Unobserved Is Not Equal to Non-existent: Using Gaussian Processes to Infer Immediate Rewards Across Contexts

Hamoon Azizsoltani, **Yeo Jin Kim**, **Markel Sanz Ausin**, **Tiffany Barnes** and **Min Chi**
North Carolina State University
{hazizso, ykim32, msanzau, tmbarnes, mchi}@ncsu.edu

Abstract

Learning optimal policies in real-world domains with delayed rewards is a major challenge in Reinforcement Learning. We address the credit assignment problem by proposing a Gaussian Process (GP)-based immediate reward approximation algorithm and evaluate its effectiveness in 4 contexts where rewards can be delayed for long trajectories. In one GridWorld game and 8 Atari games, where immediate rewards are available, our results showed that on 7 out 9 games, the proposed GP-inferred reward policy performed at least as well as the immediate reward policy and significantly outperformed the corresponding delayed reward policy. In e-learning and healthcare applications, we combined GP-inferred immediate rewards with offline Deep Q-Network (DQN) policy induction and showed that the GP-inferred reward policies outperformed the policies induced using delayed rewards in both real-world contexts.

1 Introduction

A large number of tasks in science and engineering, from robotics to game playing, tutoring systems, medical treatment design and beyond, can be characterized as sequential decision-making under uncertainty. Many interesting sequential decision-making tasks can be formulated as reinforcement learning (RL) problems [Sutton and Barto, 2018]. In an RL problem, an agent interacts with a dynamic, stochastic, and incompletely known environment, with the goal of finding a policy that optimizes some long-term *reward*. RL approaches are typically learned either online, where the agent learns while interacting with the environment; or offline, where the agent learns the policy from pre-collected data. Online RL algorithms are generally appropriate for domains where interacting with simulations and actual environments is computationally cheap and feasible. Offline RL is generally appropriate for domains such as e-learning and healthcare, where building accurate simulations or simulated students is especially challenging because both human learning and disease progression are complex, poorly understood processes. Moreover, learning RL policies while interacting with students or patients can be unethical or illegal. Therefore, we focus on offline RL approaches. Just as supervised models depend heavily on accurate *labels* for the training dataset, the effectiveness and robustness of RL approaches depend heavily on an accurate *reward* function. Applying offline RL to such domains, however, often faces two challenges related to rewards: one is delayed rewards, and the other is uncertainty.

First, delayed rewards can limit the potential of RL applications, especially when data is limited. Immediate rewards are generally more effective than delayed rewards for offline RL because it is easier to assign appropriate credit or blame when the feedback is tied to a single decision. The more we delay rewards or punishments, the harder it becomes to assign credit or blame properly. However, the most appropriate rewards in e-learning and healthcare are student learning performance and patient outcomes, which are typically unavailable until the entire trajectory is complete. This is due to the complex natures of both learning and disease progression, which make it difficult to assess students' learning or patient health states moment by moment. More importantly, many instructional or medical interventions that boost short-term performance may not be effective over the long-term.

Second, the uncertainty in real-world domains such as e-learning and healthcare often comes from incomplete or imperfect observations of underlying true reward mechanisms. Different from delayed rewards in classic mouse-in-the-maze situations where agents receive insignificant rewards along the path and a significant reward in the final goal state (the food), in e-learning and healthcare, there are immediate rewards along the way but they are often *unobservable*. Therefore, the challenge is how to infer these unobservable, immediate rewards from the delayed rewards, while taking the noise and uncertainty in the data into account.

We proposed and applied a Gaussian Processes (GP)-based approach to infer *unobservable* "immediate rewards" from the delayed rewards and then induced policies based on the inferred rewards. Much recent research focuses on the principled handling of uncertainty for modelling in environments that are dynamic, noisy, observation-costly and time-sensitive. Gaussian Processes have shown to be a robust, stable, computationally tractable and principled approach that naturally accommodates these real-world challenges [Rasmussen, 2003]. A GP is a generative model of Bayesian inference that can be used for function approximation, and it can provide a consistent and principled probabilistic framework for inferring the expected value and the point-wise variance of a hypothesis even with noisy observations [Rasmussen, 2003]. A GP is fully defined by a mean and a kernel function that defines prior function correlations, which is crucial for obtaining good posterior estimates with just a few observations. More specifically, when applying GP to inferring rewards from the delayed rewards, we assume that the inferred, unobservable rewards and the observed delayed rewards follow some prior probabilistic distributions, such as Gaussian distributions, and we incorporate information from our training data into the model using standard rules of Bayesian inference. We can then determine the rewards' corresponding posterior probabilistic distributions. By assuming a prior distribution on the value function and the parameters defining our reward functions, we avoid the trap of letting a few data points steer us away from the true parameters. Additionally, by explicitly modelling the distribution over unknown system parameters, combining GP with RL provides a principled Bayesian approach for handling uncertainty.

We have conducted a series of evaluations on our proposed GP-based immediate reward algorithm using two simulated environments and two real-world applications. In a total of 9 games including a simulated GridWorld testbed and 8 Atari games, where immediate rewards are available for comparison and online evaluation is feasible, our results showed that in 7 out 9 games, the proposed GP-inferred reward policy performed at least as well as the immediate reward policy, and significantly outperformed the corresponding delayed reward policy. In both e-learning and healthcare applications, where only delayed rewards are available, we combined GP-inferred immediate rewards with offline Deep Q-Network (DQN) and showed that in both real-world contexts, the GP-inferred reward policies outperformed the delayed reward policies.

2 Related Work

While GP has been widely and successfully applied with various RL algorithms, prior work has focused mainly on *policy induction*. For example, several works directly estimated the value function using a GP regression model [Engel *et al.*, 2003; Kuss and Rasmussen, 2004; Engel *et al.*, 2005]. In those works, the rewards were immediately available along the trajectories, and the SARSA [Sutton and Barto, 2018] algorithm was used to learn the Markov Decision Process (MDP) policy.

In most RL practice, on the other hand, reward functions were assumed to be known or manually defined based on expert knowledge, without considering the underlying true reward mechanisms and potential bias in the pre-assigned rewards. Under this circumstance, Russell [1998] presented Inverse Reinforcement Learning (IRL), which tries to solve a problem to determine the reward function that the agent optimizes by observing the environment and the agent's behaviors [Abbeel and Ng, 2004]. In his pioneering work, Ng *et al.* [1999] proposed potential-based reward shaping where the agent could be guided by the reshaped reward function with the potential difference between states and substantially reduced the learning time. Since a large set of reward functions could be found through reward shaping, Ng and Russell [2000] proposed a reward function-selecting algorithm that maximally differentiates the observed policy from other sub-optimal policies. In its extensions, Wiewiora *et al.* [2003] employed reward shaping based on both states and actions while the potential values only consider states, and Gao and Toni [2015] incorporated the potential-based reward shaping into hierarchical RL. Motivated by the psychological difference between a supplied reward and a motivated behavior, Barto [2013] and Kulkarni *et al.* [2016] developed intrinsically-motivated agents capable of exploring new behavior for their own sake. More recently, several online methods for reward shaping have been introduced; Grzes and Kudenko [2010] showed that without prior knowledge, the potential function can be learned online in parallel with the RL process, and Gimelfarb *et al.* [2018] devised a Bayesian reward shaping method that performs online updates of the weights of different hypotheses from multiple experts. However, none of these methods were evaluated in offline environments derived from real-world applications.

3 Gaussian Processes to Infer Immediate Rewards from Delayed Rewards

We apply Gaussian Processes (GP) to learn the distribution function for the expected values and the standard deviations of the immediate rewards for a historical dataset of trajectories with delayed rewards. To do this, a prior probability is given to each possible function, where higher probabilities are given to functions that we expect to observe given the delayed rewards. In context of GP, such functions are specified by their mean and covariance function (kernel). Generally speaking, there are two derivations for the GP, where the correlation between samples is represented as a covariance function or variogram. Covariance-based GP is commonly used in machine learning [Rasmussen, 2003; Nychka *et al.*, 2015; Azizsoltani and Sadeghi, 2018] and variogram-based GP is generally used in spatial statistics and Kriging [Cressie, 1992; Azizsoltani and Haldar, 2018]. We defined the prior probability as the covariance matrix $\mathbf{C_r}$ on the immediate rewards, which is equivalent to using an appropriate kernel function. Since the set of acceptable functions is uncountably infinite before observing any data, the proposed formulation will give a higher probability to the functions for which the summations of the immediate rewards are closer to the delayed rewards. The combination of this additional information and the prior will give the posterior distribution over the function.

Let $H := (s_0, a_0, r_0, s_1, a_1, r_1, \cdots)$ be a trajectory of states, actions, and rewards, and the delayed reward for the trajectory is $R = \sum_{i=0}^{n-1} \gamma^i r_i$ where n is the length of the trajectory and $\gamma \in (0, 1]$ is the reward discount factor for an episodic finite horizon problem. Furthermore, the historical data $\mathcal{D}$ consists of m trajectories with various lengths: n_1, ... n_m. Each trajectory has a single delayed reward at the final state: R_1, ... R_m. We define a reward transformation matrix $\mathbf{D} \in \mathbb{R}^{m \times l}$ where $l = \sum_{i=1}^{m} n_i$ is the number of the immediate rewards to be inferred. In $\mathbf{D}$, each row represents a trajectory, and for trajectory i, we will have n_i non-zero entries, one for each inferred reward in the trajectory. Here, $\mathbf{D}$ is the linear combiner of the delayed rewards and unknown immediate rewards as:

$$\mathbf{D}=\begin{bmatrix} \overbrace{1 \quad \gamma \quad \gamma^2 \quad \dots}^{n_1} & \overbrace{0 \qquad \dots}^{n_2} & & & 0 \\ 0 \qquad \dots \quad 0 & 1 \quad \gamma \quad \dots & 0 & \dots & 0 \\ & & & & \ddots \\ 0 \qquad \dots & & \dots & & \end{bmatrix} \tag{1}$$

By adding noise to the summation of the unobserved immediate rewards, we can represent the delayed reward as a function of immediate rewards using matrix form as $\mathbf{R}=\mathbf{D}\mathbf{r}+\varepsilon$ where $\mathbf{R} \in \mathbb{R}^m$ is the delayed reward vector of size m, $\varepsilon \in \mathbb{R}^m$ is the white noise vector, or reward error vector, of size m, and $\mathbf{r} \in \mathbb{R}^l$ is the vector of unknown immediate rewards which needs to be estimated.

We assume that the immediate rewards follow a Gaussian Process defined as $\mathbf{r} \sim \mathcal{N}(\mu_\mathbf{r}, \mathbf{C}_\mathbf{r})$ where $\mu_\mathbf{r}$ is the a priori mean and $\mathbf{C}_\mathbf{r}$ is the a priori covariance which can be any positive-definite kernel function. The a priori mean and the a priori covariance on the immediate rewards are set based on our prior knowledge about the expected value and the covariance of the immediate rewards. We further assume that the reward error vector follows an Independent, Identically Distributed Gaussian distribution with zero mean and variance $\sigma^2_\mathbf{R}$. Therefore, it is defined as $\varepsilon \sim \mathcal{N}\left(0, \sigma^2_\mathbf{R}\mathbf{I}\right)$.

Considering the column vector $\mathbf{r}$ as a vector of random variables, the expected value of the delayed rewards is $\mathbb{E}[\mathbf{R}]=\mathbf{D}\mathbb{E}[\mathbf{r}]+\mathbb{E}[\varepsilon]=\mathbf{D}\mu_\mathbf{r}$, and the covariance matrix of the immediate rewards, $\mathbb{C}_{\mathbf{rr}}$, is equal to $\mathbb{C}_{\mathbf{rr}}=\mathbf{C}_\mathbf{r}$.

Since $\mathbf{r}$ and ε are independent, by substitution of expected value of the delayed rewards into the definition of the covariance matrix for the delayed reward, $\mathbb{C}_{\mathbf{RR}}$, is calculated as:

$$\begin{aligned}\mathbb{C}_{\mathbf{RR}} &= \mathbb{E}\left[(\mathbf{D}\mathbf{r}+\varepsilon-\mathbf{D}\mu_\mathbf{r})(\mathbf{D}\mathbf{r}+\varepsilon-\mathbf{D}\mu_\mathbf{r})^\mathrm{T}\right] \\ &= \mathbf{D}\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}+\sigma^2_\mathbf{R}\mathbf{I}.\end{aligned} \tag{2}$$

Since $\mathbb{E}[\mathbf{r}\varepsilon]=0$ and $\mathbb{E}[\mu_\mathbf{r}\varepsilon]=0$, the cross-covariance between the delayed reward vector and the vector of unknown immediate rewards is calculated as:

$$\mathbb{C}_{\mathbf{rR}}=\mathbb{E}\left[(\mathbf{r}-\mu_\mathbf{r})(\mathbf{D}\mathbf{r}+\varepsilon-\mathbf{D}\mu_\mathbf{r})^\mathrm{T}\right]=\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}. \tag{3}$$

Following the theorem of conditional probability density functions for a multivariate Gaussian, the conditional distribution of immediate rewards given delayed rewards is proposed as $(\mathbf{r}|\mathbf{R}) \sim \mathcal{N}(\mathbb{E}[\mathbf{r}|\mathbf{R}], \mathbb{C}[\mathbf{r}|\mathbf{R}])$ where the posterior mean and posterior covariance of inferred immediate rewards given delayed rewards is defined as:

$$\begin{aligned}\mathbb{E}[\mathbf{r}|\mathbf{R}] &= \mathbb{E}[\mathbf{r}]+\mathbb{C}_{\mathbf{rR}}\mathbb{C}^{-1}_{\mathbf{RR}}(\mathbf{R}-\mathbb{E}[\mathbf{R}]) \\ &= \mu_\mathbf{r}+\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}\left(\mathbf{D}\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}+\sigma^2_\mathbf{R}\right)^{-1}(\mathbf{R}-\mathbf{D}\mu_\mathbf{r})\end{aligned} \tag{4}$$

$$\begin{aligned}\mathbb{C}[\mathbf{r}|\mathbf{R}] &= \mathbb{C}_{\mathbf{rr}}-\mathbb{C}_{\mathbf{rR}}\mathbb{C}^{-1}_{\mathbf{RR}}\mathbb{C}_{\mathbf{Rr}} \\ &= \mathbf{C}_\mathbf{r}-\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}\left(\mathbf{D}\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}+\sigma^2_\mathbf{R}\right)^{-1}\mathbf{D}\mathbf{C}^\mathrm{T}_\mathbf{r}.\end{aligned} \tag{5}$$

The components of Eqs. 4 and 5 can be viewed in the context of the general GP. The term $\mu_\mathbf{r}$ is an a priori estimate of the immediate rewards, $\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}$ is the cross-correlation term, $\left(\mathbf{D}\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}+\sigma^2_\mathbf{r}\mathbf{I}\right)^{-1}$ is the relative quality of the immediate and delayed rewards, $\mathbf{R}-\mathbf{D}\mu_\mathbf{r}$ is the reward prediction error, $\mathbf{C}_\mathbf{r}$ is the a priori covariance of the immediate rewards, and the rest of Eq. 5 is the reduction of covariance based on observations of the data. In short, the proposed GP can estimate the *mean* and pointwise *variance* of the inferred immediate rewards given the observed delayed reward.

Numerical calculation and computational complexity. The Cholesky factorization is used to decompose a symmetric positive definite matrix such as $\mathbb{C}_{\mathbf{RR}}$ into the product of a lower triangular matrix, $\mathcal{L}$, and its conjugate transpose, $\mathcal{L}^\mathrm{T}$, where $\mathcal{L}\mathcal{L}^\mathrm{T}=\mathbf{D}\mathbf{C}_\mathbf{r}\mathbf{D}^\mathrm{T}+\sigma^2_\mathbf{r}\mathbf{I}$. The lower triangular equations $\boldsymbol{\beta}=\mathcal{L}\backslash(\mathbf{R}-\mathbf{D}\mu_\mathbf{r})$ are solved by the forward-substitution while the upper triangular equations $\boldsymbol{\alpha}=\mathcal{L}^\mathrm{T}\backslash\boldsymbol{\beta}$ are solved by the back-substitution. By introducing the intermediary variables $\overline{\boldsymbol{k}}=\mathbf{D}\mathbf{C}^\mathrm{T}_\mathbf{r}$ and $\mathbf{v}=\mathcal{L}\backslash\overline{\boldsymbol{k}}$, we can calculate the posterior mean and posterior covariance of inferred immediate rewards introduced in Eqs. 4 and 5 as $\mathbb{E}[\mathbf{r}|\mathbf{R}]=\mu_\mathbf{r}+\overline{\boldsymbol{k}}^\mathrm{T}\boldsymbol{\alpha}$ and $\mathbb{C}[\mathbf{r}|\mathbf{R}]=\mathbf{C}_\mathbf{r}-\mathbf{v}^\mathrm{T}\mathbf{v}$. The computational complexity of the Cholesky factorization is $O\left(m^3\right)$ where m is the number of trajectories. The computational complexity of forward-substitution and back-substitution algorithms are both $O\left(m^2\right)$. One potential caveat of the proposed algorithm is the asymptotic computational complexity of $O\left(m^3\right)$. However, it may not be a major concern for offline RL.

4 Two Simulated Environments

Our simulated environments include a grid world game and 8 Atari games, and for each game, we collected trajectories with immediate rewards; for each trajectory, we summed up all immediate rewards to be the final delayed rewards; and then we applied our GP-based algorithm to infer rewards using the delayed reward trajectories. Finally, we applied different offline RL approaches to induce policies using the immediate (Imm), the inferred (Inf), and the delayed (Del) rewards.

4.1 Experimental Settings

GridWorld Game

Figure 1 (Left) shows our simple benchmark: a GridWorld testbed. Here the agent (small circle) starts from the start state (bottom right corner), explores the 2D space and eventually finishes at the end state (upper left corner). Several walls are designed in the GridWorld, and the agent bounces back to its original state after hitting the walls. The agent can take 3 actions (UP, Down, Left) and collects +1, 0, or -1 rewards. The goal is to maximize the rewards. We explored two types of reward functions: 1) a state-based, $R(s)$ and 2) a state-action-state based, $R(s, a, s')$.

Atari Games

The OpenAI Gym toolkit [Brockman *et al.*, 2016] was used to simulate eight Atari games. We first randomly played the games and collected 200,000 steps per game, where each step consists of a (s, a, r) tuple. To induce policies, we used the offline Double-DQN algorithm [Van Hasselt *et al.*, 2016] with prioritized experience replay [Schaul *et al.*, 2015]. We repeated this process for immediate, inferred, and delayed rewards. The OpenAI Baselines was used for training the

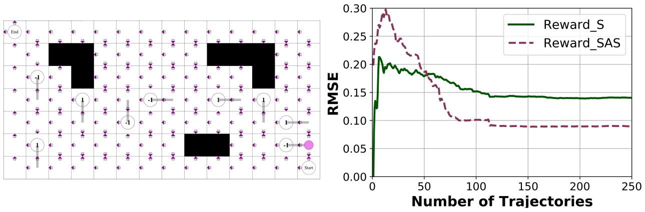

Figure 1: Left: GridWorld. Right: RMSE for GridWorld.

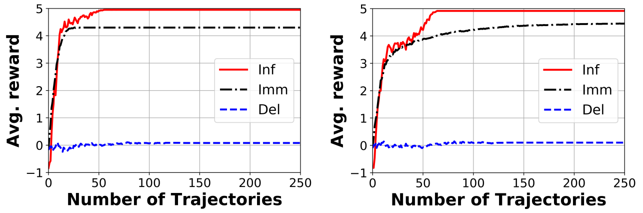

Figure 2: Avg. reward on GridWorld. Left:$R(s)$. Right:$R(s, a, s^{'})$.

models. The Double-DQN model architecture and training procedure followed those by Mnih [Mnih *et al.*, 2015], e.g. the input is a 84x84 pixel image, and the last 4 frames are stacked.The main difference was that we did not clip the positive and negative rewards to 1 and -1, respectively; we kept the rewards provided by the environment to capture more accurate delayed rewards, to better estimate inferred rewards.

4.2 Game Simulation Results

GridWorld Game

We used three evaluation metrics: first Root Mean Square Error (RMSE) was used to estimate how close the inferred immediate rewards were to the ground truth immediate rewards. Figure 1 (Right) shows the RMSE decreases as the number of trajectories increases for both the state reward function (Reward_S) and the state-action-next state reward function (Reward_SAS). RMSE stabilizes after 100 trajectories. The asymptotic RMSE constants 0.14 RMSE for Reward_S and 0.08 for Reward_SAS are due to the presence of some states with only one deterministic action, where there is not enough evidence to distribute the reward accordingly. The second metric, the average collected reward, was used with online policy induction using Watkin's $Q(\lambda = 0.7)$ [Sutton and Barto, 2018] and online evaluation for the Imm, Inf, and Del policies. Figure 2 shows the comparison of using greedy Q-learning as policy induction with three policies for $R(s)$ (left) and $R(s, a, s')$ (right). Here we report the average performance over 200 randomly initiated rounds using the sample-average technique. For each round, we explored the performance of the induced policy by increasing the number of trajectories from 1 up to 250. The results show that the Imm and Inf policies performed much better than the Del policies. Indeed, by considering the noise and uncertainty in the data, the Inf policies performed even better than the Imm policy. Finally, GP temporal difference learning [Engel *et al.*, 2003] was used for offline policy induction and its online evaluation is shown in figure 4, along with the Atari games.

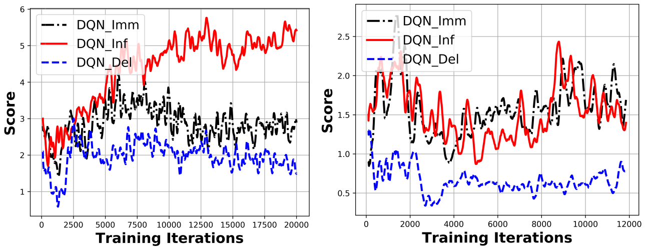

Figure 3: Evaluation during training. Left: Amidar. Right: Berzerk.

Atari Games

Given space limits, we will only report the online evaluations. The evaluation on each of the Atari games was done by training a neural network for 25 iterations, playing the game once, and storing the reward obtained on that episode. We repeated this for 20,000 episodes (which means we trained the algorithm for around 80 epochs). Figure 3 shows the rewards collected per episode during the training process on two different Atari games. Figure 4 shows the normalized performance of using each type of reward on several Atari games after the algorithm has converged. The black horizontal line shows the performance obtained when training DQN using immediate rewards and is indicated as 100%. This allows us to compare the three types of rewards, after normalizing the performance for each game. Overall, the DQN_Inf performed much better than DQN_Del across all games and the exceptions are Alien and ChopperCommand. Note that on 5 out of 8 games, DQN_Inf performed close to the DQN using the immediate rewards and on the first two games, DQN_Inf performed even better than the policy using immediate rewards. It is likely that DQN_Inf performed poorly on ChopperCommand because 200,000 steps of random play are not enough data to learn the optimal policy.

5 Real-World Environments

We describe our experiments and results in two real-world domains: e-learning and healthcare.

5.1 Experimental Settings

E-learning

Grounded in artificial intelligence concepts and cognitive theory, Intelligent Tutoring Systems (ITSs) are computer systems that provide students with customized and individualized learning experiences by guiding them through each step of a problem solution and providing adaptive hints and feedback as needed. We used a logic ITS where students solve problems by applying logic rules to derive new logical statements. Each problem in the tutor can be presented as a worked example (WE) or problem solving (PS). In WE, the student observes how the tutor solves a problem; in PS, the student solves the problem. The logic tutor problems are organized into six strictly ordered levels; in each level students complete 3–4 problems. The last problem on each level is used to calculate a level score, as students must solve this last level problem (PS) without tutor help. When inducing RL policies, our rewards were based on the difference between the student's current and prior level scores.

Our training corpus contains 786 student-tutor interaction logs collected over five years. On average, students spent about 2 hours on the tutor and each trajectory contains more than 19 decisions. Because both *WE* and *PS* are always considered to be *reasonable* educational interventions in this learning context, we refer to such a policy as a *random yet reasonable* policy or *random*. From student-system interaction logs, we extracted a total of 142 state features including: 10 **Autonomy** features describing the amount of work done by the student; 29 **Temporal** features such as the average time per step or the total time so far; 35 **Problem Solving** features such as the difficulty of the current problem; 57 **Performance** features such as the percent correct; and 11 **Hints** related features such as the total number of hints requested.

We induced policies by applying DQN [Mnih *et al.*, 2015] with two Long Short-Term Memory (LSTM) layers [Hochreiter and Schmidhuber, 1997] of 100 LSTM units each, followed by the fully-connected output layer and ReLU as the activation function. To evaluate our GP-based algorithm, DQN is applied to induce DQN policies for both the delayed rewards (DQN-Del) and inferred rewards (DQN-Inf).

Healthcare

Our ultimate goal is to learn an "optimal" treatment policy for septic shock patients. Our electronic health records (EHR) were collected from Christiana Care Health System from July, 2013 to December, 2015 consisting of $210,289$ visits and $9,029,493$ events. By combining the International Classification of Diseases, Ninth Revision (ICD-9) and clinician rules, we sampled $2,964$ positive septic shock trajectories and $2,964$ negative trajectories (no shock), keeping the same distribution of age, gender, race, and the length of hospital stay as in the original dataset. Twenty-two sepsis-related features including vital signs, lab results, and oxygen controls were extracted. The average rate of missing data was 83.2%, and we imputed the missing values using the expert imputation rules as described in [Kim and Chi, 2018], which forward-fills 8 hours for vital signs and oxygen control and 24 hours for lab results, combined with mean imputation for the remaining missing values. After cutting off the positive septic shock trajectories after the onset of septic shock, the data were aggregated with 1-hour time windows, and we extracted mean, min, and max value in the time window. The final number of features was 66, and the final dataset includes $5,928$ visits and $210,494$ aggregated events where the average length of trajectories is about 34 and can go up to 507.

To approximate patients' hidden states and state-action (Q) values, we leveraged 4 layers of fully connected neural networks with 128 hidden units for each layer. We extracted three types of actions: antibiotic, vasopressor, and oxygen control, and thus our action space is 2^3 because these actions were often combined during the aggregation process. To define rewards, five septic stages were defined based on the clinical rules, and the delayed reward for each stage was set as follows: Infection (±1), Inflammation (±50), OrganFailure (±100), Shock (±1000), and Death (±10000). The designated negative reward was given when a patient enters into the corresponding stage, and its positive reward was given back when the patient recovers from the stage. In this way, an optimal policy should keep patients from getting negative rewards and help them stay in non-negative states.

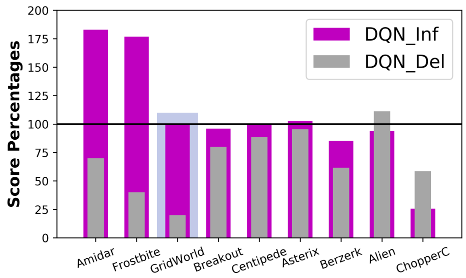

Figure 4: Performance of Atari games and GridWorld. The black line shows the immediate reward, normalized to show 100%.

We compared two types of policies for the EHR data. The physician policies were induced using SARSA, which followed the recorded physician actions, and the agent policies were induced using the dueling double DQN with prioritized experience replay [Raghu *et al.*, 2017]. The data was split into 80% for training and 20% for test. To evaluate the inferred rewards, we compared the physician policies with delayed and inferred rewards: Phys_Del and Phys_Inf respectively with the two DQN policies with delayed and inferred rewards: DQN_Del and DQN_Inf respectively.

5.2 Real World Results

E-learning

The effectiveness of DQN-Del and DQN-Inf were evaluated theoretically using Expected Cumulative Reward (ECR) and empirically through two controlled experiments. Figure 5 (Left) shows that DQN-Inf converged faster and had higher ECR than DQN-Del.

Two empirical studies were performed to evaluate the effectiveness of DQN-Del in Spring 2018 and DQN-Inf in Fall 2018, respectively. They were conducted in an undergraduate Discrete Mathematics course as a regular homework assignment. In each study, the effectiveness of the corresponding RL-induced policy was compared against the *Random* policy. The students were randomly assigned into the two conditions while balancing their incoming competence. Overall, the results from both experiments showed no significant difference between the DQN-Del and Random in Spring 2018 and between the DQN-Inf and Random in Fall 2018 on any measures of learning performance. Therefore, despite the fact that our theoretical results showed that the ECRs of the two RL induced policy look very reasonable, our empirical results showed they are no better than the Random policy.

There are two potential explanations for such findings. First, while random policies are normally bad in many RL tasks, in the context of WE vs. PS, our random policies can be pretty strong baselines. Indeed, a lot of learning literature suggests that the best instructional intervention is to alternate WE and PS [Renkl *et al.*, 2002; Schwonke *et al.*, 2009]. Second, there may be an **aptitude-treatment interaction (ATI)** effect [Cronbach and Snow, 1977; Snow, 1991], where certain students achieve similar learning performance regardless of the induced policies whereas other students' learn-

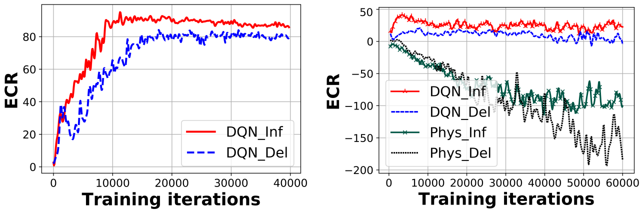

Figure 5: ECR value during the training process of real environments. Left: E-learning. Right: Septic Treatment Policies.

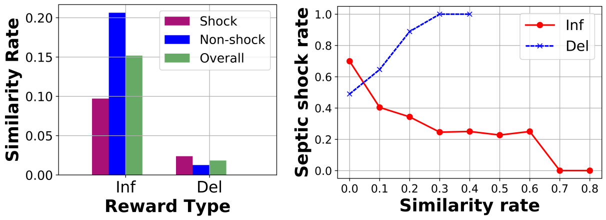

Figure 6: The qualitative analysis of the septic treatment policies. Left: Shock vs. Non-shock group. Right: Correlation between the septic shock rate and the policy similarity.

ing is highly dependant on the effectiveness of the policies. Thus, we divided the students into High vs. Low based on their incoming competence and investigated the ATI effect. While no ATI effect was found between DQN-Del and Random for Spring 2018, a significant ATI effect was found between DQN-Inf and Random in Fall 2018. Based on their incoming competence, the students were split into four groups: DRL-Inf-High ($n = 20$), DRL-Inf-Low ($n = 21$), Random-High ($n = 22$), Random-Low($n = 21$). As expected, no difference was found between the two High groups or between the two Low groups on the incoming competence. A two-way ANCOVA test on the post-training performance, using Condition {*DRL-Inf*, *Random*} and Competency {*High*, *Low*} as factors and incoming competence as a covariate, showed a significant interaction effect, $F(1, 79) = 4.687$, $p = 0.033$. More specifically, while no difference was found between the two Low Competency groups, a significant difference was found between the two High Competency groups: $F(1, 39) = 5.513, p = 0.024$ in that DRL-Inf-High scored significant higher than the Random-High group.

Healthcare

The four septic treatment policies were evaluated using the average ECR for each policy every 200 training iterations until reaching $60,000$ iterations, where the agent sampled 32 states every iteration with the prioritized importance weights on temporal difference errors. Figure 5 (Right) shows the ECRs of the four policies as the training iteration increases. Since the patients' states could get worse as their diseases progressed, the average rewards at a state of the test set were negative: Del (-18.8) and Inf (-17.7), and the average ECRs before training were Del (-267.6) and Inf (-134.3). Note that the average reward of Inf can be dissimilar from the one of Del due to the noise factor, and their ECRs can also significantly diverge because of the reward distribution and the discount factor, especially with such a large EHR dataset. In this sense, the ECR metric is unreliable when comparing policies with different rewards, even though ECR can be supplementary when comparing policies that use the same rewards. Thus, we used ECR values to evaluate the RL methods (Phys & DQN) but only considered the convergence rate of ECR to evaluate the reward types (Del & Inf). In both DQN and Phys, Inf converged faster than Del, and DQN_Inf (23.6) and Phys_Inf (-96.3) achieved higher ECRs than DQN_Del (4.5) and Phys_Del (-157), respectively.

To validate the impact of the agent policies with Inf and Del for septic treatment, we analyzed two qualitative aspects of the agents' policies: 1) whether the policy can better prevent the septic shock progress, and 2) whether the septic shock rate monotonically decreases as the treatments are more similar to the agent policy. Here, the similarity rate metric indicates how close the visit-level treatment in the dataset is to the agent policy [0:different, 1:same]. Since a hospital visit consists of multiple temporal events, the similarity rate was averaged by trajectory. First, Figure 6 (Left) contrasts the policy similarities to Inf and Del between the shock and the non-shock group. Overall, 15.2% of trajectories were similar to Inf and 1.8% of them were to Del, which means Inf better learned the physician policy than Del. When using Inf, the non-shock group (20.6%) followed Inf more than the shock group (9.7%), whereas when using Del, the non-shock group (1.3%) followed Del less than the shock group (2.4%). Thus, the more treatments followed Inf, the more effectively they could reduce the septic shock rate. Next, shown in Figure 6 (Right), the shock rate of Inf almost monotonically decreases as the similarity rate to Inf increases, while Del's fluctuates. This indicates that there is a negative correlation between the policy similarity and the shock rate when using Inf but no correlation when using Del. In sum, Inf more effectively learned the optimal treatment policy to prevent septic shock than Del.

6 Conclusions

We proposed a GP-based estimator to infer the posterior mean and variance of the immediate yet unobservable rewards from delayed rewards. Our evaluations using the GridWorld testbed showed that the proposed framework is capable of approximating the inferred state-based rewards as well as state-action-state based rewards. Moreover, we demonstrated the effectiveness of the proposed framework by inducing policies from inferred rewards that performed as well as those directly using the immediate rewards and a similar number of trajectories. Furthermore, we evaluated our proposed algorithm in 8 Atari games and two real-world applications, e-learning and healthcare, and the empirical experimental results demonstrated that the benefits of the proposed algorithm were still valid for these tasks. These results confirm that our algorithm can be used in domains where online interaction with the environment is prohibited or impossible, the collected data is noisy and only the delayed rewards are available at the end of each trajectory.

Acknowledgements

This research was supported by NSF grants #1522107, #1651909, #1660878 and #1726550.

References

[Abbeel and Ng, 2004] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[Azizsoltani and Haldar, 2018] Hamoon Azizsoltani and Achintya Haldar. Reliability analysis of lead-free solders in electronic packaging using a novel surrogate model and kriging concept. *Journal of Electronic Packaging*, 140(4):041003, 2018.

[Azizsoltani and Sadeghi, 2018] Hamoon Azizsoltani and Elham Sadeghi. Adaptive sequential strategy for risk estimation of engineering systems using gaussian process regression active learning. *Engineering Applications of Artificial Intelligence*, 74:146–165, 2018.

[Barto, 2013] Andrew G Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pages 17–47. Springer, 2013.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[Cressie, 1992] Noel Cressie. Statistics for spatial data. *Terra Nova*, 4(5):613–617, 1992.

[Cronbach and Snow, 1977] Lee J Cronbach and Richard E Snow. *Aptitudes and instructional methods: A handbook for research on interactions.* Irvington, 1977.

[Engel *et al.*, 2003] Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *ICML*, 2003.

[Engel *et al.*, 2005] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *ICML*, pages 201–208. ACM, 2005.

[Gao and Toni, 2015] Yang Gao and Francesca Toni. Potential based reward shaping for hierarchical reinforcement learning. In *IJCAI*, 2015.

[Gimelfarb *et al.*, 2018] Michael Gimelfarb, Scott Sanner, and Chi-Guhn Lee. Reinforcement learning with multiple experts: A bayesian model combination approach. In *NeurIPS*, 2018.

[Grześ and Kudenko, 2010] Marek Grześ and Daniel Kudenko. Online learning of shaping rewards in reinforcement learning. *Neural Networks*, 23(4):541–550, 2010.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Kim and Chi, 2018] Yeo Jin Kim and Min Chi. Temporal belief memory: Imputing missing data during RNN training. In *IJCAI*, 2018.

[Kulkarni *et al.*, 2016] Tejas Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *NeurIPS*, 2016.

[Kuss and Rasmussen, 2004] Malte Kuss and Carl E Rasmussen. Gaussian processes in reinforcement learning. In *NeurIPS*, 2004.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Ng *et al.*, 1999] Andrew Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, 1999.

[Ng *et al.*, 2000] Andrew Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

[Nychka *et al.*, 2015] Douglas Nychka, Soutir Bandyopadhyay, Dorit Hammerling, Finn Lindgren, and Stephan Sain. A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599, 2015.

[Raghu *et al.*, 2017] Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.

[Rasmussen, 2003] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[Renkl *et al.*, 2002] Alexander Renkl, Robert K Atkinson, Uwe H Maier, and Richard Staley. From example study to problem solving: Smooth transitions help learning. *The Journal of Experimental Education*, 70(4):293–315, 2002.

[Russell, 1998] Stuart J Russell. Learning agents for uncertain environments. In *COLT*, pages 101–103, 1998.

[Schaul *et al.*, 2015] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[Schwonke *et al.*, 2009] Rolf Schwonke, Alexander Renkl, Carmen Krieg, Jörg Wittwer, Vincent Aleven, and Ron Salden. The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior*, 25(2):258–266, 2009.

[Snow, 1991] Richard E Snow. Aptitude-treatment interaction as a framework for research on individual differences in psychotherapy. *Journal of consulting and clinical psychology*, 59(2):205, 1991.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Van Hasselt *et al.*, 2016] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

[Wiewiora *et al.*, 2003] Eric Wiewiora, Garrison W Cottrell, and Charles Elkan. Principled methods for advising reinforcement learning agents. In *ICML*, 2003.