use of MPC, as it bypasses explicit SE and avoids the cost of MPC's online optimizations. Our approach is centered around the use of *Imitation Learning* (IL) [10], where the control policy is trained on examples provided by MPC experts. We employ IL as it alleviates the *covariate shift* [11] arising in simpler approaches such as behavioral cloning, in which the learner is trained on trajectories explored by the expert. This distribution of trajectories necessarily deviates from that explored by the learner at test time; the resulting shift can lead to unpredictable behavior by the learner, and hence jeopardize patient safety.

Similar to the PLATO framework [12], at training time, we instrument the MPC teacher to access the full state of the patient model. As such, the learner policy, using only CGM measurements, will learn to mimic MPC-based decisions based on the true model state, thus avoiding the pitfalls of SE. In this way, the learned policy implicitly learns SE, and subsumes both SE and control.

We learn stochastic policies via approximate Bayesian inference, using *Monte Carlo dropout* [13]. The resulting Bayesian neural network policies allow us to quantify prediction uncertainty [14], information we actively use to make robust therapy decisions. In contrast, PLATO's policy actions are normally distributed with non-learnable variance. Uncertainty quantification is crucial in medical decision making, especially in the AP setting where variations in



(a) (b)

Fig. 1: a) A typical MPC-based AP system, where the controller requires a state estimate. b) Our learned end-to-end insulin policy instead only requires (noisy) observations.

the patient's physiological status are the norm. Situations of this nature can be challenging for a deterministic policy, with consequences for the patient's health.

In summary, the main contribution of this paper is an IL-based method for deriving Bayesian neural network policies for AP control. Our method overcomes two main shortcomings of established MPC-based approaches, namely, SE errors and computational cost. We show that: 1) our IL-based approach outperforms behavioral cloning, while requiring less supervision data; 2) the learned stochastic policies outperform MPC with SE and deterministic policies; and 3) our stochastic policies generalize to never-before-seen disturbance distributions and patient parameters arising in virtual patient cohorts; in the same setting, MPC with SE exhibits consistent performance degradation. Overall, our best stochastic policy keeps BG in euglycemia 8.4%–11.75% longer than MPC with SE and 2.94%–9.07% longer than the deterministic policy.

II. Background on MPC for the AP

We consider an *in silico* AP system, where the T1D patient is represented by a glucose-insulin metabolism model, including absorption, excretion and transport dynamics between different body compartments. In particular, we choose the well-established Hovorka's model [1], a nonlinear ODE model with 14 state variables, which is one of the most sophisticated and realistic models built from real patient data. Figure 1 (a) shows a diagram of the MPC-based AP system, whose state-space description is given in equations (1–3) below. The notation $a_{i,\dots,i+j}$ stands for the indexed sequence $a_i, a_{i+1}, \dots, a_{i+j}$.

$$\mathbf{x}_{t+1} = \mathbf{F}_{\lambda_t}(\mathbf{x}_t, u_t, d_t), \ \lambda_t \sim \Lambda(\lambda \mid t), d_t \sim \mathcal{D}(d \mid t) \quad (1)$$

$$y_t = h(\mathbf{x}_t) + \epsilon_t \quad (2)$$

$$u_t = \pi^*(\hat{\mathbf{x}}_t, d_{t,\dots,t+N_p}) \quad (3)$$

$$\hat{\mathbf{x}}_t = g(y_{t-N_b,\dots,t}, u_{t-N_b,\dots,t}, d_{t-N_b,\dots,t-1}) \quad (4)$$

Equation (1) describes the T1D patient model, where $\mathbf{x}_t$ is the patient state at time t, u_t is the insulin input, λ_t are the patient parameters with distribution $\Lambda(\lambda \mid t)$, and d_t is the meal disturbance, represented by ingested carbohydrate (CHO), with distribution $\mathcal{D}(d \mid t)$. Note that both parameters and disturbances are random and time-dependent. Meal disturbances depend on the patients' eating behaviours followed by daily and weekly patterns. Similarly, parameters are typically subject to intra-patient variations such as daily oscillations. The parameter distribution $\Lambda(\lambda \mid t)$ can be used to describe a patient population as we do in our experiments. The output y_t observed at time t, i.e., the CGM measurement, is subject to Gaussian sensor noise $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon)$ [15]. We denote the MPC-based control policy with π^*, which given state estimate $\hat{\mathbf{x}}_t$ and future meal disturbances $d_{t,\dots,t+N_p}$ as inputs, computes the optimal insulin therapy u_t at each time t by solving the following online optimization problem [9],

$$\min_{u_{t,\dots,t+N_p-1}} J(\hat{\mathbf{x}}_t, d_{t,\dots,t+N_p}, u_{t,\dots,t+N_p-1}) = \sum_{k=1}^{N_p} d_{BG}(\tilde{\mathbf{x}}_{t+k}) + \beta \cdot \sum_{k=0}^{N_c-1} (\Delta u_{t+k})^2 \quad (5)$$

subject to

$$u_{t+k} \in D_u, \quad k = 0, \dots, N_c - 1 \quad (6)$$

$$u_{t+k} = \bar{u}, \quad k = N_c, \dots, N_p - 1 \quad (7)$$

$$\tilde{\mathbf{x}}_t = \hat{\mathbf{x}}_t \quad (8)$$

$$\tilde{\mathbf{x}}_{t+k+1} = \mathbf{F}_{\lambda_{t+k}}(\tilde{\mathbf{x}}_{t+k}, u_{t+k}, d_{t+k}), \ k = 0, \dots, N_p - 1 \quad (9)$$

where N_p is the MPC prediction horizon, and $N_c \leq N_p$ is the control horizon; D_u is the set of admissible insulin values; (7) states that u is fixed to the basal insulin rate $\bar{u}$ outside the control horizon. (8) and (9) describe how the predicted state $\tilde{\mathbf{x}}$ evolves from estimated state $\hat{\mathbf{x}}_t$ at time t following the plant model (1). The objective function contains two terms, weighted by hyper-parameter $\beta > 0$: in the first term, $d_{BG}(\tilde{x}_{t+k}) = (BG_{t+k} - BG_{target})^2$ is designed to penalize deviations between predicted and target BG; in the second, $(\Delta u_{t+k})^2 = (u_{t+k} - u_{t+k-1})^2$ avoids abrupt changes of the insulin infusion rate. To avoid introducing prediction inaccuracy due to uncertain disturbance values (which would require using robust MPC designs or disturbance estimation [16], [9], [17]), here we assume that the true future disturbance values are known.

Finally, (4) describes state estimation g, which, in our study, uses *moving horizon estimation* (MHE) [18]. MHE is related to MPC in that they both use model-based predictions; however, MHE estimates the most likely state given a sequence of N_b past measurements, control inputs, and disturbances. We call N_b the estimation horizon. The estimated state minimizes the discrepancy between observed CGM outputs and the corresponding model predictions. As such, *the quality of the estimate directly depends on the accuracy of the predictive model and the quality of the measurements*. Details of the MHE optimization problem for the AP can be found in [17]. The combination of MPC and MHE is one of the most often used designs in AP control research. Alternative SE methods include Kalman filters, which are not appropriate for nonlinear systems [19], and particle filters [18], but these are subject to the same kinds of estimation errors as MHE.

We stress that, even though the main requirement is to maximize the time within the safe glucose range, brief excursions from the safe range are not an issue and are actually often inevitable (e.g., after a meal). Hence, we do not enforce safe BG levels via hard state constraints (which would make the problem infeasible) but rather with an objective function that tracks a target safe BG level.

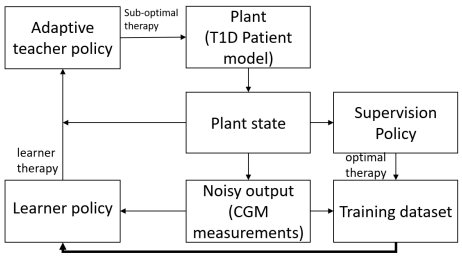


Fig. 2: Overview of the IL scheme for the AP. Training-time trajectories are generated by applying the adaptive teacher policy, which is a trade-off between the optimal supervision policy and the learner's policy. For training the learner, visited states are labelled with the optimal action of the supervision policy.

III. Overview of the Method

The main goal of our work is to design an end-to-end insulin control policy for the AP, i.e., policies that directly map noisy system outputs (CGM measurements) into an optimal insulin therapy, without requiring knowledge of the system state. The control loop of such a policy is illustrated in Figure 1 (b). To this purpose, we take an imitation learning approach where the *learner policy* is trained from examples provided by an MPC-based expert, called the *supervision policy*. At training time, the supervision policy is instrumented with full state information, so that its demonstrations are not affected by SE errors and thus, can be considered optimal.

A common approach, called *behavioral cloning* (BC), is to train the learner via supervised learning (SL) using trajectories explored by the expert. However, BC is not suitable for our problem because SL assumes i.i.d. training and test data, while our case is subject to covariate shift: the training state distribution explored via the expert is different from the test state distribution induced by the learner policy. If the learner cannot imitate perfectly the expert, the learner's actions could bring the system into out-of-distribution states, where the behaviour of the learner becomes unpredictable, and cause safety implications for our AP system. In IL, one should provide demonstrations on trajectories that the learner would explore, but without knowing the learner in advance. To do so, a common solution is to reduce IL into a sequence of SL problems, where at each iteration, the learner is trained on the distribution induced by the previous learners or by a "mixture" of learner and supervision policy [20], [10].

Our IL method builds on the PLATO algorithm for adaptive trajectory optimization [12]. See Section VII for a summary of our extensions to PLATO. PLATO also reduces IL into a sequence of SL problems, where at each iteration, the teacher's actions gradually adapt to those of the current learner policy. This *adaptive teacher* is an MPC-based policy whose objective function is extended with a term that penalizes its mismatch to the learner's behavior. As such, it is non-optimal, and is only used to generate trajectories that approach the distribution induced by the learner, thereby alleviating the covariate shift problem. The training data is obtained by labelling the adaptive teacher trajectories with the original (optimal) MPC policy. We call the latter *supervision policy*. The training process is summarized in Figure 2. In our approach the learner policy is stochastic, represented as a Bayesian neural network, as explained in Section V.

IV. Imitation Learning Algorithm

The supervision policy π^* is the MPC policy for the AP system of (5–9). We denote its control action at time t as u_t^*, which is obtained by solving the MPC problem given the true system state $\mathbf{x}_t$ (instead of the estimated state) and future meal disturbances: $u_t^* = \pi^*(\mathbf{x}_t, d_{t\ldots t+N_p})$.

The learner policy π_θ^L is represented by an LSTM network with parameters θ. In particular, θ is a vector of random parameters derived via approximate Bayesian inference, as explained in the next section. The choice of a recurrent architecture is natural for our application, because our policies have to subsume both control and SE and, as discussed in Section II, SE for nonlinear models can be seen as a sequence prediction problem, see (4). At time t, the learner policy derives u_t^L, the control action at time t, as $(\mathbf{s}_t, u_t^L) = f_\theta(\mathbf{s}_{t-1}, y_t, d_{t+N_p}, u_{t-1})$, given in input the observation y_t, past control action u_{t-1}, and future disturbance d_{t+N_p}. f_θ is the LSTM function with (random) parameters θ and $\mathbf{s}_t$ is the hidden state of the LSTM network. Since the resulting policy is stochastic, we will denote the learner by the conditional distribution $\pi_\theta^L(u_t^L \mid \mathbf{s}_{t-1}, y_t, d_{t+N_p}, u_{t-1})$.

The adaptive teacher policy π^T extends the supervision policy π^* with a term to penalize the mismatch between the learner policy and itself. Its output u_t^T is the first control action in the solution of the following MPC problem.

$$\min_{u^T_{t,\ldots,t+N_p-1}} \quad J(\mathbf{x}_t, d_{t,\ldots,t+N_p}, u^T_{t,\ldots,t+N_p-1}) + \rho \cdot J_M \tag{10}$$

subject to (6–9). The first term of (10) is the cost function of the supervision policy (5). The second term J_M quantifies the discrepancy between the actions of the supervision and learner policies. In particular, we define J_M as the p-th Wasserstein distance W_p between (an empirical approximation of) the output distribution of the stochastic learner policy $\pi_\theta^L(u_t^L \mid \mathbf{s}_{t-1}, u_{t-1}, y_t, d_{t+N_p})$ and the optimal control action u_t^T, or more precisely $\delta_{u_t^T}$, the Dirac distribution centered at u_t^T. The factor ρ determines the relative importance of matching the behavior of the learner policy π_θ^L against minimizing the cost J. In our experiments, we set $\rho = 1 - 0.8^{i-1}$ where i is the IL iteration of Algorithm 1. In this way, the relative importance of matching π_θ^L increases as the learner improves and as i increases. By gradually matching the behaviour of the learner, the control actions taken by π^T will lead to exploring a state space similar to the one induced by the learner policy.

Algorithm 1 outlines our IL scheme, which consists of a sequence of N SL iterations. The learner policy is initialized at random. At each iteration, we start from a random initial state of the plant and generate a trajectory of the system of length T_e. To do so, we first sample a sequence of random meal disturbances and patient parameters (lines 4-7). Then, for each time point t of the trajectory, the adaptive teacher π^T computes an insulin action u_t^T by solving (10), that is, by optimizing the MPC objective while matching the current learner policy $\pi_{\theta_i}^L$ (line 11). The MPC supervision policy π^* is used to compute the optimal control action u_t^* by solving (5) (line 12). The optimal action is used to label the corresponding

Algorithm 1: Imitation learning for AP control policies

1: Initialize training dataset $S \leftarrow \emptyset$.
2: Randomly initialize learner policy $\pi^L_{\theta_1}$.
3: **for** $i = 1$ **to** N **do**
4: **for** $t = 1$ **to** $T_e + N_p$ **do**
5: Sample random patient parameters $\lambda_t \sim \Lambda(\lambda \mid t)$.
6: Sample random carb disturbance $d_t \sim \mathcal{D}(d \mid t)$.
7: **end for**
8: Initialize the patient state $\mathbf{x}_1$.
9: **for** $t = 1$ **to** T_e **do**
10: Collect CGM measurements y_t as per (2).
11: Compute sub-optimal therapy with adaptive teacher: $u^T_t \sim \pi^T(u^T_t \mid \mathbf{x}_t, d_{t,\cdots,t+N_p}, \pi^L_{\theta_i})$.
12: Compute optimal therapy with supervision policy: $u^*_t = \pi^*(\mathbf{x}_t, d_{t,\cdots,t+N_p})$.
13: Append $\big((y_t, u^T_{t-1}, d_{t+N_p}), u^*_t\big)$ to S.
14: State evolves as $\mathbf{x}_{t+1} = \mathbf{F}_\lambda\left(\mathbf{x}_t, u^T_t, d_t\right)$.
15: **end for**
16: Train $\pi^L_{\theta_{i+1}}$ on S.
17: **end for**

training example $\big((y_t, u^T_{t-1}, d_{t+N_p}), u^*_t\big)$, which is added to the training set S (line 13), while the sub-optimal action by the adaptive teacher is used to evolve the system state (line 14). At the end of each iteration, $\pi^L_{\theta_i}$ is trained using the examples in S. As the teacher gradually matches the learner, the training-time distribution of the system state gradually approximates that at test time.

The time complexity of Algorithm 1 is dominated by the two MPC instances (i.e., π^* and π^T), which are solved $N \cdot T_e$ times, and the training of the learner π^L, repeated N times.

Behavioral cloning policy. We obtain a corresponding BC policy by using the MPC expert for both exploration and supervision, i.e., by replacing π^T with π^*.

V. Bayesian inference of control policy

We take a Bayesian approach to learn our control policy, which results in a stochastic policy represented by a neural network with random parameters θ. This provides us with a distribution of policy actions, the *predictive distribution*, from which we can derive uncertainty measures to inform the final therapy decision. Such uncertainty should capture both data uncertainty, e.g., noisy measurements, and epistemic uncertainty, i.e., the lack of confidence of the model about a given input [14].

Given training data S, performing Bayesian inference corresponds to computing the posterior $p(\theta \mid S)$ from some prior $p(\theta)$ by applying Bayes rule. The distribution of policies is induced by the random parameters $\theta \sim p(\theta \mid S)$. The predictive distribution $p(u^L_t \mid \mathbf{s}_{t-1}, y_t, u_{t-1}, d_{t+N_p}, S)$ is derived from the posterior and the policy by marginalizing θ:

$$p(u^L_t|x, S) = \int \pi^L_\theta(u^L_t|x) \cdot p(\theta|S)\ \mathrm{d}\theta. \tag{11}$$

For the non-linearity of the neural network function, precise inference is, however, infeasible and thus one needs to resort to approximate methods [21], one of which is *Monte Carlo Dropout* (MCD) [13]. Dropout is a well-established regularization technique based on dropping some neurons at random during training with some probability p, by multiplying the weights with a dropout mask, i.e., a vector of Bernoulli variables with parameter p. Then, at test time, standard dropout derives a deterministic network by scaling back the weights by a factor of $1/(1-p)$. On the other hand, in MCD the random dropout mask is applied at test time too, resulting in a distribution of network parameters. Studies show that applying dropout to each weight layer is equivalent to performing approximate Bayesian inference of the neural network [13]. This property efficiently reduces the problem of inferring $p(\theta \mid S)$ to drawing Bernoulli samples.

Decision rule. The output of our policy is the predictive distribution of insulin actions (11). Hence, we need to define a decision rule that produces a value u^t out of this distribution and accounts for the predictive uncertainty of the policy. Consider an empirical approximation of (11) given by an iid sample $u^L_{t,1}, \cdots, u^L_{t,n}$ of size n. W.l.o.g., assume $u^L_{t,1}, \cdots, u^L_{t,n}$ be ordered. Let y_{t-1} be the last measured glucose value. Our rule selects a particular order statistic $u^L_{t,M}$, i.e., one of the sampled values, depending on the relative distance of y_{t-1} w.r.t. the safe BG upper bound BG_{ub} and lower bound BG_{lb}. We call this *adaptive rule* because the selected order statistic is adapted on y_{t-1}. Formally,

$$u^t = u^L_{t,M}, M = n \cdot \lceil (y_{t-1} - BG_{lb})/(BG_{ub} - BG_{lb}) \rceil. \tag{12}$$

In this way, if the patient is approaching hypoglycemia (y_{t-1} close to BG_{lb}), we select a conservative insulin value, and we select instead an aggressive therapy if y_{t-1} is close to hyperglycemia (BG_{ub}). Importantly, as the policy uncertainty increases (and so does the spread of the sample), u^t gets more conservative when y_{t-1} is in the lower half of the safe BG range, i.e., we take safer decisions when the policy is less trustworthy because protecting against hypoglycemia is the primary concern. For the same principle, when y_{t-1} is in the upper half of the range, higher uncertainty yields a more aggressive therapy, but this poses no safety threats because y_{t-1} is well away from the hypoglycemia threshold BG_{lb}.

In our evaluation, we compare our adaptive rule with the commonly used rule that sets u^t to the sample mean of (11).

VI. Experimental Evaluation

We conducted *in-silico* computational experiments to validate the following claims:

1) Our IL-based approach converges faster to an optimal policy than BC.
2) The stochastic IL-based policies outperform both MPC with SE and deterministic policies.
3) The stochastic IL-based policies generalize well to unseen patient physiological parameters and meal disturbances.
4) For a stochastic policy, the adaptive decision rule (12) outperforms the mean prediction rule.
5) The predictive uncertainty of the policy increases with out-of-distribution test inputs.

Runtime performance. On our workstation (an Intel i7-8750H CPU with 16GB DDR4 SDRAM) the stochastic policy

TABLE I: Performance of MPC with state information (MPC+SI), MPC with state estimation (MPC+SE), deterministic learner policy (DLP), stochastic learner policy with mean value output (SLP-M), and stochastic learner policy with adaptive output (SLP-A). For each column, in bold are the significantly best policies, i.e., with $p < 0.005$ in all pairwise sign tests (one-sided) [22]. Performance of SLP-A with unseen disturbances is shown in the last row (*).

	Fixed Patient Configuration			Varying Patient Configuration			Patient Cohort Configuration		
	t_{hypo} (%)	t_{eu} (%)	t_{hyper} (%)	t_{hypo} (%)	t_{eu} (%)	t_{hyper} (%)	t_{hypo} (%)	t_{eu} (%)	t_{hyper} (%)
MPC+SI	0.00 ± 0.00	99.84± 0.60	0.16± 0.60	0.00±0.00	99.80±0.83	0.20±0.83	0.00±0.00	99.15±1.81	0.85±1.81
MPC+SE	0.92±2.76	80.40±5.33	18.68±4.76	1.44±3.18	79.88±5.46	18.69±4.95	0.85±3.57	77.17±15.39	21.98±14.30
DLP	0.21±0.99	85.40±4.15	14.38±3.90	0.53±1.87	82.66±6.42	16.82±6.25	0.45±1.91	82.63±6.28	16.93±6.15
SLP-M	0.23±1.10	86.33±4.17	13.44±4.09	0.33±1.09	86.34±4.20	13.33±4.06	0.32±1.19	85.45±4.34	14.23±4.00
SLP-A	**0.13±0.64**	**91.73±3.45**	**8.14±3.39**	**0.05±0.27**	**91.73±3.18**	**8.23±3.15**	**0.04±0.41**	**85.57±4.12**	**14.39±4.08**
SLP-A*	0.35±1.08	91.41±3.67	8.24±3.43	0.67±1.83	90.77±4.44	8.56±3.73	0.00±0.00	85.92±3.96	14.08±3.96

executes in ∼20 milliseconds, which is well within the CGM measurement period of 5 minutes, and consistently more efficient than MPC-based optimization (on average, 150+ times faster in our experiments)[1]. Thanks to platforms such as TensorFlow Lite, we believe that similar runtimes can be obtained after deploying the policy on embedded hardware.

LSTM architecture. We represent the learner policy as a multi-layer regressor trained to minimize RMSE loss. It contains three LSTM layers with 200 hidden units, tanh state activation and sigmoid gate activation, followed by a fully-connected layer, ReLu nonlinearity and a regression layer. We place an MCD layer (p=0.2) before each weight layer, and use Adam [24] in the training. At both training and test times, we sample 50 realizations of the predictive distribution. This architecture exhibited the best performance during evaluation. We also experimented with feed-forward architectures but they performed poorly, confirming the need for recurrent architectures to adequately represent SE and control.

Performance measures. To evaluate and compare policies, we consider three performance measures that are typically used for assessing clinical outcomes in the AP domain [25]: $t_{hypo}, t_{hyper}, t_{eu}$, the average percentage of time a virtual patient's BG is in hypoglycemia (i.e., BG $\leq$ 70 mg/dL), hyperglycemia (i.e., BG $\geq$ 180 mg/dL), and euglycemia (i.e., safe range of 70 mg/dL $\leq$ BG $\leq$ 180 mg/dL), respectively. The goal of diabetes treatment and BG control is to maximize t_{eu} while minimizing t_{hypo} (hypoglycemia leads to more serious acute consequences than hyperglycemia).

Experimental settings. During training, we performed time-invariant parameterization of the T1D virtual patient model, and used the disturbance distribution of Table III. The length of each training trajectory is set to $T_e = 1,440$ minutes (1 day). Algorithm 1 runs $N = 34$ iterations, generating a training set of size $|S| = 48,960$ minutes. We trained with 500 maximum epochs, an initial learning rate of 0.005 which decreases by a factor of 0.2 every 125 epochs, and mini-batch size of 1440. We stopped the training when the validation loss, calculated every 30 epochs, is non-decreasing five consecutive times. At test time, we consider the following three configurations of the model parameters:

- *Fixed:* virtual patient with constant parameters;
- *Varying:* virtual patient with time-varying parameters;
- *Cohort:* based on inter-patient and intra-patient variations. Here, the parameter distribution models a cohort of virtual patients with time-varying parameters.

The above distributions for modeling intra- and inter-patient variability were taken from [26], where the authors derived them from clinical data. We randomly sampled the model parameters from these distributions, some of which oscillate during the entire experiments. We stress that intra- and inter-patient variations can have a significant impact on the BG response and thus, on the effectiveness of the insulin policy.

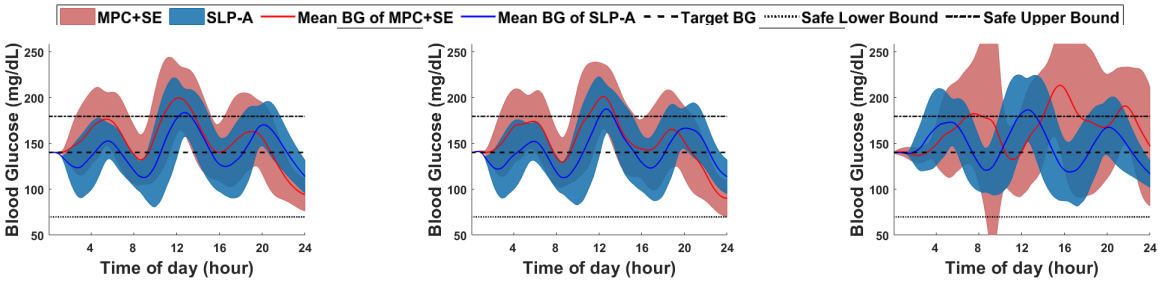
For each configuration and policy, we conduct 90 one-day simulations. For each simulation, we draw fresh realizations of random disturbances and patient parameters. To ensure a fair comparison, these are kept the same across all evaluated policies (and all policies are given full meal disturbances information). We evaluate and compare the following policies:

- **MPC+SI:** the supervision policy π^* with full information on model state and parameters. Full observability is an unrealistic assumption, yet useful to establish an ideal performance level. We set control and prediction horizons to $N_c = 100$ minutes and $N_p = 150$ minutes.
- **MPC+SE:** the supervision policy π^* with *state estimation*, as described in (5–9) with estimation horizon $N_b = 200$ minutes. The parameters of the T1D prediction model are set to the average of their distributions.
- **DLP:** the *deterministic learner policy* equivalent to the stochastic IL policy except it uses ordinary dropouts.
- **SLP-M:** the stochastic learner policy with a different decision rule which selects the mean of the empirical predictive distribution.
- **SLP-A:** the stochastic learner policy with the adaptive rule of (12). Such rule actively uses uncertainty information by choosing ordered statistics of the insulin distribution to commensurate the sensed glucose. We show that SLP-A outperforms all the others (but MPC+SI).

The results are summarized in Table I. We use the non-parametric paired sign test [22] to perform pairwise comparisons and establish whether, for each performance measure, the best performing policy is (statistically) significantly better than all the others (but MPC+SI).

We also test SLP-A with a meal disturbance distribution different from the training-time one. This distribution reflects

[1]MPC- and SE-based optimization problems are solved using MATLAB's implementation of the interior-point algorithm of [23].



(a) Fixed patient (b) Varying patient (c) Patient cohort

Fig. 3: Mean BG $\pm$ standard deviation in experiments regulated by MPC+SE and SLP-A under three virtual patient configurations.

TABLE II: Average time in euglycemia for the patient cohort configuration, at different iterations of Algorithm 1.

Epoch Number	1	5	10	15	20	25	30
BC (%)	35.45	14.99	15.15	18.23	17.35	17.43	15.04
IL (%)	24.69	70.30	24.63	71.43	79.60	76.45	69.95

TABLE III: Attributes of meal disturbance distribution during training. CHO amounts and starting times are sampled uniformly from the reported intervals.

	breakfast	snack 1	lunch	snack 2	dinner	snack 3
Probability (%)	100	50	100	50	100	50
CHO (gram)	40-60	5-25	70-110	5-25	55-75	5-15
Time of day (hour)	1:00-5:00	5:00-8:00	8:00-12:00	12:00-15:00	15:00-19:00	19:00-21:00

a late eating habit and snacks of higher probabilities and CHO, leading to an overall higher carb intake Because we aim to show that our approach outperforms MPC with state estimation, we omit other types of controllers (e.g., PID controllers) from our comparison.

1. IL converges faster than BC on our testbed. In Table II, we compare, at different iterations of Algorithm 1, the performance of the SLP-A policy against the corresponding stochastic policy trained using BC (described at the end of Section IV), that is, without the adaptive teacher policy that guides the exploration of training trajectories but only using the supervision policy. From Table II, we found that SLP-A attains superior performance, obtaining an average of 79.60% of time in euglycemia only after 20 iterations, while the BC counterparts only 17.35%. This suggests that our IL approach manages to efficiently explore more useful trajectories, resulting in a policy that guarantees safety for larger portions of the state space earlier than BC. Our approach consistently outperforms BC in all other performance metrics too.

2. Stochastic IL-based policies outperform MPC with SE and deterministic policies. Results in Table I evidence that SLP-A outperforms the MPC policy with state estimation in essentially all performance measures and configurations, and in a statistically significant manner. In particular, we observe that, on average, the SLP-A policy stays in euglycemia for 8.4%–11.75% longer than MPC+SE. This is visible also from the BG profiles of Figure 3. It also performs better than DLP, with time in euglycemia 2.94%–9.07% longer, which shows superiority of Bayesian inference and uncertainty quantification. These results suggests that MPC with SE introduces estimation errors that have a detrimental effect on BG control, as also confirmed by the fact that the same control

TABLE IV: Attributes of a different meal disturbance distribution during testing to that during training.

	breakfast	snack 1	lunch	snack 2	dinner	snack 3
Probability (%)	100	80	100	80	100	80
CHO (gram)	40-60	15-30	70-110	15-30	55-75	15-30
Time of day (hour)	3:00-7:00	7:00-10:00	10:00-14:00	14:00-17:00	17:00-21:00	21:00-23:00

algorithm but with full state information (i.e., MPC+SI) is far superior. In realistic settings where the true patient state is not accessible, our analysis shows that an end-to-end policy is to be preferred to explicit SE.

3. Stochastic IL-based policies generalize to unseen patient parameters and disturbances. From Table I and Figure 3, we observe that SLP-A is robust to never-before-seen patient parameters, with time in euglycemia constantly well above 85% despite inter- and intra-patient variations. The superiority of SLP-A under these configurations evidences that both imitation learning and incorporating prediction uncertainty make huge differences when policy is deployed in environments that deviate from the training ones. Furthermore, from Table I, we show that SLP-A outperforms MPC+SE under all configurations, and the difference is both statistically and practically significant (with an approximate 8.4% average improvement of the time in euglycemia). We further evaluate the robustness of SLP-A under an unseen meal disturbance distribution characterized by a higher and late carbs intake, shown in Table IV. Results for this experiment are reported in the last row in Table I (SLP-A* row) and evidence that SLP-A generalizes well also in this case (there is no significant performance degradation w.r.t. the SLP-A row).

4. Adaptive rule outperforms mean-value rule. The SLP-A policy obtains a time in euglycemia approximately 5.4% longer than SLP-M in the fixed patient and varying patient configurations, see Table I. There is an improvement also in the patient cohort experiment, albeit less significant. With the adaptive rule, the policy adopts a more conservative or aggressive therapy depending on the measured glucose and the predictive uncertainty, which can lead to more stable BG trajectories and explain the observed difference.

5. Policy uncertainty increases at out-of-distribution test inputs. We observe a statistically significant increase in output variability when SLP-A is applied on out-of-distribution data resulting from the patient cohort configuration. As a measure of variability, we consider the coefficient of variation (CV), i.e., the mean-normalized standard deviation, of SLP-A's output distribution. Figure 4 shows the cumulative distributions of the CV values under three patient configurations. We

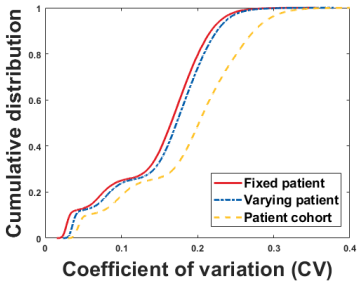


Fig. 4: CDFs of Coefficients of variations (CV) of the SLP-A outputs under three different patient configuration.

remark that, in this case, a faster growing CDF implies a higher probability of smaller CV values (and thus, smaller variability). To compare the three CV distributions, we applied pair-wise two-sample K-S tests at level $\alpha = 0.05$, resulting in a statistically significant difference between the patient cohort CV distribution and the other two. The higher predictive uncertainty in the patient cohort configuration evidences that Bayesian inference via MCD adequately captures epistemic uncertainty. No significant difference in the predictive uncertainty was found between the fixed- and the varying-patient configurations.

VII. Related Work

Traditional methods for insulin control in the AP mostly rely on MPC [1], [27], [28], [29], [9], PID control [30], [31], and fuzzy rules based on the reasoning of diabetes caregivers [7], [32]. Reinforcement learning approaches have been proposed as well, including policy iteration [33], actor-critic methods [34], and deep Q-networks for dual-hormone therapy [35]. Neural networks have been explored in the AP space not just to represent insulin policies [36], [35], but also to predict BG concentration based on inputs such as insulin dosage, nutritional intake, and daily activities [37], [38], [39], [40]. Our work is different from the above papers as it: 1) uses imitation learning to learn the insulin policy, mitigating potential (and dangerous) test-time distribution drifts; 2) incorporates in the policy uncertainty information obtained via Bayesian inference; and 3) produces end-to-end policies that do not require learning a separate BG prediction model. For commercial AP systems using linear patient models, a recent study shows an average time in range of 79.2% after 7 weeks of AP usage [5]. Our in-silico experiments result in an average time in range of 85% with unseen patients and unseen disturbances, and we expect our approach to exhibit similar performance on real patients.

A variety of IL methods have been proposed in the literature, including [41], [20], [10], [42]. Some of these, like our approach, are tailored to work with MPC teachers [43], [12], [44], [45]. Similarly, several recent papers [46], [47], [48], [49] have proposed neural network-based approximations for MPC. Our method is also akin to [50], [51] where Bayesian extensions of IL are presented to quantify the learner's predictive uncertainty and better guide the queries to the teacher policy. Other Bayesian approaches in policy learning include [52], [53]. Our work builds on the PLATO IL algorithm [12] and extends it in three main direction: 1) we consider recurrent architectures, which are more suitable than feedforward ones (used in PLATO) to represent nonlinear state estimation and control; 2) PLATO also derives stochastic policies but, unlike our work, no uncertainty-aware decision-making strategies are considered; 3) PLATO policies do not support systems with external disturbances beyond noise. In our policies instead, random meal disturbances are central.

VIII. Conclusion

We introduced a method based on MPC-guided Imitation Learning and Bayesian inference to derive stochastic policies for insulin control in an artificial pancreas. Our policies are end-to-end in that they directly map CGM values into insulin control actions. By using Bayesian neural networks, we can crucially quantify prediction uncertainty, information that we incorporate in insulin therapy decision-making. We empirically demonstrated that our stochastic insulin policies outperform traditional MPC with explicit state estimation; they are also more robust than their deterministic counterparts, as they generalize well to unseen T1D patient parameters and meal-disturbance distributions.

Acknowledgment

Research supported in part by the National Science Foundation under Grant CNS-1553273 (CAREER), DCL-2040599, CCF-1918225, CPS-1446832.

References

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

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

[3] J. E. Pinsker, J. B. Lee, E. Dassau, D. E. Seborg, P. K. Bradley, R. Gondhalekar, W. C. Bevier, L. Huyett, H. C. Zisser, and F. J. Doyle, "Randomized crossover comparison of personalized MPC and PID control algorithms for the artificial pancreas," *Diabetes Care*, p. dc152344, 2016.

[4] A. Weisman, J.-W. Bai, M. Cardinez, C. K. Kramer, and B. A. Perkins, "Effect of artificial pancreas systems on glycaemic control in patients with type 1 diabetes: a systematic review and meta-analysis of outpatient randomised controlled trials," *The lancet Diabetes & endocrinology*, vol. 5, no. 7, pp. 501–512, 2017.

[5] J. E. Pinsker, L. Müller, A. Constantin, S. Leas, M. Manning, M. McElwee Malloy, H. Singh, and S. Habif, "Real-world patient-reported outcomes and glycemic results with initiation of control-iq technology," *Diabetes technology & therapeutics*, vol. 23, no. 2, pp. 120–127, 2021.

[6] G. M. Steil, "Algorithms for a closed-loop artificial pancreas: the case for proportional-integral-derivative control," *Journal of diabetes science and technology*, vol. 7, no. 6, pp. 1621–1631, 2013.

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

[8] R. Gondhalekar, E. Dassau, and F. J. Doyle, "Moving-horizon-like state estimation via continuous glucose monitor feedback in MPC of an artificial pancreas for type 1 diabetes," in *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014.

[9] N. Paoletti, K. S. Liu, H. Chen, S. A. Smolka, and S. Lin, "Data-driven robust control for a closed-loop artificial pancreas," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 6, pp. 1981–1993, 2019.

[10] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *14th international conference on artificial intelligence and statistics*, 2011.

[11] A. Storkey, "When training and test sets are different: characterizing learning transfer," *Dataset shift in machine learning*, pp. 3–28, 2009.

[12] G. Kahn, T. Zhang, S. Levine, and P. Abbeel, "PLATO: Policy learning using adaptive trajectory optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[13] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[14] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, PhD thesis, University of Cambridge, 2016.

[15] P. Soru, G. De Nicolao, C. Toffanin, C. Dalla Man, C. Cobelli, L. Magni, A. H. Consortium *et al.*, "MPC based artificial pancreas: strategies for individualization and meal compensation," *Annual Reviews in Control*, vol. 36, no. 1, pp. 118–128, 2012.

[16] C. Hughes, S. D. Patek, M. Breton, and B. P. Kovatchev, "Anticipating the next meal using meal behavioral profiles: A hybrid model-based stochastic predictive control algorithm for T1DM," *Computer methods and programs in biomedicine*, vol. 102, no. 2, pp. 138–148, 2011.

[17] H. Chen, N. Paoletti, S. Smolka, and S. Lin, "Committed moving horizon estimation for meal detection and estimation in type 1 diabetes," in *American Control Conference (ACC)*, 2019.

[18] J. B. Rawlings, "Moving horizon estimation," *Encyclopedia of Systems and Control*, pp. 1–7, 2013.

[19] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–193.

[20] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the 13th international conference on artificial intelligence and statistics*, 2010.

[21] R. M. Neal *et al.*, "MCMC using hamiltonian dynamics," *Handbook of markov chain monte carlo*, vol. 2, no. 11, p. 2, 2011.

[22] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric statistical methods*. John Wiley & Sons, 2013, vol. 751.

[23] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical programming*, vol. 89, no. 1, pp. 149–185, 2000.

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

[25] D. M. Maahs, B. A. Buckingham, J. R. Castle, A. Cinar, E. R. Damiano, E. Dassau, J. H. DeVries, F. J. Doyle, S. C. Griffen, A. Haidar *et al.*, "Outcome measures for artificial pancreas clinical trials: a consensus report," *Diabetes Care*, vol. 39, no. 7, pp. 1175–1179, 2016.

[26] M. E. Wilinska, L. J. Chassin, C. L. Acerini, J. M. Allen, D. B. Dunger, and R. Hovorka, "Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes," *Journal of diabetes science and technology*, vol. 4, no. 1, pp. 132–144, 2010.

[27] L. Magni, D. M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli, "Model predictive control of type 1 diabetes: An in silico trial," *Journal of Diabetes Science and Technology*, vol. 1, no. 6, pp. 804–812, 2007.

[28] H. Lee, B. A. Buckingham, D. M. Wilson, and B. W. Bequette, "A closed-loop artificial pancreas using model predictive control and a sliding meal size estimator," *Journal of Diabetes Science and Technology*, vol. 3, no. 5, pp. 1082–1090, 2009.

[29] F. Cameron, B. W. Bequette, D. M. Wilson, B. A. Buckingham, H. Lee, and G. Niemeyer, "A closed-loop artificial pancreas based on risk management," *Journal of diabetes science and technology*, vol. 5, no. 2, pp. 368–379, 2011.

[30] G. M. Steil, A. E. Panteleon, and K. Rebrin, "Closed-loop insulin delivery—the path to physiological glucose control," *Advanced drug delivery reviews*, vol. 56, no. 2, pp. 125–144, 2004.

[31] L. M. Huyett, E. Dassau, H. C. Zisser, and F. J. Doyle III, "Design and evaluation of a robust PID controller for a fully implantable artificial pancreas," *Industrial & engineering chemistry research*, vol. 54, no. 42, pp. 10 311–10 321, 2015.

[32] R. Nimri, E. Atlas, M. Ajzensztejn, S. Miller, T. Oron, and M. Phillip, "Feasibility study of automated overnight closed-loop glucose control under MD-logic artificial pancreas in patients with type 1 diabetes: the DREAM project," *Diabetes technology & therapeutics*, vol. 14, no. 8, pp. 728–735, 2012.

[33] M. De Paula, L. O. Ávila, and E. C. Martínez, "Controlling blood glucose variability under uncertainty using reinforcement learning and Gaussian processes," *Applied Soft Computing*, vol. 35, pp. 310–332, 2015.

[34] E. Daskalaki, P. Diem, and S. Mougiakakou, "Model-free machine learning in biomedicine: Feasibility study in type 1 diabetes," *PloS one*, 2016.

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

[36] J. F. de Canete, S. Gonzalez-Perez, and J. Ramos-Diaz, "Artificial neural networks for closed loop control of in silico and ad hoc type 1 diabetes," *Computer methods and programs in biomedicine*, vol. 106, no. 1, pp. 55–66, 2012.

[37] S. M. Pappada, B. D. Cameron, and P. M. Rosman, "Development of a neural network for prediction of glucose concentration in type 1 diabetes patients," *Journal of diabetes science and technology*, vol. 2, no. 5, pp. 792–801, 2008.

[38] C. Pérez-Gandía, A. Facchinetti, G. Sparacino, C. Cobelli, E. Gómez, M. Rigla, A. de Leiva, and M. Hernando, "Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring," *Diabetes technology & therapeutics*, vol. 12, no. 1, pp. 81–88, 2010.

[39] S. Dutta, T. Kushner, and S. Sankaranarayanan, "Robust data-driven control of artificial pancreas systems using neural networks," in *International Conference on Computational Methods in Systems Biology*, 2018.

[40] K. Li, C. Liu, T. Zhu, P. Herrero, and P. Georgiou, "GluNet: A deep learning framework for accurate glucose forecasting," *IEEE journal of biomedical and health informatics*, 2019.

[41] H. Daumé, J. Langford, and D. Marcu, "Search-based structured prediction," *Machine learning*, vol. 75, no. 3, pp. 297–325, 2009.

[42] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *the 30th International Conference on Neural Information Processing Systems*, 2016.

[43] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," in *2016 IEEE international conference on robotics and automation (ICRA)*, 2016.

[44] B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable MPC for end-to-end planning and control," in *the 32nd International Conference on Neural Information Processing Systems*, 2018.

[45] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control," in *International Conference on Learning Representations*, 2019.

[46] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *Annual American control conference (ACC)*, 2018.

[47] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.

[48] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[49] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust mpc and neural network control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.

[50] C. Cronrath, E. Jorge, J. Moberg, M. Jirstrand, and B. Lennartson, "BAgger: A Bayesian algorithm for safe and query-efficient imitation learning," in *Machine Learning in Robot Motion Planning – IROS Workshop*, 2018.

[51] K. Lee, K. Saigol, and E. A. Theodorou, "Safe end-to-end imitation learning for model predictive control," in *International conference on robotics and automation (ICRA)*, 2019.

[52] Y. Gal, R. McAllister, and C. E. Rasmussen, "Improving PILCO with bayesian neural network dynamics models," in *Data-Efficient Machine Learning workshop, ICML*, 2016.

[53] K. Polymenakos, A. Abate, and S. Roberts, "Safe policy search using gaussian process models," in *the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.