Pre-emptive learning-to-defer for sequential medical decision-making under uncertainty

Shalmali Joshi* and Sonali Parbhoo* and Finale Doshi-Velez

13 September, 2021

Abstract

We propose SLTD ('Sequential Learning-to-Defer') a framework for learning-to-defer pre-emptively to an expert in sequential decision-making settings. SLTD measures the likelihood of improving value of deferring now versus later based on the underlying uncertainty in dynamics. In particular, we focus on the non-stationarity in the dynamics to accurately learn the deferral policy. We demonstrate our pre-emptive deferral can identify regions where the current policy has a low probability of improving outcomes. SLTD outperforms existing non-sequential learning-to-defer baselines, whilst reducing overall uncertainty on multiple synthetic and real-world simulators with non-stationary dynamics. We further derive and decompose the propagated (long-term) uncertainty for interpretation by the domain expert to provide an indication of when the model's performance is reliable.

1 Introduction

Machine learning (ML) methods are now being deployed for decision-making in complex domains such as loan approvals and criminal justice. In many cases, an available ML-based policy may not generalize to situations not encountered during training. In practice, it may be safer to defer to a human expert when using the policy may not improve outcomes. Many have considered the problem of learning to defer in myopic, non-sequential settings (e.g. Mozannar and Sontag [2020], Madras et al. [2017]).

In situations such as managing health, however, two key challenges remain. First, focusing on long-term outcomes is critical to decide *when* to defer to an expert. Deferring too late may lead to unintended harms that are difficult to recover from in the long term. Deferring too early may increase the burden on the domain expert. Second, learning to defer at the right time requires a well-characterized model of the dynamics to estimate the impact of delayed deferral, which may be difficult to estimate in non-stationary settings.

Existing methods for learning-to-defer to an expert aim to improve the performance of a prediction task e.g. Mozannar and Sontag [2020], Madras et al. [2017], Gennatas et al. [2020] by deferring to the expert. These methods defer to experts either based on the confidence of a model prediction or by characterizing the trade-off of paying a cost (to defer) and improving outcomes using human decisions. These approaches do not account for the sequential nature of decision-making settings, nor the non-stationary dynamics over time. Non-stationarity leads to propagated uncertainty that increases with time for longer-term decisions, and we demonstrate ignoring non-stationarity leads to underestimating this propagated uncertainty, resulting in delayed deferrals and worse long-term outcomes.

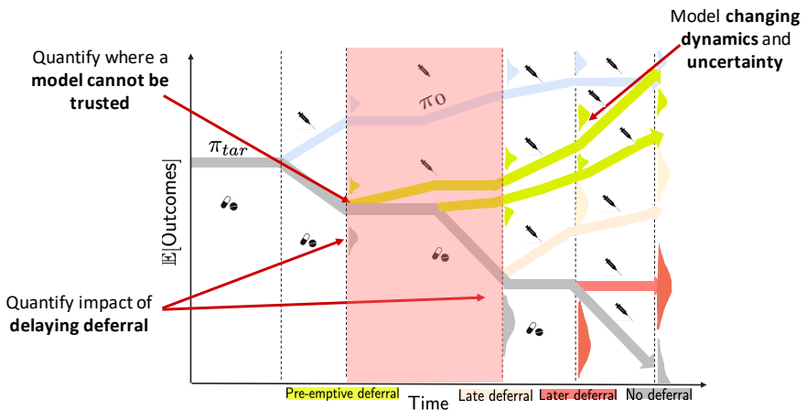


Figure 1: Overview of possible deferral strategies in a medical setting. The target policy π_{tar} recommends continuing a pill-based therapy, while the domain expert (π_0) suggests switching to a shot. Our approach (SLTD, green) defers in the shaded region where π_{tar} is unlikely to improve expected future outcomes beyond a certain threshold, reducing unintended consequences and managing the overall uncertainty. To do so, SLTD models changing disease dynamics, and accounts for the impact of delayed deferral, to identify when π_{tar} is unlikely to improve outcomes. Late deferral produces poorer outcomes and exacerbates uncertainty.

In contrast, we focus on learning to defer to a domain expert by accounting for *changing disease dynamics* modelled by a non-stationary Markov Decision Process (MDP), along with quantifying the impact of *delaying*

*Equal Contribution

Preprint. Under review.

deferral. Specifically, our work makes the following contributions: We develop a learn-to-defer approach for sequential settings using model-based reinforcement learning that defers to a human expert pre-emptively i.e. *as soon as we anticipate our model-based policy is unlikely to improve long-term outcomes (within a user-defined threshold)* (see illustration in Figure 1).We introduce a model of the non-stationary dynamics to reliably estimate the impact of delayed deferral, and show that SLTD can help manage propagated uncertainty and outperforms existing baselines on several different domains. We also demonstrate that under-estimation of the propagated uncertainty can lead to sub-optimal outcomes and learn mis-calibrated deferral policies. Finally, we interpret the agent's decisions to defer using SLTD by decomposing the sources of uncertainty, which could help improve outcomes beyond the behavior policy by guiding domain experts to, for instance, collect additional data where necessary or consult with other experts where uncertainty is high.

2 Related Work

Mixture-of-Experts (MoE). A number of methods focus on deciding between two or more policies to execute. For example, Jacobs et al. [1991], Jordan and Jacobs [1994] switch between different types of expertise in decision-making by partitioning the input space into different regions that may be assigned to different specialized sub-models. Variants of this framework enforce an explicit preference for a specific expert e.g. a human expert, and train other experts to complement the human expert [Pradier et al., 2021]. In sequential settings, Parbhoo et al. [2017], Gottesman et al. [2019], Parbhoo et al. [2018] combine parametric and non-parametric experts to learn more accurate estimates of the value function. Our work differs from these works in two ways: first, we focus on *pre-emptive deferral to human experts when future outcomes using the current ML-based policy are potentially undesirable*; and second, we model the impact of delayed deferral to decide *when* to defer.

Policy Improvement with Expert Supervision. Sonabend et al. [2020] use hypothesis testing to assess whether, at each state, a policy from a human expert would improve value estimates over a target policy *during training* to improve the target policy. In contrast, our work identifies the value of *delaying deferral to a human expert at test time*. Additionally, while expert supervision can significantly help during model development, such updates to policies may not be feasible, especially in deployment settings. In such cases, learning-to-defer with respect to a fixed target policy will be crucial for practical decision-making. Other works such as Chandak et al. [2020b,a] focus on safe policy optimization and improvement in a non-stationary MDP setting. In particular, Chandak et al. [2020a] assume that the non-stationarity in an MDP is governed by an exogenous process, or that past actions do not impact the underlying non-stationarity. Our work differs in two ways: first, we posit that model misspecification not only affects our estimates of the probability of deferring to an expert at each time step, but also affects the underlying uncertainty. In the sequential setting, this non-stationarity leads to *propagated uncertainty* that grows with time for longer term decisions. Second, we propose incorporating human expertise by deferring to a domain expert such that *future stochasticity can be controlled*.

Learning-to-defer to Human Expertise. Madras et al. [2017], Mozannar and Sontag [2020] propose models for triage, where only the most critical decisions are deferred to a medical expert. Here, the classifiers are trained based solely on the samples of an expert's decisions. Madras et al. [2017] train a separate rejection and prediction function, while Mozannar and Sontag [2020] learn a joint predictor for all targets and deferral. Madras et al. [2017] is conceptually closer to our work, but in a non-sequential setting. Other approaches such as Raghu et al. [2019], Wilder et al. [2020] first train a standard classifier on the data and then compute uncertainty estimates for this classifier and the human expert. The decision is ultimately deferred to the expert if the model is highly uncertain or can significantly benefit from deferral. Recently, Liu et al. [2021] propose incorporating uncertainty in Learning-to-Defer algorithms for classification tasks. Unlike these, we focus on the learning to-defer to a human expert in the non-stationary, sequential setting.

Decomposing Uncertainty for Interpreting Policies. Uncertainty, if well calibrated and communicated can help decision-makers understand the failure modes of a model [Bhatt et al., 2020, Tomsett et al., 2020, Zhang et al., 2020]. As a result, several methods have estimate predictive uncertainty for machine learning [Gal and Ghahramani, 2016, Guo et al., 2017]. In this work, we focus on capturing the *propagated uncertainty* in a sequential setting by learning a non-stationary dynamics model to learn a deferral policy. Second, we interpret the (different) sources of propagated uncertainty when SLTD defers to the expert. Decomposing the sources of uncertainty over predictions has been explored in classification and prediction settings [Yao et al., 2019, Depeweg et al., 2018]. In this work, we interpret two different sources of propagated long-term uncertainty, particularly modeling uncertainty and the stochasticity in the system at the time of deferral. Interpreting deferral by decomposing the uncertainty can help experts understand how to further manage uncertainty by potentially deviating from the expert policy e.g. by obtaining second opinions.

3 Sequential Learning-to-Defer

We now present the SLTD framework for pre-emptive deferral under uncertainty. SLTD consists of three key steps: first we learn a model of the non-stationary dynamics and use posterior sampling to capture uncertainty over this model. Next, we quantify the impact delaying deferral would have on the long-term outcomes. Based on this model uncertainty, the deferral policy is defined as the probability that the (long-term) outcome cannot be improved beyond some threshold by delaying deferral. Finally, we decompose the uncertainty at deferral time (provided by SLTD) to explain the decision to defer to a domain expert and highlight where

this uncertainty comes from. This information can subsequently be used by the expert to determine how to act.

Problem Setup. We consider a finite horizon MDP defined by $\mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, p_0, \gamma)$ where $\mathcal{S}$ indicates the state-space, $\mathcal{A}$ indicates the action-space, $\mathcal{P}$ the transition dynamics, $r : s \times a \to \mathbb{R}_+$ the reward function, p_0 the initial state distribution and the discount factor γ. Consider a fixed and known stochastic policy $\pi_{tar} : \mathcal{S} \times \mathcal{A} \to [0, 1]$. We consider non-stationary dynamics where $\mathcal{M}_t$ denotes the MDP at the t^{th} time in an episode. We assume the existence of a true set of non-stationary dynamics governing all episodes and denote it by $\{\mathcal{M}_t^*\}_t$. In the rest of the manuscript, $\mathcal{M}_t$ denotes a sample approximating the true dynamics $\mathcal{M}_t^*$. With slight abuse of notation we write $\mathcal{M}$ to refer to the dynamics of the MDP when clear from context.

We assume that we have observational data collected from some behavior policy π_0, denoted by $\mathcal{D} = \{s_{i,0}, a_{i,0}, r_{i,0}, \cdots, s_{i,T}, a_{i,T}, r_{i,T}\}_{i=1}^N$ where T is the episode-length and N is the number of episodes. The value of a policy π at t is given by $V_{\pi,t}^{\mathcal{M}}(s) = \mathbb{E}_{\mathcal{M}}[\sum_{j=t}^{T} \gamma^j r^j(s, a) | s_t = s, \pi]$. Deferral to an expert is denoted by the action $\perp$.

We now formalize how to recover a pre-emptive deferral policy with respect to a target policy.

Definition 1. *Let a target policy $\pi_{tar,t}(a|s)$ be such that such that there exists $\varnothing \subseteq s^t \subseteq \mathcal{S}\, \forall t \in \{0, 1, \cdots, T\}$ where $P(V_{\pi_{tar},t}^{\mathcal{M}*}(s) < V_{\perp,t}^{\mathcal{M}*}(s) - c) > \tau$ for constant cost of deferral c[1] and threshold $\tau > 0$, $\forall s \in s^t$. Let a policy $g_{\pi_{tar}}(s, t)$ be such that $g_{\pi_{tar}}(s, t) \triangleq \mathbf{1}(P(V_{\pi_{tar},t}^{\mathcal{M}*}(s) < V_{\perp,t}^{\mathcal{M}*}(s) - c) > \tau)$.*

Corollary 1. *By Definition 1, $g_{\pi_{tar}}(s, t)$, includes the earliest time in the episode where $P(V_{\pi_{tar},t}^{\mathcal{M}*}(s) < V_{\perp,t}^{\mathcal{M}*}(s) - c) > \tau$. Therefore, $g_{\pi_{tar}}(s, t)$ is also a pre-emptive deferral policy.*

Intuitively, τ can be considered a safety threshold that trades off the tolerance for outcome and the learned policy will defer often. Deferring using $g_{\pi_{\text{tar}}}(s, t)$ will thus reflect in a higher value and lower propagated uncertainty.

Definition 1 indicates that to reliably learn the deferral policy, we need to estimate the probability that outcomes will not improve. This in-turn identifies regions of the state-space that increase future uncertainty and do not improve long-term outcomes. To accurately estimate this probability, we should model all sources of uncertainty in the system, including the non-stationarity in the dynamics, and the uncertainty associated with our modeling assumptions. In the following we describe how to account for all these sources of uncertainty and then decide when to defer.

Algorithm 1 Sequential Learning to Defer

Input: Posterior estimates $\{p_t(\cdot|\mathcal{D})\}_{t=0}^T$, target policy π_{tar}, behavior policy π_0.
Initialization: Deferral function $g_{\pi_{\text{tar}}}(s, t) = 0$ for all $s \in \mathcal{S}$ and $t \in \{1, 2, \cdots, T\}$.
for $t \in \{T, T-1, \cdots, 1\}$ **do**
  for $s \in \mathcal{S}$ **do**
    Compute $\{V_{\pi_{\text{tar}},t}^{\mathcal{M}}(s)\}$ and $\{V_{\perp,t}^{\mathcal{M}}(s) - c\}\, \forall \mathcal{M}$
    Update $g_{\pi_{\text{tar}}}(s, t) \leftarrow\approx \mathbb{E}_{\mathcal{M} \sim p_t(\cdot|\mathcal{D})}[\mathbf{1}(V_{\pi_{\text{tar}},t}^{\mathcal{M}}(s) > V_{\perp,t}^{\mathcal{M}}(s) - c) > \tau]$
  end for
end for
return $g_{\pi_{\text{tar}}}(s, t)$

Modeling non-stationary dynamics with posterior sampling. To quantify all sources of (propagated) uncertainty, our approach is based on Bayesian RL. Specifically, we model the posterior distribution over the non-stationary MDPs using Bayesian inference to estimate the non-stationary dynamics $\mathcal{M}_t$ and the reward function $r(s_t, a_t)$. We make parametric assumptions on the family of these distributions and use conjugate priors over the parameters of the distributions. Learning a non-stationary model allows us to capture sources of irreducible uncertainty in the system. We can now estimate the impact of delayed deferral, by averaging over this uncertainty.

Quantifying the impact of delaying deferral. Using the above samples, we can estimate the likelihood of improvement by deferring based on Definition 1. That is, we choose to defer to an expert when $g_{\pi_{\text{tar}}}(s, t) \approx \mathbb{E}_{\mathcal{M} \sim p_t(\cdot|\mathcal{D})}[\mathbf{1}(V_{\pi_{\text{tar}},t}^{\mathcal{M}} < V_{\perp,t}^{\mathcal{M}} - c) > \tau]$.

We test the objective at every time-point to update $g_{\pi_{\text{tar}}}(s, t)$ if not-deferring is unlikely to improve outcomes. Note that the estimates $V_{\pi_{\text{tar}},t}^{\mathcal{M}}(s, a)$ will also account for *future* potential deferrals thus allowing us to learn a pre-emptive policy. Note that here we are focused on deferring for a fixed target policy, rather than policy improvement, which is particularly useful in situations where such updates may not be permitted due to safety concerns. In practice, $V_{\perp,t}^{\mathcal{M}}(s)$ could be estimated based on how a clinician might actually choose to treat the patient and deviate from our reference policy π_0. Our procedure is summarised in Algorithm 1.

4 Decomposing the uncertainty at deferral

A key contribution of our work is that we provide a justification to the clinician for the need to defer at every point of deferral identified by Algorithm 1. We do so by explicitly highlighting sources of uncertainty resulting in deferral. Apart from focusing on total uncertainty, we also convey different sources of uncertainty

[1]We introduce a constant cost for deferral so as to defer to the domain expert only when necessary

by decomposing the total uncertainty at deferral. That is, we consider *epistemic/modeling uncertainty*, which captures whether the model has high uncertainty, as well as *aleatoric uncertainty* resulting from the stochasticity itself. Conveying both these sources of uncertainty can subsequently help the clinician determine how to act.

Concretely, let t_d correspond to the first realization such that $g_{\pi_{\text{tar}}}(s_{t_d}, t_d) = 1$ in state s_{t_d}. Then, we are interested in the reward (and uncertainty over the reward) at time T due to deferring at time t_d.When we defer, we rely on the behavior policy. To estimate the total and epistemic uncertainty, we leverage our posterior sampling framework once again. For any time t, the MDP $\mathcal{M}$ is a Dirichlet sample (for a given state-action pair) and is denoted by μ_t. These dirichlet samples allow us to capture modeling uncertainty. The model parameters that parameterize the distribution over the MDPs is denoted by θ_t and are indexed by the state-action pair (s_t, a_t). That is, we sample from posterior distribution $p(\theta_{t'}|\mathcal{D})$, followed by sampling the posterior MDP dynamics $\mathcal{M}_t \triangleq \mu_t \sim p(\mu_{t'}|\theta'_t(s_{t'}, a_{t'}))$, which in-turn allows us to sample the next state. For specificity, we denote the state at the time of deferral t_d by s_{t_d}. The outcome we are interested in is given by:

$$\mathbb{E}[r_T|s_{t_d}, \mu_{t_d}] = \int_{s_{t_d+1}}^{s_T} \int_{a_{t_d}}^{a_T} \int_{\mu_{t_d+1}}^{\mu_T} \int_{\theta_{t_d}}^{T} r(s_T, a_T) \prod_{t'=t_d+1}^{T} p(s_{t'}|\mu_{t'}) p(\mu_{t'}|\theta'_t(s_{t'}, a_{t'})) \pi_{t'}(a_{t'}|s_{t'}) p(\theta_{t'}|\mathcal{D}) d\mathbf{s}_{t_d+1}^T d\mathbf{a}_{t_d}^T d\boldsymbol{\mu}_{t_d+1}^T d\boldsymbol{\theta}_{t_d}^T \tag{1}$$

Here for brevity, integrands are written in short-hand: $\mathbf{s}_{t_d+1}^T = \{s_{t_d+1}, s_{t_d+2}, \cdots, s_T\}$ (analogously for other quantities, hidden in the above equation) and $\pi_{t'} = \pi_0$ for $t' = t_d$ and can be either the the behavior or target policy for all future times $t' > t_d$ to account for potential *future* deferrals.

First, note that we maintain only one estimate of parameter $\theta_{t'}$ and sample posterior samples $\mu_{t'}$ from this distribution. That is, $p(\theta_{t'}|\mathcal{D}) = \delta_{\theta_{t'}}$ which is a delta function centered at $\theta_{t'}\, \forall\, t' \in \{0, 1, 2, \cdots, T\}$. Thus, the epistemic uncertainty we capture is due to the uncertainty over dynamics under fixed parameters. High variability in sampling $\mu_{t'}$ indicates the current state $s_{t'}$ (and action) is out-of-distribution. We propose that the (multi-step) total uncertainty can be decomposed using the law of total variance as follows:

$$\underbrace{\text{Var}(r_T|s_{t_d}, \mathcal{D})}_{\text{Total Uncertainty}} = \underbrace{\mathbb{E}_{\mu_{t_d} \sim p(\mu_{t_d}|\mathcal{D})}[\text{Var}(r_T|\mu_{t_d}, s_{t_d}, \mathcal{D})]}_{\text{Irreducible/ Aleatoric Uncertainty}} + \underbrace{\text{Var}_{\mu_{t_d} \sim p(\mu_{t_d}|\mathcal{D})}(\mathbb{E}[r_T|\mu_{t_d}, s_{t_d}, \mathcal{D}])}_{\text{Epistemic/Modeling Uncertainty}} \tag{2}$$

The second term in the above equation captures the variance *conditioned* on knowledge of the model at deferral time (μ_{t_d}), therefore marginalizing only over current aleatoric uncertainty and future total uncertainty (i.e. over future $\mu_{t'}$, future deferral, and reward). This term therefore captures *propagated uncertainty due to modeling/epistemic uncertainty at* t_d. The first term averages over the variance due to μ_{t_d} and thus captures *propagated total uncertainty to due to aleatoric uncertainty at* t_d. We approximate this integration using Monte-carlo sampling.

High *propagated epistemic uncertainty* can convey that the current uncertainty of model prediction (of the dynamics) is high enough but could be improved if additional data in this region could be collected to improve the reducible sources of uncertainty. High *propagated aleatoric uncertainty* indicates high variability in the patient's dynamics that may need to be managed with careful interventions and is otherwise not manageable. Based on the communicated uncertainty, the clinician may choose to deviate from their usual practice for rare cases with high epistemic uncertainty and instead consult multiple experts and/or attempt experimental treatments.

	Synthetic (mean ± 2 s.e.)	Sepsis-diabetes (mean ± 2 s.e)	Diabetes (mean ± 2 s.e.)
Value (SLTD-π_{tar})	5.845 ± 0.04	**-0.337 ± 0.006**	**65.162 ± 0.364**
Value (SLTD-stationary-π_{tar})	**5.937 ± 0.039**	-1.46 ± 0.01	50.449 ± 0.301
Value (SLTD-one step-π_{tar})	4.867 ± 0.041	-0.386 ± 0.006	64.869 ± 0.362
Value (π_{tar})	4.879 ± 0.04	-0.831 ± 0.008	25.801 ± 0.262
Augmented-MDP	2.235 ± 3.227	-2.405 ± 0.606	-0.898 ± 0.022
Madras et. al.	-0.002 ± 0.0	-2.817 ± 0.491	0.336 ± 0.032

Table 1: Expected rewards for SLTD compared with baselines. The table shows the value of using π_{tar} with our deferral method SLTD and without, including the one-step and stationary variants. Higher values indicate better performance. For all datasets, we see a significant benefit due to early deferral. Augmented-MDP baselines performs poorly as it defers only when the rewards are suboptimal. SLTD-one step only relies on immediate rewards and uncertainty thus failing to improve long-term outcomes for Synthetic data. However there are benefits to myopic deferral for Sepsis-diabetes and Diabetes as π_{tar} also has suboptimal rewards in regions where it takes random actions. The benefits are less compared to SLTD-π_{tar}. The mis-specified model, SLTD-stationary performs reasonably well for synthetic simulation, it underestimates long-term uncertainty when the system is highly non-stationarity as in Diabetes data. Finally, the supervised Madras et. al. baseline is myopic and unable to maximize long-term reward. SLTD-one-step focuses on immediate reward, therefore unable to perform well for Synthetic data where modeling long-term reward is critical; whereas it shows some benefit for Sepsis-diabetes and Diabetes where π_{tar} can have suboptimal immediate rewards in the deferral region.

5 Experiments

We conduct our experimental analysis to evaluate the ability of SLTD to defer pre-emptively for a known and fixed policy π_{tar} to behavior policy π_0. More specifically, we test the benefits of key aspects of SLTD: i) the

ability of SLTD to identify regions where the target policy is unreliable (as defined in Definition 1), ii) the utility of learning the deferral policy by estimating the impact of delaying deferral in these regions, and iii) the utility of modeling the non-stationarity in the system. To that end we first design a synthetic data simulation where i) the region of early deferral is known apriori by careful design of π_{tar} and ii) true non-stationary dynamics are simulated. Further we test our method on a non-stationary version of the sepsis-diabetes simulator introduced by Oberst and Sontag [2019] and a non-stationary diabetes simulator [Chandak et al., 2020b].[2] While Chandak et al. [2020b] introduce non-stationarity across episodes, our setting is more variable, as we learn to defer when stochasticity increases over time in every episode. In the following we describe the experimental setup and baselines in detail.

Synthetic Data: A Toy Demonstration. We design a synthetic data simulation with 8 discrete states. All samples start at state 0 and progress toward a sink state at 7. State 6 has low reward (-5) while all other states have a reward of $+1$. Action 0 reduces likelihood of landing in stage 6 and action 1 increases the likelihood of reaching state 6. π_{tar} increases the chances to reach state 6 by taking unfavorable actions (1) in states 2, 3, 4 at times $3 \leq t \leq 8$. The dynamics are non-stationary such that stochasticity of transitions progressively increases for higher states as well as over time. Note that rewards are designed to be positive in the states $\{2, 3, 4\}$ so that sub-optimal outcomes can only be observed in the future. Hence this example will demonstrate key aspects of pre-emptive deferral in combination with modeling non-stationary compared to baselines. A pre-emptive policy learnt will defer to a relatively better policy π_0 between the regions $\{2, 3, 4\}$ and as early as time $t = 3$. A myopic method will defer in state 6 irrespective of the time. A mis-specified model (that assumes stationarity of dynamics) will underestimate propagated uncertainty and learn mis-calibrated deferral probabilities.

Sepsis-diabetes. The Sepsis-diabetes simulator is designed for sepsis treatment of diabetic and non-diabetic subpopulations [Oberst and Sontag, 2019]. The simulator uses physiological measurements (heart-rate, glucose level, blood pressure and oxygen concentrations) discretized to a total of 720 states. Interventions include mechanical ventilation, vasopressors, and antibiotics. The reward is high $+1$ when all measurements are 'normal', treatments have been discontinued while it is -1 when all measurements are simultaneously not 'normal' and 0 otherwise. We modify this simulator to introduce non-stationarity over time, specifically increasing stochasticity towards completely random transitions by increasing the likelihood of fluctuations for heart-rate, blood-pressure and glucose transitions over time. We only sample diabetic patients as they have higher baseline stochasticity based on glucose levels. A non-stationary behavior policy is estimated by using policy-iteration on trajectories of size $N = 200$ fixed-length ($T = 10$) episodes. The expert policy π_0 we defer to is an ϵ-greedy version of the behavior policy which can also non-myopically degrade (toward uniform) over time. The target policy π_{tar} is such that it deteriorates to random at a specific time $t \geq 3$ (for all states). Thus $t = 3$ is the precise time-point of pre-emptive deferral. Additional details are in the Appendix.

Real-world simulator: Diabetes Data. We use open-source implementation of the FDA approved Type-1 Diabetes Mellitus simulator (T1DMS) for modelling treatment of Type-1 diabetes. We sample 10 adolescent patient trajectories (episodes) over 24 hours (aggregated at 15 minute intervals). Glucose levels are discretized into 6 states according to ranges suggested in the simulator. Combination interventions of insulin and bolus are discretized to generate a total of 8 actions. We introduce non-stationarity within each episode by increasingly changing the adolescent patient properties to an alternative patient over the episode. This significantly affects the utility of the initial target policy necessitating deferral as the patient properties change over the course of the day. The non-stationary behavior policy is estimated using Q-learning. We defer to an epsilon-greedy version of such a policy that is made to degrade over time by increasing stochasticity. As before, the target policy resembles the behavior policy, except is changed to take random actions in the time-window $35(= 8\text{hrs}) \leq t \leq 50(= 13\text{hrs})$ for all states. This is the desired region of pre-emptive deferral.

5.1 Baselines

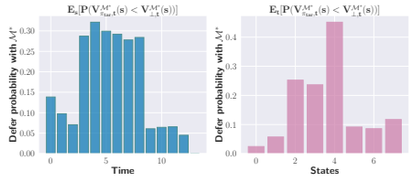
To the best of our knowledge, there are no prior works focusing on pre-emptive deferral to expert (for stationary or non-stationary) sequential settings. We compare to the following baselines.

Madras et. al. [Madras et al., 2017]: This is a supervised learning-to-defer method not designed for sequential or non-stationary domains, that learns a deferral function along with a myopic treatment policy. This baseline allows us to compare to myopic deferral decisions. We train this baseline with action targets. Note that while this baseline learns a separate regressor to predict actions, we modify it to use π_{tar} to learn deferral for π_{tar} as with SLTD.

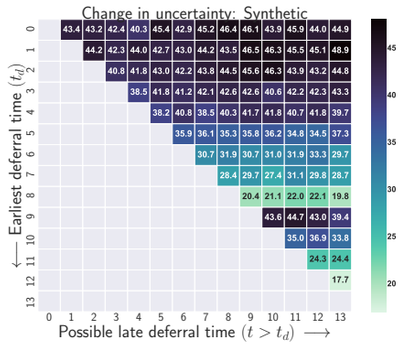
Augmented-MDP: We consider a baseline that defers permanently to the expert (as opposed to multi-step deferral as in SLTD) using an Augmented MDP. Augmented MDPs are commonly used to incorporate domain knowledge. To learn a deferral policy, we augment the action and state-space, so that the action-space is $\mathcal{A} \cup \perp$ and the augmented state-space is $\mathcal{S} \cup s_{defer}$ (s_{defer} is the deferred state). The transition dynamics are updated to reflect the augmented action and state-space transitions. This baseline models non-stationary dynamics, and is designed to defer in sequential settings. However since the dynamics have to be augmented to defer permanently to the expert, this baseline incurs a larger deferral cost in practice.

SLTD-one step: We also compare to a myopic version of SLTD that chooses to defer only based on the immediate reward. This baseline is called SLTD-one step. The key difference with the myopic Madras et. al.

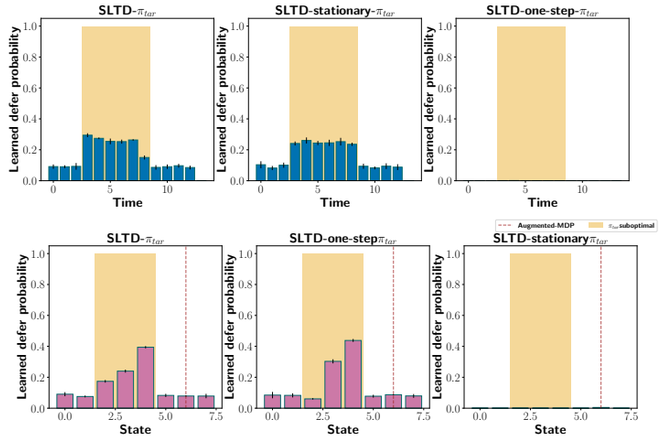
[2] Jinyu Xie. Simglucose v0.2.1 (2018) [Online]. Available: `https://github.com/jxx123/simglucose`. Accessed on: 07-24-2021.



(a) Deferral probabilities under true dynamics over time (left) and states (right).



(b) Heatmap of increased uncertainty (variance) due to delayed deferral. The y-axis shows SLTD's earliest chosen deferral time. The x-axis shows possible later deferral times. The values in each cell are the relative increase in variance over cumulative reward if we defer *after* the SLTD's first deferral time. Except when SLTD itself defers late, delaying deferral always increases uncertainty.



(c) Learned deferral probabilities for SLTD, its variants, and Augmented-MDP. Top row shows learned probabilities over time (marginalized by state) and bottom over states (marginalized by time). Shaded yellow indicates the region of pre-emptive deferral. SLTD-π_{tar} which models non-stationarity best approximates the distribution on the left as can be seen by higher deferral likelihood early in the shaded yellow region (top row). SLTD-stationary does not learn calibrated probabilities in the yellow region over time. The deferral probabilities w.r.t the states (bottom row) are less well calibrated due to imperfect model estimation and potential bias in data collected from π_0. SLTD-π_{tar} and SLTD-stationary still learn state 4 has highest likelihood of deferral. SLTD-one-step (seen as a small bump on the state histogram) and Augmented-MDP (dotted red line) only defer when in state 6. Augmented-MDP defers with probability 1 in state 6.

baseline here is that SLTD-one step models the uncertainty on the immediate reward of deferring versus not deferring, while Madras et. al. defers based on their confidence in predicting the next action.

SLTD-stationary: To assess the impact of misspecifying the stationarity of the dynamics on the learned deferral policy, we further compare to a version of SLTD with assumes the dynamics (and rewards) are stationary.

5.2 Implementation Details

Data is generated using behavior policy π_0 in all cases. We estimate posteriors of non-stationary dynamics with Dirichlet priors over each state-action pair and learn non-stationary deferral policies. Additional details are in the Appendix.

6 Results

We validate the following aspects of SLTD using the empirical evaluation. First, we assess whether SLTD learns a pre-emptive deferral policy for a given target policy π_{tar}. Second, we evaluate whether such pre-emptive deferral improves long-term cumulative outcomes over methods that myopically recommend deferral or those are not pre-emptive. We also analyze the change in uncertainty due to deferral decisions of SLTD (and its variants). Finally we interpret the deferral decision by conveying the decomposed uncertainty that led to deferral.

SLTD improves long-term outcomes. Cumulative rewards obtained from all methods are in Table 1. SLTD (and the stationary variant) that optimize for long-term outcomes outperform other baselines for all datasets. SLTD is able to significantly improve over the target policy π_{tar}. In cases where the systematic non-stationarity is not too high, as in Synthetic data, the mis-specification induced by using the stationary variant of SLTD is less concerning. In this case, a stationary model gains from the additional samples available for dynamics model estimation resulting in comparable performance to modeling non-stationarity. However, in the case of the Diabetes simulator with significantly more induced non-stationarity, the improvement is significant. For all datasets, myopic deferral, as learned by Madras et. al. is unable to improve long term cumulative rewards. Here, focusing on predictive confidence, as the Madras et. al. baseline does, is not sufficient to identify regions of the state-space that increase propagated uncertainty by deploying the model (π_{tar}). Note that π_{tar} actually has reasonably high reward (+1) in regions where it takes sub-optimal actions for Synthetic data (since these actions worsen long-term reward), which is challenging for SLTD-one-step to identify. In Sepsis-diabetes and Diabetes data, SLTD-one-step gains from focusing on immediate reward and uncertainty when π_{tar} has suboptimal rewards in the deferral region. This allows it to defer early. Augmented-MDP fails to incur meaningful benefit and is worse than the nominal π_{tar}. Augmented-MDP also focuses on the immediate reward, and thus defers precisely only when the immediate reward is sub-optimal.

SLTD learns a pre-emptive deferral policy. Figure 2c shows the learned deferral probabilities from SLTD and its variants. Top row shows the probabilities over time and bottom row shows the probabilities

	Total Uncertainty	Epistemic Uncertainty	Mean Outcome
Synthetic data ($t_d = 6$)	27.00	0.267	2.52
Sepsis-diabetes ($t_d = 3$)	1.474	0.129	-4.15
Diabetes ($t_d = 17$)	2921.49	35.207	71.950

Table 2: Interpreting first time of deferral for a sample trajectory. Modeling uncertainty remains low in all cases whereas in comparison, total variance is high. This indicates irreducible stochasticity of the dynamics is the primary source of uncertainty in all datasets. We observe a similar pattern for other examples (not included here due to space constraints).

over states. Figure 2a shows the probabilities under the true model $\mathcal{M}^*$. Thus, SLTD-π_{tar} approximates the true probabilities better compared to the SLTD-stationary and SLTD-one-step. The likelihood of deferral is higher at the beginning of this region compared to the stationary variant that does not learn calibrated deferral probabilities over time. The SLTD-one-step baseline only defers in state 6, when the immediate reward is negative, as can be seen by the bump in deferral probabilities over states (bottom rows). Similarly, Augmented-MDP (denoted by the red dashed line when it defers) is not pre-emptive and deterministically defers when the reward is sub-optimal indicating Augmented-MDP also does not learn a pre-emptive policy by focusing on regions where the reward is unfavorable.

Figures 4 and 5 in the Appendix show the learned deferral policy for Sepsis-diabetes and Diabetes data. Qualitatively, learned deferral policies over time are similar with differences in deferral probabilities for states for these datasets. The qualitative differences in probabilities reflect in the mean improvement in outcomes as demonstrated in Table 1. Although the SLTD variants find the right time of deferral, the SLTD-π_{tar} policy for Diabetes learns to defer with higher probability earlier in the sub-optimal region of π_{tar} in comparison to the stationary variant. The deferral likelihoods for different states is significantly different indicating strong differences in qualitative behavior of SLTD variants.

Pre-emptively deferring using SLTD reduces overall uncertainty. In general, a higher value and lower uncertainty here demonstrates the utility of quantifying *propagated* uncertainty. Figure 2b shows a heatmap that demonstrates the propagated uncertainty by delaying deferral. Specifically, the y-axis in the heatmap denotes the earliest deferral time chosen by SLTD-π_{tar}. Conditioned on this decision, we then test the effect of delaying deferral by varying amounts to the times denoted on the x-axis. Thus each row in the heatmap demonstrates the effect of delaying deferral *after* SLTD-π_{tar}'s chosen time. If SLTD recommends earlier deferral (top rows on the y-axis), the increase in uncertainty is higher by virtue of delaying deferral to the time point denoted on the x-axis. Similar results for other variants of SLTD, i.e. SLTD-stationary and SLTD-one step as well as for other datasets are in the Appendix. When the non-stationarity is not high, as discussed before, SLTD-stationary benefits significantly from additional samples. This results in relatively better characterization of and reduction in uncertainty resulting in comparable reduction in overall uncertainty. SLTD-one step is unable to reduce uncertainty as much as these variants for Synthetic data where long-term modeling is critical. The change in uncertainty is comparable to SLTD for the Sepsis-diabetes and Diabetes data indicating that in regions where π_{tar} is designed to be sub-optimal, rewards are relatively lowered suggesting some benefits to deferring myopically.

Decomposing uncertainty in SLTD can help interpret deferral. The reduction in epistemic uncertainty is promising. Conveying the high uncertainty (as an interpretation of deferral) along with the type of uncertainty to a domain expert can enable them to identify the dominant source of uncertainty that resulted in a deferral in relation to their own standard practice (expert policy). Table 2 shows this decomposition for a few selected deferral times for all datasets. In all cases, the modeling/epistemic uncertainty is a small fraction of the total uncertainty. This suggests that the systematic non-stationarity is the dominant source of uncertainty which generally cannot be reduced by collecting data or second opinions. However, knowledge of the contribution of the model/epistemic uncertainty can enable users to further improve decision-making by clinicians choosing to potentially rely on their standard expert behavior or by other means (like second opinions).

6.1 Limitations

While quantifying the uncertainty when deferring is an important aspect of our method, there is an additional computational cost associated with modeling non-stationarity. Additionally, we assumed that data are generated by a single behaviour policy which is in general, an untestable assumption that may not necessarily hold in practice.

7 Discussion

In this work we formulate the problem of learning-to-defer in a sequential decision-making setting. Our goal is to learn a pre-emptive deferral policy, where a deferral policy is considered pre-emptive if it defers in regions where delaying defer has a higher likelihood of leading to worse long-term outcomes. We learn a pre-emptive deferral probability by approximating the likelihood of worse outcomes under systematic uncertainty.

Non-stationary dynamics can exacerbate this propagated uncertainty while increased stochasticity over time can further hinder our ability to "recover" (still obtain high reward) from executing a sub-optimal policy. Based on this insight, we estimate the deferral probabilities by modeling the underlying non-stationary dynamics. We showed that the proposed method SLTD learns pre-emptive deferral policy in comparison to other non-pre-emptive methods. Myopic methods designed to rely on prediction confidence do not defer pre-emptively as they rely heavily on their confidence, thus failing to identify regions where the current policy is unreliable. Augmented-MDP baselines account for sequential outcomes and model the underlying non-stationary dynamics however, are unable to pre-emptively defer by focusing on regions where reward is sub-optimal. Relying on an Augmented-MDP alone can result in delayed deferral. We also demonstrated that pre-emptively deferring can manage propagated uncertainty which we further interpreted by decomposing its effect on *long-term* outcomes. Evaluating the impact of these policies on long-term improvements as well as on a clinician's ability to intervene in a more informed manner is an exciting avenue for future work, including user-studies. Relaxing the assumptions about knowledge of the dynamics is left for future work.

References

Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Gauthier Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. *arXiv preprint arXiv:2011.07586*, 2020.

Yash Chandak, Scott M Jordan, Georgios Theocharous, Martha White, and Philip S Thomas. Towards safe policy improvement for non-stationary mdps. *arXiv preprint arXiv:2010.12645*, 2020a.

Yash Chandak, Georgios Theocharous, Shiv Shankar, Martha White, Sridhar Mahadevan, and Philip Thomas. Optimizing for the future in non-stationary mdps. In *International Conference on Machine Learning*, pages 1414–1425. PMLR, 2020b.

Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/gal16.html`.

Efstathios D. Gennatas, Jerome H. Friedman, Lyle H. Ungar, Romain Pirracchio, Eric Eaton, Lara G. Reichmann, Yannet Interian, José Marcio Luna, Charles B. Simone, Andrew Auerbach, Elier Delgado, Mark J. van der Laan, Timothy D. Solberg, and Gilmer Valdes. Expert-augmented machine learning. *Proceedings of the National Academy of Sciences*, 117(9):4571–4577, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1906831117. URL `https://www.pnas.org/content/117/9/4571`.

Omer Gottesman, Yao Liu, Scott Sussex, Emma Brunskill, and Finale Doshi-Velez. Combining parametric and nonparametric models for off-policy evaluation. In *International Conference on Machine Learning*, pages 2366–2375. PMLR, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

Jessie Liu, Blanca Gallego, and Sebastiano Barbieri. Incorporating uncertainty in learning to defer algorithms for safe computer-aided diagnosis. *arXiv preprint arXiv:2108.07392*, 2021.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. *arXiv preprint arXiv:1711.06664*, 2017.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. *arXiv preprint arXiv:2006.01862*, 2020.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

Sonali Parbhoo, Jasmina Bogojeska, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Combining kernel and model based learning for hiv therapy selection. *AMIA Summits on Translational Science Proceedings*, 2017:239, 2017.

Sonali Parbhoo, Omer Gottesman, Andrew Slavin Ross, Matthieu Komorowski, Aldo Faisal, Isabella Bon, Volker Roth, and Finale Doshi-Velez. Improving counterfactual reasoning with kernelised dynamic mixing models. *PloS one*, 13(11):e0205839, 2018.

Melanie F Pradier, Javier Zazo, Sonali Parbhoo, Roy H Perlis, Maurizio Zazzi, and Finale Doshi-Velez. Preferential mixture-of-experts: Interpretable models that rely on human expertise as much as possible. *arXiv preprint arXiv:2101.05360*, 2021.

Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

Aaron Sonabend, Junwei Lu, Leo Anthony Celi, Tianxi Cai, and Peter Szolovits. Expert-supervised reinforcement learning for offline policy learning and evaluation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18967–18977. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/daf642455364613e2120c636b5a1f9c7-Paper.pdf`.

Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 1(4):100049, 2020.

Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. *arXiv preprint arXiv:2005.00582*, 2020.

Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of uncertainty quantification for bayesian neural network inference. *arXiv preprint arXiv:1906.09686*, 2019.

Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305, 2020.

8 Appendix

8.1 Datasets

Measurement	Stationary fluctuation probability	Modified probabilities T=episode length t=time in episode
Heart-rate	0.1	$\min(0.5, 0.1 + 0.8t1.0/T)$
	0.2	$2\min(0.5, 0.1 + 0.8t1.0/T)$
Systolic- BP	0.1	$\min(0.5, 0.1 + 0.5t1.0/T)$
	0.2	$2\min(0.5, 0.1 + 0.5t1.0/T)$
Percoxyg.	0.1	$\min(0.5, 0.1 + 0.5t1.0/T)$
	0.2	$2\min(0.5, 0.1 + 0.5t1.0/T)$
Glucose (Diabetes only)	0.3	$\min(0.5, 0.3 + 0.5t1.0/T)$
	0.6	0.6

Table 3: Modified sepsis-diabetes simulator to induce non-stationarity

Sepsis-Diabetes simulator. The Sepsis-diabetes simulator is designed for sepsis treatment of diabetic and non-diabetic subpopulations [Oberst and Sontag, 2019]. The simulator uses physiological measurements (heart-rate, glucose level, blood pressure and oxygen concentrations) discretized to a total of 720 states. Interventions include mechanical ventilation, vasopressors, and antibiotics. The reward is high $+1$ when all measurements are 'normal', treatments have been discontinued while it is -1 when all measurements are simultaneously not 'normal' and 0 otherwise. We modify this simulator to introduce non-stationarity over time, specifically increasing stochasticity towards completely random transitions by increasing the likelihood of fluctuations for heart-rate, blood-pressure and glucose transitions over time. We only sample diabetic patients as they have higher baseline stochasticity based on glucose levels. In particular, the fluctuations are modified from the original simulator as shown in Table 3.

Diabetes simulator. We use open-source implementation of the FDA approved Type-1 Diabetes Mellitus simulator (T1DMS) for modelling treatment of Type-1 diabetes. The simulator models managing an in-silico patient's blood glucose levels when consuming a meal. If the blood glucose level is either too high (hyperglycemia) or too low (hypoglycemia), this can have fatal consequences such as organ failure. As a result, a clinician must administer an insulin dosage to minimize the risk of such events. While a doctor's initial dosage prescription is usually available, the insulin sensitivity of a patient's internal organs changes over time, thereby introducing non-stationarity that should be accounted for. We sample 10 adolescent patient trajectories (episodes) over 24 hours (with measurements aggregated at 15 minute intervals). Glucose levels are discretized into 6 states according to ranges suggested in the simulator (additional details are in the appendix). Further, insulin and bolus intervention combinations are discretized to generate a total of 8 actions. We introduce non-stationarity within each episode by increasingly changing the adolescent patient properties to an alternative patient over the episode. This significantly affects the utility of the initial target policy necessitating deferral as the patient properties change over the course of the day. The non-stationary clinician/behavior policy is estimated using Q-learning. We use an epsilon-greedy version of such a policy that is further made to degrade over time by increasing stochastic. The target policy resembles the clinician policy except is changed to take random actions in the time-window $35(= 8\text{hrs}) \leq t \leq 50(= 13\text{hrs})$. This is the desired region of pre-emptive deferral. Discretization of Glucose levels is provided in Table 4 and discretization of interventions is summarized in Table 5.

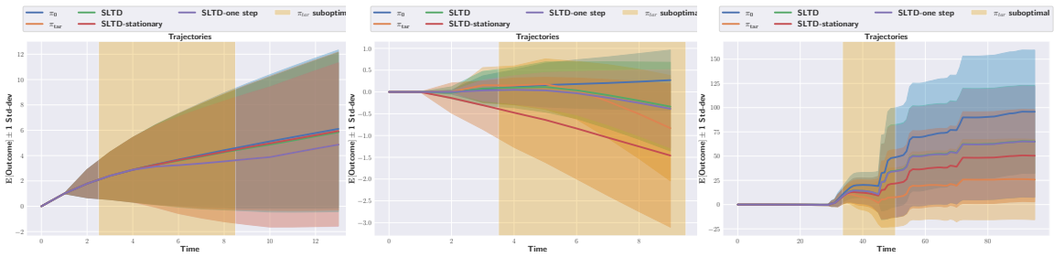
Blood Glucose (mg/dL) - BG	Discrete state
$50 < \text{BG} \leq 70$	0
$70 < \text{BG} \leq 90$	1
$90 < \text{BG} \leq 110$	2
$110 < \text{BG} \leq 180$	3
$180 < \text{BG} \leq 300$	5
otherwise	6

Table 4: Discretization of Blood Glucose for the Diabetes simulator

Incorporating non-stationarity into the simulator: We use the "Navigator" sensor to generate blood-glucose measurements and the "Insulet" pump to simulate interventions. For each episode, non-stationarity is induced by modifying the patient configurations over a period of 24 hours. This result in different dynamics over the course of the day. These configurations modify insulin sensitivity, glucose absorption and the insulin action on glucose production among other parameters. For each episode, two random adolescent patients are sampled (say 'a', and 'b'), over every minute the patient parameters are then sampled as a convex combination of patient 'a' and patient 'b' where, as we progress in time, the convex combination increasingly shifts from 0 to 1 thus changing patient parameters. Over the episode, the patient parameters increasingly look like that of patient 'b' instead of 'a'. The rate of change of this convex combination can be controlled and is set to $\cos(t \times \text{speed} \times 0.0005) \times 0.5 + 0.5$, where speed $= 5$ for our simulations. A similar policy was used by Chandak et al. [2020b] to induce non-stationarity. However Chandak et al. [2020b] do not induce non-stationarity within an episode, but across different episodes. Our setting is thus highly variable.

Bolus (g/min)	Insulin (U/min)	Discrete Action
0.00	0.00	0
21.00	10.58	1
21.00	5.25	2
51.00	18.19	3
71.00	17.75	4
9.00	2.25	5
9.00	5.823	6
9.00	10.09	7

Table 5: Discretization of bolus and insulin combination treatments



Synthetic data Sepsis-diabetes Diabetes

Figure 3: Sample trajectories for SLTD variants, target policy π_{tar} and clinician policy π_0. Mean improvement are comparable for discrete data but significant using SLTD for Sepsis-diabetes and Diabetes. Relative benefits of SLTD-stationary and SLTD-one-step are less compared to modeling non-stationarity particularly for Sepsis-diabetes and Diabetes data. Overall long-term uncertainty is comparable for all baselines for Synthetic data and Diabetes data but significantly worse for Sepsis-diabetes for SLTD-stationary baseline. A detailed analysis of this propagated uncertainty is provided in Figure 6.

8.2 Implementation Details

Data is generated using the behavior policy π_0 in all cases. We estimate posteriors of the non-stationary dynamics with Dirichlet priors over each state-action pair and learn non-stationary deferral policies for all datasets. Note that we also account for uncertainty over rewards by estimating posteriors via Bayesian inference, as in the case of the dynamics. In this case, for discrete rewards, Dirichlet priors are used, while for continuous rewards, a normal-gamma prior over each state-action pair is used. Supervised learning-to-defer baselines are commonly evaluated for accuracy of recommended treatment decision as opposed to outcome improvement. Hence, for a fair comparison with sequential decision-settings, we evaluate all methods using virtual roll-outs by deploying the respective deferral policies using the true dynamics. We average cumulative rewards over 1000 trajectories for each method and 100000 trajectories to estimate propagated uncertainty. For all baselines the cost of deferral is constant for each time-step when the policy defers to ensure fairness to myopic baselines. Deferral threshold τ are fixed based on the policy histograms for each dataset separately, but can be tuned more finely depending on the application. All policies are learned by averaging over 5 random seeds where each run is averaged over 5 bootstrapped samples.

8.3 Computation Infrastructure

All code is implemented using Python 3.8. Models were trained on a single Intel 8268 “Cascade Lake" CPUs using minimum 12GB of memory. Operating system: CentOS7. Code has also been reproduced on MacOS Catalina 10.15.7 (8 GB 2133 MHz LPDDR3, 2.3 GHz Dual-Core Intel Core i5). Code appendix includes Anaconda package dependencies required to reproduce the results.

8.4 Additional Results

Evaluating early vs late deferral: Sample Trajectories Figure 3 shows 100000 trajectories sampled with π_{tar}, π_0, SLTD, SLTD-stationary, and SLTD-one-step baselines over time, with the expected outcome ($\pm$ 1 std-dev) for all datasets. Vertical yellow shaded regions show the regions where the target policy takes suboptimal actions indicating the desired region of pre-emptive deferral. The mean improvement in outcome is apparent. Qualitative differences in deferral times can also be seen for all datasets. Quantification of propagated uncertainty is plotted in more detail in Figure 6.

Evaluating learned deferral policy. To qualitatively analyze our policies, we plot the stochastic policies learned using SLTD, SLTD-one step, SLTD stationary in the following. Note that our policy function is $g_{\pi_{\text{tar}}}(s,t)$ is non-stationary. To visualize, we plot the marginal probability of deferral over time and states separately.

Figure 4 demonstrates the policy histograms learned for the Sepsis-diabetes data. All baselines behave similarly and are able to find the appropriate region of deferral. However, the differences in the actual deferral probabilities, which reflect particularly differing in the probabilities of deferral observed at the state

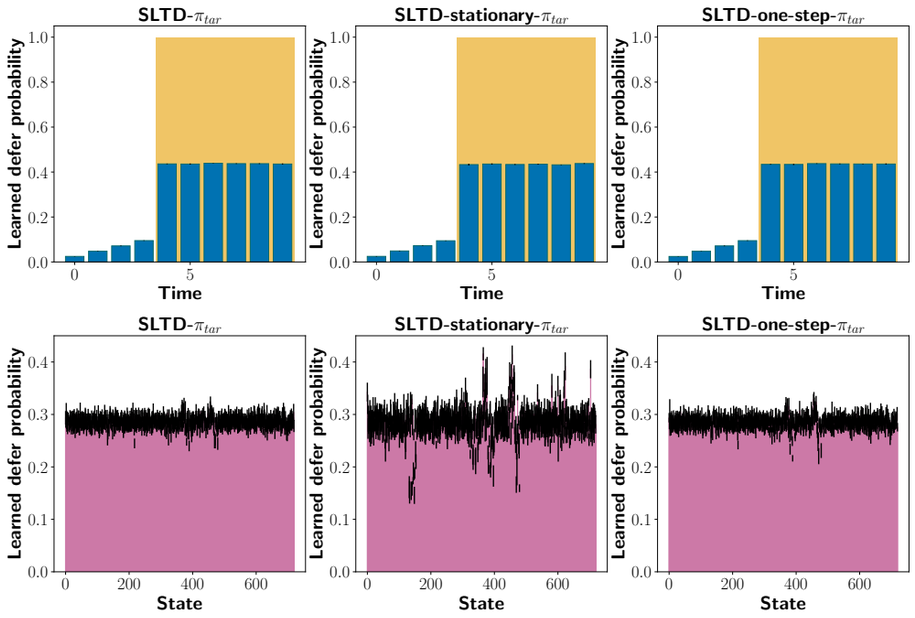


Figure 4: Learned deferral policy to Sepsis-diabetes data. Top row shows learned probabilities over time (marginalized by state) and bottom over states (marginalized by time). Shaded yellow indicates the region of pre-emptive deferral.

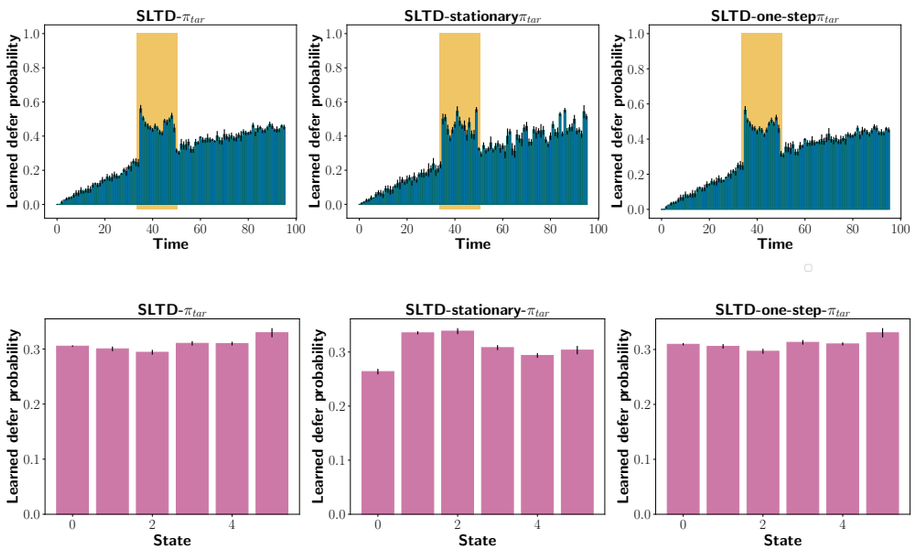


Figure 5: Learned deferral policy to Diabetes data. Top row shows learned probabilities over time (marginalized by state) and bottom over states (marginalized by time). Shaded yellow indicates the region of pre-emptive deferral.

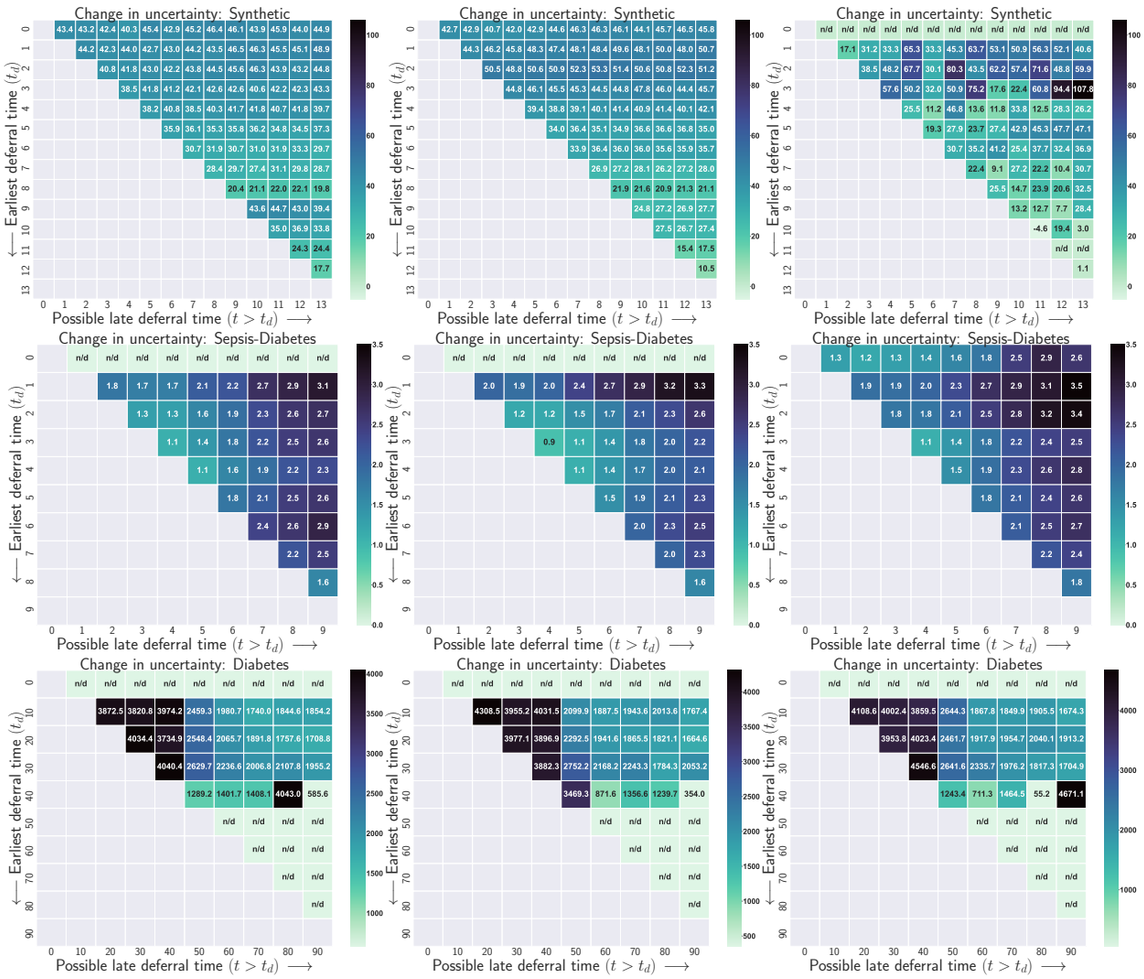
level. Although the target policy is designed to uniformly deteriorate for all states, the SLTD-stationary baseline shows high variability in deferral probabilities in states, indicating that this policy underestimates the propagated uncertainty for certain states. Similarity between SLTD-one-step and SLTD-π_{tar} suggests that introducing random actions in π_{tar} also results in a change in the immediate reward distributions providing some benefit to myopic deferral and resulting in similar deferral probabilities learned by these methods.

Figure 5 similarly demonstrates the non-stationary policy learned for the Diabetes data. All baselines learn higher likelihood of deferral in shaded regions. As in the case of Sepsis-diabetes, qualitative differences in actual learned probabilities manifest in the gap in the long-term outcomes or value as is demonstrated in Table 1. The SLTD-stationary baseline is not completely pre-emptive, with higher likelihoods of deferral toward the end of the shaded region, suggesting this policy underestimates the propagated uncertainty and therefore does not defer earlier. The probabilities of deferral by state are also significantly different as a result. Similarities between the one-step baseline and SLTD-π_{tar} indicate that immediate rewards deteriorate in the shaded yellow region as well, suggesting some benefit of myopic deferral. However, the mean outcomes do not improve as much as modeling long-term outcomes.

Uncertainty quantification. Figure 6 shows how propagated uncertainty evolves due to deferral from different SLTD variants for all datasets. Analogous to Figure 2b, this quantifies the relative increase in uncertainty when we delay deferral from the time chosen by the appropriate baseline. Top row corresponds to Synthetic data, middle rows shows the results for Sepsis-diabetes, and bottom row shows results of Diabetes data. Each column corresponds to heatmaps for SLTD, SLTD-stationary, and SLTD-one-step respectively (with π_{tar}. Each row within a heatmap corresponds to the earliest deferral time chosen by the baseline.

In each heatmap, the y-axis is the deferral time chosen by the respective deferral policy. The x-axis shows possible late deferrals (after the policy's chosen deferral time). Overall delaying deferral increases uncertainty due to the stochasticity in the system. For Synthetic data, the change in uncertainty using SLTD-stationary is slightly higher than SLTD particularly when SLTD recommends earlier deferral (top rows in the heatmaps). On the other hand, the differences are not significant when SLTD and the stationary variants recommend to defer late. On the other hand, SLTD-one-step has a significantly higher impact on variance, increasing relative uncertainty even more compared to other baselines, suggesting the one-step variant reduces uncertainty signficantly. However, managing this uncertainty by deferring according to SLTD-one-step is not reflected in improved mean outcomes.

Change in uncertainty is relatively comparable for Sepsis-diabetes and Diabetes data, for SLTD and its stationary variant. The non uniform increase in variance suggests that deferring at times *after* the region where π_{tar} takes sub-optimal action e.g. $t \geq 50$ for Diabetes does not eventually impact variance as much as delaying deferral before or within the sub-optimal/shaded region. Similar behavior is observed for the stationary and one-step variants. The relative similarity between uncertainty patterns of SLTD and the one-step variant indicates that rewards also deteriorate in the region where π_{tar} takes sub-optimal decisions, indicating that myopic deferral has some benefits. Nonetheless the improvement in mean outcomes is not similar with SLTD-one-step baseline. Note that for Diabetes data, we subsample time points to show every 10 time-steps.



(a) SLTD uncertainty (b) SLTD-stationary uncertainty (c) SLTD-one step uncertainty

Figure 6: Heatmap of increased uncertainty due to delayed deferral. The y-axis shows SLTD's earliest chosen deferral time. The x-axis shows possible later deferral times. The values in each cell are the relative increase in variance over cumulative reward if we defer *after* the baseline's first deferral time. 'n/d' implies the baseline did not recommend deferral at that time on the y-axis for any trajectory. SLTD and SLTD-stationary reduce uncertainty comparably in the system for all datasets. Change in uncertainty using SLTD-one step is higher and highly variable as long-term modeling is critical but is not accounted for by this baseline. In practice, SLTD-one step also results in worse outcomes compared SLTD being more certain of suboptimal outcomes.