# TIME AFTER TIME: SCALABLE EFFECT ESTIMATION FOR INTERVENTIONS ON WHEN AND WHAT TO DO

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Decision support in fields such as healthcare and finance requires reasoning about treatment timing. Artificial Intelligence holds great potential for supporting such decisions by estimating the causal effect of policies such as medication regimens, or resource allocation schedules. However, existing methods for effect estimation are limited in their ability to handle *irregular time*. While treatments and observations in data are often irregularly spaced across the timeline, existing techniques either discretize time, do not scale gracefully to large models, or disregard the effect of treatment time.

We present a solution for effect estimation of sequential treatment times called Earliest Disagreement Q-Evaluation (EDQ). The method is based on Dynamic Programming and is compatible with flexible sequence models, such as transformers. It provides accurate estimates under the assumptions of ignorability, overlap, and no-instantaneous effects. We validate the approach through experiments on a survival time prediction task.

## 1 INTRODUCTION

Sequential decision-making is common in fields like healthcare, finance, and beyond. In hospitals, medical professionals administer treatments based on the evolving observations of a patient's condition; in financial markets, traders execute orders based on sequential information flows. Algorithmic decision support systems can optimize these processes by evaluating different *policies* with respect to their expected outcomes. To achieve this, these systems need to answer *causal* questions.

Consider the scenario of a doctor and patient forming a preventative treatment strategy for Atherosclerotic Cardiovascular Disease (ASCVD). The American Heart Association recommends calculating the 10-year risk of developing ASCVD and starting preventative treatments for patients with a high predicted risk. At first glance, this practice may seem straightforward, but other considerations may come into play. For instance, some patients may prefer to delay starting treatment because of expected side effects; in other cases alternative treatments may be considered due to comorbidities. After each decision point, an important question is when to schedule the next checkup to re-evaluate the treatments. Estimating the difference in expected outcomes between various policies one could follow is a causal effect estimation question. This question involves several future treatment decisions taken at varying time points, hence it is a sequential decision-making problem. This type of data, where observations and treatments are given across time, appears in many applications. For instance, in intensive care units and in other domains like finance and social networks (Chen et al., 2021a; Upadhyay et al., 2018).

Formally, as we describe in section 2, the problem falls within the framework of off-policy evaluation (Fu et al., 2021; Uehara et al., 2022). A defining feature is that timings of observations and treatments are irregularly spaced, and are governed by a stochastic point process with intensity $\lambda$, whereas the *type* of the treatments at specific times are specified as the marks of this process, governed by a distribution $\pi$. This is in contrast with many applications in domains such as robotics and control, where only $\pi$ is considered. These times must be accounted for in algorithmic solutions, as they have a large effect on the outcome. *Estimating the effect of intervening in treatment timing* is a crucial part of evaluating sequential policies. With irregular times, frameworks for sequential decision-making that discretize time are problematic, as the discretization can be inaccurate or inefficient and requires choosing appropriate time scales for each dataset. Existing methods for continuous-time causal

inference do not scale gracefully, since they solve complex estimation problems such as integrating importance weights across time (Røysland, 2011). Those that do scale to high capacity models and large datasets do not handle dynamic policies (i.e., policies that take past states into account) and are implemented with differential equation solvers (Seedat et al., 2022), limiting the choices of architecture.

In this work, we give two methods for off-policy evaluation with irregularly sampled data. Our contributions are as follows:

- We define off-policy evaluation with *decision point processes* and develop Earliest Disagreement Q-Evaluation (EDQ), a model-free solution to the problem. While other methods are intractable in high dimensions or are limited to static treatments, EDQ eliminates these restrictions. EDQ is based on direct regressions and dynamic programming, which makes it easily applicable to flexible architectures including sequence models such as transformers.
- In Theorem 1, we prove that EDQ is an empirical estimator of the correct policy value. The estimator produces an accurate causal effect under assumptions on causal validity based on Røysland (2011); Røysland et al. (2022).
- With an experimental demonstration on a time-to-failure prediction task for which we implement a transformer based solution, we validate the efficacy of EDQ, where baselines that discretize time are suboptimal.

We define the effect estimation problem in section 2 using the formalism of temporal point processes and develop the solution in section 3. section 4 discusses related work to better frame our solution before validating it experimentally in section 5.

## 2 Off-Policy Evaluation with Decision Point Processes

Consider a decision process defined by a marked point process $P$ (Andersen et al., 2012; Snyder and Miller, 2012) over observations $X \in \mathcal{X}$, treatments $A \in \mathcal{A}$ and outcome $Y \in \mathbb{R}$. For convenience, we let $Y = \sum_k Y_k$, where $k$ is an index of observed outcomes along the trajectory. Though the methods extend to other outcome functions like discounting future outcomes.[1]

**Marked point processes.** Informally, a marked point process defines a distribution over event times, along with distributions over *marks*, or details of the events at each time (i.e. treatment times and which treatment was given at each time). Events occur in the time interval $[0, T]$. The events are defined as trajectories $\mathcal{H}_t^l = \{(t_k, \mathbf{l}_k) : t_k \leq t\}$ where $l$ is an action $a$, state $x$, or outcome $y$. These event times are also called jumps. The collection of all jumps and marks on a trajectory is $\mathcal{H} = \mathcal{H}^a \cup \mathcal{H}^x \cup \mathcal{H}^y$. For $l \in \{a, x, y\}$, the event times are tracked by counting process $N^l(t)$, which equals the number of events by time $t$. Define $dN^l(t)$ such that $dN^l(t) = 1$ if $N^l$ jumps at time $t$. We assume that intensity functions for processes exist and are given by $\lambda^l(t|\mathcal{H}_t) = \mathbb{E}[dN^l(t)|\mathcal{H}_t]$. We assume the process can depend on its own history, i.e., the filtration is the $\sigma$-algebra generated by the random variables $N^l(t)$ and their marks, and that $\lim_{n\to\infty} t_n = \infty$, i.e., that the number of events for each trajectory is countable (Aalen et al., 2008; Jacobsen and Gani, 2006).

### 2.1 Problem Definition

We follow the notation of Upadhyay et al. (2018) from the RL literature. We begin with the data generating process and then summarize our goal of inferring causal effects of intervening on a policy. This involves off-policy evaluation under a distribution $P$, while observing samples from $P_{\text{obs}}$.

**Definition 1.** *A marked decision point process $P$ is a marked point process with observed components $N^l$ for $l \in \{x, y, a\}$ that have corresponding intensity functions $\lambda^l$, and mark spaces $\mathcal{X}, \mathbb{R}, \mathcal{A}$, and a multivariate unobserved process with intensity $\lambda^u$. Realizations of the observed process are given by trajectories $\mathcal{H} = \{(t_0, \mathbf{z}_0), (t_1, \mathbf{l}_1), \ldots, (t_n, \mathbf{l}_n)\}$, where $t_k$ is the jump time and $\mathbf{l}_k \in \{\mathcal{X} \cup \mathcal{A} \cup \mathbb{R}\}$ is the mark, and we denote $\mathcal{H}_t := \{(t_k, \mathbf{l}_k) : t_k \leq t\}$. $\mathcal{H}^a \subseteq \mathcal{H}$ is the subset of events that correspond to*

[1] Note that we assume the number of rewards in the segment $[0, T]$ is finite. Furthermore, in practice, discounting summed outcomes might be necessary for convergence (Sutton, 2018).

*jumps in $N_a$, and likewise for $x, y$. The intensity function and mark distribution $\lambda^a(t|\mathcal{H}_t), \pi(A_t|\mathcal{H}_t)$ are called the policy. Mark distributions for $X, Y$ are denoted $P_X(\mathbf{x}_t|\mathcal{H}_t), P_Y(y_t|\mathcal{H}_t)$.*

**Off-policy evaluation.** We are given a dataset of $m$ trajectories, where trajectory $\mathcal{H}_i$ has $n_i$ observations: $\{(t_{i,k}, \mathbf{l}_{i,k})\}_{i\in[m],k\in[n_i]}$. These are sampled from an observed decision process $P_{\text{obs}}$ with policy $(\lambda^a_{\text{obs}}, \pi_{\text{obs}})$. Treatment times are samples from a counting process with intensity $\lambda^a_{\text{obs}}$ and treatments at those times are sampled from $\pi_{\text{obs}}$. We wish to reason about outcomes when $(\lambda^a_{\text{obs}}, \pi_{\text{obs}})$ is replaced with a target policy $(\lambda^a, \pi)$, with other processes in $P_{\text{obs}}$ fixed. The resulting decision process is denoted $P$, and our goal is to estimate $\mathbb{E}_P[Y|\mathcal{H}_t]$ for all $\mathcal{H}_t \in \text{supp}(P)$ and $t \in [0, T]$.

**When to treat.** To simplify notation in what follows, we omit the marks $\pi(A_t|\mathcal{H}_t)$ and focus on intensities $\lambda^a(t|\mathcal{H}_t)$. That is, we explore interventions on when to treat (*should a medicine be taken weekly or monthly?*) instead of how to treat (*which medication should be taken?*). Technically, this is the more challenging and underexplored part of the problem, and the algorithmic solutions can be easily extended to incorporate interventions on $\pi$ using existing methods (Chakraborty and Murphy, 2014; Li et al., 2021). In the ASCVD prevention example, this corresponds to reasoning about questions like: "what would be the expected change in 10-year risk for patients with characteristics $\mathcal{H}_t$, if, going forward, we prescribe a daily dose of statins for patients with LDL cholesterol above 180 mg/dL, instead of the policy that has been followed in the population?".

### 2.2 Roadmap to identifiability via Local Independences

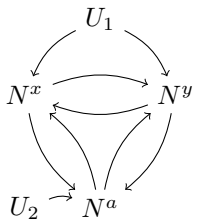



Figure 1: The assumed local independence graph for a decision point processes, where our estimand is identifiable from observed data $(N^x, N^a, N^y)$.

The goal of this section is to display results from existing work that elucidate the conditions under which the algorithm we present in section 3 estimates valid causal effects. Our assumptions to ensure identifiability of causal estimands follow Didelez (2008); Røysland (2011); Røysland et al. (2022), who study graphical models for point processes. In this setting, where the goal is to intervene on $N^a$ and estimate $E_P[Y|\mathcal{H}_t]$ under $P$ rather than $P_{\text{obs}}$, in presence of confounders $U$, Røysland (2012) show:

- A condition called *causal validity* ensures that changing treatment intensity $\lambda^a_{\text{obs}}$ for interventional treatment intensity $\lambda^a$, while changing no other intensities, changes the joint distribution from $P_{\text{obs}}$ to $P$. A graph may not be causally valid when it contains unobserved variables $U$.
- *Local independence* (Aalen, 1987; Schweder, 1970) adapts the sequential exchangeability or "ignorability" condition in discrete time processes (Hernan and Robins, 2023; Robins, 1986) to the continuous time point process setting.
- A certain set of local independences ensure causal validity even when $U$ is unobserved.

Consider the graph in Figure 1, which we study in this work. An edge means that the history of the source node affects the future of the target node (Didelez, 2008). It is possible to show that the graph satisfies the required conditions such that replacing $\lambda^a_{\text{obs}}$ with a new treatment intensity $\lambda^a$ results in sampling the interventional distribution $P$. These conditions involve local independence, which means that the intensity of a process only makes use of certain information from other processes:

**Definition 2.** *For a multivariate process $N(t) = (N^a(t), N^b(t), \ldots)$ on variables $V$ we say that $N^a$ is locally independent of $N^b$ given $N^{V\setminus b}$, or $N^b \not\to N^a|N^{V\setminus b}$, if the intensity $\lambda^a(t|\mathcal{H}_{t-}) = \lambda^a(t|\mathcal{H}^{V\setminus b}_{t-})$. A graphical local independence model $(\mathcal{P}, G)$ is a class of processes $\mathcal{P}$ on $V$ and directed graph $G = (V, E)$, such that $(b \to a) \notin E \Rightarrow N^b \not\to N^a|N^{V\setminus b}$ holds for all $P \in \mathcal{P}$.*

Røysland et al. (2022) then show that the set of local independences that ensure causal validity are those that satisfy *Eliminability*:

**Definition 3.** *If $U$ can be written as a sequence $(U_1, \ldots, U_K)$ such that for each $k$, either*

- *$(N^y, U_{>k})$ is locally independent of $U_k$ given $(N^x, N^y)$, or*
- *$N^a$ is locally independent of $U_k$ given $(N^x, N^y)$.*

*then the graph is said to satisfy Eliminability.*

Eliminability is akin to using d-separation to check the backdoor criterion in directed acyclic graphs, but accounts for fact that local independence may be asymmetric. We summarize that *causal validity* holds under the set of *local independences* that satisfy *eliminability*. Here, we stick with the graph in Figure 1, which satisfies these assumptions. In addition, we assume independence of increments to rule out any instantaneous effects, and refer to the two conditions together as **ignorability**, in accordance with existing terminology for conditions that rule out confounding.

**Assumption 1.** *Ignorability (in continuous time) is satisfied when:*

1. *the graph satisfies causal validity,*
2. *the increments of features, treatments, and outcome are mutually independent given the history, i.e., $((dN^x(t), X_t) \perp\!\!\!\perp (dN^a(t), A_t) \perp\!\!\!\perp (dN^y(t), Y_t))|\mathcal{H}_t$.*

In addition to ignorability, we require a second, standard assumption, **overlap**, for the conditional expectations we estimate to be well-defined. Recall that the interventional distribution $P$ is defined by replacing the treatment distribution in $P_{\text{obs}}$, i.e., replacing $\Lambda^a_{\text{obs}}(dt)$ with $\Lambda^a(dt)$ and $\pi_{\text{obs}}(\cdot|\mathcal{H}_t)$ with $\pi(\cdot|\mathcal{H}_t)$.

**Assumption 2.** *Overlap is said to hold between the observational and interventional distributions, $P_{\text{obs}}$ and $P$, if $P$ is absolutely continuous with respect to $P_{\text{obs}}$, denoted by $P \ll P_{\text{obs}}$.*

Ignorability and overlap are the core assumptions that allow identification in our setting. Under these assumptions, we can now present algorithms for estimating causal effects in continuous time.

## 3 Model Free Off-Policy Evaluation for Decision Point Processes

To estimate $\mathbb{E}_P[Y|\mathcal{H}_t]$ for times $t$ and $\mathcal{H}_t$ that overlap with $P_{\text{obs}}$, we express the expectation recursively as a function of expectations $\mathbb{E}_P[Y|\widetilde{\mathcal{H}}_{t+\delta}]$ for some $\delta > 0$ and trajectory $\widetilde{\mathcal{H}}$. Then, assuming that expectations at times larger than $T$ have been learned correctly, this recursive expression allows us to propagate information for conditioning on earlier histories. This type of dynamic programming, by going backwards in time, is a well known solution for discrete time problems. Thus, we first describe it and then address the challenges that arise when adapting it to continuous time.

**Fitted Q evaluation (FQE) in discrete time.** Q-evaluation relies on the tower property of conditional expectations, given below in eq. (1). In discrete time decision processes, where we consider $\delta = 1$, the property suggests a dynamic programming solution that we lay out in algorithm 1 (Le et al., 2019; Watkins and Dayan, 1992). Here, $\mathcal{H}_t$ includes all actions and observations up to and including time $t$, and since they occur simultaneously, there are exactly $t$ of each. $\widetilde{\mathcal{H}}_{t+1}$ is defined in the same manner, except that it includes $\tilde{a}_{t+1}$ sampled from the target policy $\pi$.

**Algorithm 1** Backward-in-time Q-Evaluation

**Input**: Trajectories $\{\mathcal{H}_i\}_{i=1}^m$,
  Policy $\pi : \cup_{t=1}^T \mathcal{X}^t \times \mathcal{A}^{t-1} \to \Delta^{|\mathcal{A}|}$
$Q_T(\mathcal{H}_i) = \arg\min_f \sum_{i=1}^m \left(f(\mathcal{H}_i) - y_i\right)^2$
**for** $t \leftarrow T-1$ to $1$ **do**
  $\hat{y}_i = y_{i,t} + \mathbb{E}_{\tilde{a}_{t+1} \sim \pi(\cdot|\mathcal{H}_{i,t}, \mathbf{x}_{i,t+1})} Q^T(\widetilde{\mathcal{H}}_{t+1})$
  $Q_t(\mathcal{H}_t) = \arg\min_f \sum_{i=1}^m \left(f(\mathcal{H}_{i,t}) - \hat{y}_i\right)^2$
**end for**
**Return** $\{Q_t(\cdot)\}_{t=1}^T$

$$\mathbb{E}_P\left[Y|\mathcal{H}_t\right] = \mathbb{E}_{\widetilde{\mathcal{H}}_{t+\delta} \sim P(\cdot|\mathcal{H}_t)}\left[\mathbb{E}_P\left[Y|\widetilde{\mathcal{H}}_{t+\delta}\right]\right], \tag{1}$$

An attractive property of this algorithm is that it is *model-free*. That is, to form the label $\hat{y}_i$ we only need to sample $\tilde{a}_{t+1}$ from our target policy $\pi$, while $\mathbf{x}_{i,t+1}$ is taken from our training data.

The algorithm is correct because if we assume $Q_{t+1}(\mathcal{H}_{t+1})$ is given and accurately estimates $\mathbb{E}_P[\sum_{s\geq t+1} Y_s|\mathcal{H}_{t+1}]$, then, provided enough samples, the minimizer of the regression in the algorithm is the conditional expectation, which equals $\mathbb{E}_P[\sum_{s\geq t} Y_s|\mathcal{H}_t]$ according to eq. (1). The model-free solution is enabled by the equality $P(\mathbf{x}_{t+1}|\mathcal{H}_t) = P_{\text{obs}}(\mathbf{x}_{t+1}|\mathcal{H}_t)$, which saves us from needing to estimate $P(\mathbf{x}_{t+1}|\mathcal{H}_t)$. In practical implementation, we apply gradient steps on randomly drawn times and training samples instead of walking backward in time from $T$ to 1. Crucially, for $\delta > 1$, e.g. $\delta = 2$, we have $P(\mathbf{x}_{t+2}|\mathcal{H}_t) \neq P_{\text{obs}}(\mathbf{x}_{t+2}|\mathcal{H}_t)$. Hence an algorithm based on eq. (1) will either be model-based, or resort to using solutions such as products of importance weights that suffer from high variance (Hallak et al., 2016; Precup et al., 2000), or restrict the problem, e.g., by discounting rewards (Harutyunyan et al., 2016; Munos et al., 2016; Precup et al., 2000).

**Challenges in application to continuous time.** Moving to continuous time, the tower property turns into a differential equation, and solving it requires tools that go beyond the common FQE solution, e.g. (Jia and Zhou, 2023). Regressing to an outcome that is arbitrarily close to the observation at time $t$ is ill-defined. While we may work under a fine discretization of time, this approach is wasteful, as a single update in the minimization for estimating $Q_t$ takes into account the development of the process in the interval $[t, t+\delta]$, and for small values of $\delta$ this will usually yield a very small change to the estimate. Hence, intuitively, when updating $Q_t$, we would like to use estimates of $Q_{t+\delta}$ for a large $\delta$. As explained above, this is seemingly difficult to achieve in a model-free fashion. However, for point processes, since the number of decisions over $[0, T]$ is countable, it seems plausible that a simple and efficient dynamic programming solution can be devised. In what follows, this is what we present.

### 3.1 EDQ: Fitted Q-Evaluation for Decision Point Processes via Earliest Disagreement Times

Consider the following intuition. Assume we observe the entire trajectory of a patient treated by doctor $c$, who practices policy $\lambda_a^{\text{obs}}$. We wish to reason about the patient's outcome had they, from time $t$ onwards, been treated by doctor $\tilde{c}$ who practices policy $\lambda_a$. With $\mathcal{H}_t$ as the patient history up to time $t$, we look at $c$'s treatment decisions and find the first time in the trajectory, say $t+\delta$, where $\tilde{c}$ would have acted differently from $c$. It seems reasonable to assume the following: the expected outcome of a patient with history $\mathcal{H}_t$, had they been treated with $\lambda_a$ from time $t$ onwards, would be similar to the expected outcome of patients treated with $\lambda_a$ from time $t+\delta$, with history similar to $\mathcal{H}_{t+\delta}$. This is because we have no reason to assume $\mathcal{H}_u$ for $u \in [t, t+\delta)$ would have changed under $\tilde{c}$'s policy $\lambda$, as it agreed with $\lambda_{\text{obs}}$ up until that time point. The time $t+\delta$ is the earliest disagreement time between the two policies.[2] Let us now formalize this intuition and use it to derive an estimation method based on fitted $Q$-evaluation.

**Definition 4.** *Consider a fixed trajectory $\mathcal{H}$ sampled from $P_{obs}$ and a time $t \in [0, T)$, we define a marked point process $\widetilde{P}_t^a$ on the interval $(t, T]$ with intensity function $\lambda^a(u|\mathcal{H}_u)$ and mark distribution $\pi(A_u|\mathcal{H}_u)$. We use the notation $\widetilde{\mathcal{H}}_u^a$ to denote trajectories sampled from $\widetilde{P}_t^a$ up to time $u$ and define $\widetilde{\mathcal{H}}_u := \mathcal{H}_t^{\text{a}} \cup \widetilde{\mathcal{H}}_u^{\text{a}} \cup \mathcal{H}_u^{\text{x,y}}$ for $u \in (t, T]$. For the trajectory $\widetilde{\mathcal{H}}$ we define $\delta_{\widetilde{\mathcal{H}}}(t) = \min\{u - t : u > t, (u, \cdot) \in \widetilde{\mathcal{H}}^a\}$,[3] as the first jump time of a trajectory sampled from $\widetilde{P}_t^a(\cdot|\mathcal{H}_t)$, and likewise $\delta_{\mathcal{H}}(t)$ for the observed trajectory $\mathcal{H}$ instead of $\widetilde{\mathcal{H}}$.*

Here, $\widetilde{\mathcal{H}}$ holds the decisions after time $t$ sampled from our target policy, conditioned on the features observed in $\mathcal{H}$. Notice that to obtain $\widetilde{\mathcal{H}}$, we only need to sample from the target policy $\lambda^a$ and not from $\lambda^{x,y}$, as $\mathcal{H}$ can be taken from our training set sampled from $P_{\text{obs}}$. Our model-free evaluation method is thus based on the following result, which expresses our estimand as an expectation over the trajectories $\widetilde{\mathcal{H}}$.

**Theorem 1.** *Let $P, P_{\text{obs}}$ be multivariate marked decision point processes, $t \in [0, T)$, and $\mathcal{H}_t$ a list of events up to time $t$. For any trajectory $\mathcal{H}$, we let $\widetilde{P}_t^a(\cdot|\mathcal{H}), \delta_{\widetilde{\mathcal{H}}}(t), \delta_{\mathcal{H}}(t)$ as in definition 4. Under*

[2] Under our formulation, with probability 1, this time will be the first treatment after $t$ drawn by one of the policies. This is because we choose to work with continuous and differentiable compensators $\Lambda^a, \Lambda^a_{\text{obs}}$ (a compensator is $\Lambda$ such that $d\Lambda$ is the intensity $\lambda$). Yet this is a technical convenience, and we can also form our arguments under conditions where treatment times have a non-zero probability to coincide, and the arguments about first disagreement times will stay in tact.

[3] In case the set is empty we define the minimum as $T - t$

**Algorithm 2** Earliest Disagreement Fitted Q-Evaluation

**Input**: Trajectories $\{\mathcal{H}_i\}_{i=1}^m$,
Policy $\lambda_a(\cdot|\mathcal{H}_t), \pi(\cdot|\mathcal{H}_t)$
Initialize $\boldsymbol{\theta}$ randomly
**for** $N$ rounds **do**
  Draw $t \sim \text{Unif}([0, T])$ and $i \sim \text{Unif}([m])$
  Draw $\widetilde{\mathcal{H}} \sim \widetilde{P}_t^a(\cdot|\mathcal{H}_{i,t})$ and set $\delta = \delta_{\mathcal{H}}(t) \wedge \delta_{\widetilde{\mathcal{H}}}(t)$
  $\hat{y}_i = \sum_{\substack{(t_k, y_k) \in \mathcal{H}_t^y: \\ t_k \in (t, t+\delta]}} y_k + Q_{t+\delta}(\widetilde{\mathcal{H}}_{t+\delta}; \boldsymbol{\theta})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \left(Q_t(\mathcal{H}_{i,t}; \boldsymbol{\theta}) - \hat{y}_i\right)^2$
**end for**
**Return** $\{Q_t(\cdot; \boldsymbol{\theta})\}$

*assumptions* 1-3*, we have that*

$$\mathbb{E}_P\left[Y|\mathcal{H}_t\right] = \mathbb{E}_{\mathcal{H} \sim P_{\text{obs}}(\cdot|\mathcal{H}_t)}\left[\mathbb{E}_{\widetilde{\mathcal{H}} \sim \widetilde{P}_t^a(\cdot|\mathcal{H})}\left[\mathbb{E}_P\left[Y\middle|\widetilde{\mathcal{H}}_{t+\delta_{\mathcal{H}}(t) \wedge \delta_{\widetilde{\mathcal{H}}}(t)}\right]\right]\right]. \tag{2}$$

**Takeaways from Theorem 1.** Our method, described in algorithm 2, is an empirical version of eq. (2) akin to FQE being the empirical version of the tower property of conditional expectations, eq. (1), in discrete time. An analogous result to ours holds for discrete-time decision processes, where EDQ does not reduce to FQE and instead it bears some resemblance to the eligibility traces approach of Precup et al. (2000). We provide this result in the appendix for completeness but focus here on the point process case, as this is our main motivation and where the earliest disagreement approach is most fruitful. Note that Algorithm 2 does not go backward in time as Algorithm 1 and follows a more practical version of FQE, e.g. (Le et al., 2019, alg. 1), which updates the $Q$ functions at randomly drawn times across the trajectory according to the dynamic program implied by the towered expectations identity. We draw an update time and trajectory uniformly from the interval of the process and the training set accordingly, although other distributions on time may be considered, as well as batched optimization over trajectories. Another notable point is that updates are done upon disagreement in *treatments*, and the trajectory in time $(t, t + \delta]$ may include multiple observations and outcomes $\mathbf{x}, \mathbf{y}$. This is desirable in cases like ICU data, where some vitals are being continuously monitored (e.g., blood pressure), while decisions, such as changing medication dosages, occur on a more coarse timescale. The algorithm will then performs updates on the coarser timescale instead of the finer one.

## 4 Related Work

Our coverage of related work is divided into an overview of works that solve adjacent tasks to ours, before transitioning into a detailed discussion in section 4.1 about techniques more closely aligned with our goal of large scale causal inference in sequential decision making.

**Causal inference with sequential decisions.** Estimation of causal effects for sequential treatments is usually studied in discrete time under the sequential exchangeability assumption (Hernan and Robins, 2023; Robins, 1986). Addressing unobserved confounders is also a topic of interest (e.g. (Namkoong et al., 2020; Tennenholtz et al., 2020)), but this is beyond the scope for our work. As described in section 2.2, the framework of Røysland et al. (2022) draws a parallel to sequential exchangeability in continuous-time processes, which we adopt in this work. Regarding estimation, several approaches for continuous or irregularly sampled times have been explored (Lin et al., 2004; Lok, 2008; Røysland, 2011; Rytgaard et al., 2022; Zhang et al., 2011). Most estimation methods studied in this context do not scale to large and high-dimensional datasets. For instance, Røysland (2011) requires estimating an integral of propensity weights across time, and Rytgaard et al. (2022) propose a targeted estimator that fits each process in $P_{\text{obs}}$. However, this method is limited to interventions at fixed time points (i.e. no interventions on scheduling of treatments) and it is unclear how to implement the proposed estimations with large and expressive models.

**Reinforcement learning techniques.** The dynamic treatment regimes literature (e.g. (Chakraborty and Moodie, 2013; Chakraborty and Murphy, 2014)) studies policy learning and evaluation in non-Markov decision processes, which is a similar setting to ours but in discrete time. Q-learning is

a prominent model-free solution in this scenario, where dynamic programming going "backwards in time" yields the correct policy value (Murphy, 2005). This solution is unsuitable for irregularly sampled times, but motivates EDQ. We discuss the connection of Q-learning and related n-step methods from Reinforcement Learning (De Asis et al., 2018; Munos et al., 2016; Precup et al., 2000) to EDQ in section 3. Some work in reinforcement learning has considered irregularly sampled times, but their solutions are not applicable to our setting of interest. They either do not intervene on time or operate in the on-policy setting (Qu et al., 2023; Upadhyay et al., 2018), or incorporate continuous time positional embeddings into decision transformers (Chen et al., 2021b). The latter facilitates recommending sets actions to arrive at a desired outcome (Zhang et al., 2023) rather than evaluating a policy of interest. Furthermore, recent work suggests these goal-conditioned imitation learning methods such as decision transformers may fail to estimate the causal effect of actions (Malenica and Murphy, 2023) in some scenarios where there are no unobserved confounders, whereas $Q$-learning methods produce correct estimates.

We now turn to a more in-depth discussion of scalable causal estimation methods for sequential treatments, where we delineate some of the technical assumptions, implementation choices, and subsequent properties of solutions covered in recent work on the problem.

### 4.1 Large Scale Estimation Approaches for Sequential Treatments

Table 1: Qualitative comparison of effect estimation methods under sequential treatments, compatible with large scale ML implementation. The method we presented in this work is marked in bold, while others can be found in (Bica et al., 2020; Chakraborty and Moodie, 2013; Le et al., 2019; Li et al., 2021; Lim, 2018; Melnychuk et al., 2022; Schulam and Saria, 2017; Seedat et al., 2022)

| | Properties | | | Estimation Method | | | |
|---|---|---|---|---|---|---|---|
| | **Irregular Times** | **Large Scale** | **Dynamic Policy** | **Prop. Weights** | **Model Based** | **Balancing Rep.** | **DP** |
| CGP | ✓ | ✗ | ✓ | | ✓ | | |
| CRN , CT | ✗ | ✓ | ✗ | | | ✓ | |
| R-MSN | ✗ | ✓ | ✗ | ✓ | | | |
| TE-CDE | ✓ | ✗[4] | ✗ | | ✓ | ✓ | |
| G-Net | ✗ | ✓ | ✓ | | ✓ | | |
| FQE | ✗ | ✓ | ✓ | | | | ✓ |
| **EDQ** | ✓ | ✓ | ✓ | | | | ✓ |

Notable early work on using machine learning models for estimating counterfactual quantities related to treatments on a timeline, (Schulam and Saria, 2017), used Gaussian Processes to tackle the estimation problem. Limitations such as scaling to large datasets and incorporating various features prompted the development of deep learning approaches to the problem.

**Large scale models.** One family of solutions (Bica et al., 2020; Lim, 2018; Melnychuk et al., 2022) took an important step forward by using RNNs and transformers for the estimation problem. All of these methods build on the idea of learning balancing representations (Johansson et al., 2016). Roughly, these are representations under which the treatment is randomly assigned. This facilitates the training of high-capacity effect estimators on large datasets, but these works are restricted to discrete times.

**Dynamic policies.** The above methods can only estimate effects of *static* treatments, meaning the treatment plan cannot dynamically depend on future observations. For instance, consider a policy that prescribes a daily dose of statins, and if the patient develops side effects in the future, switches to another medication. This policy depends on possible future states, which the above methods do not accommodate for that. G-Net (Li et al., 2021; Xiong et al., 2024) takes a model-based approach, where models are fit for both $\pi_{\text{obs}}(A(t)|\mathcal{H}_{t-1}, X(t))$, and $P_{\text{obs}}(X(t), Y(t)|\mathcal{H}_{t-1})$. Then at inference time, a dynamic policy is estimated when $\pi_{\text{obs}}$ is replaced with the desired policy $\pi$ and conditional expectations of $Y$ are estimated with Monte-Carlo simulations.

[4]We mark TE-CDE with a crossed checkmark for scalability, since the algorithm relies on differential equation solvers which limits the architectures one can use.

**Irregular times.** None of the above solutions handle irregular times of observations and actions, which is one of the main goals of this paper. Some steps in this direction have been taken by Seedat et al. (2022); Vanderschueren et al. (2023). Seedat et al. (2022) use balanced representations, like previously mentioned works, but combine them with a neural CDE architecture that is shown to be more suitable for irregular sampling times. Vanderschueren et al. (2023) experiment with a reweighting technique to account for sampling times that are informative of the outcome. These works do not estimate outcomes under interventions on treatment times but instead seek to mitigate biases induced by sampling times on effect estimation. We further discuss aspects of these related works, including their identifiability conditions, in appendix C.

Table 1 summarizes the properties of all the above techniques, along with Fitted Q-Evaluation and EDQ from section 3. Notably, EDQ possesses all the desirable qualities mentioned here and handles interventions on $\lambda^a$, which these solutions do not.

## 5 Implementation and Experiments

To implement EDQ for experimentation in section 5 we use a GPT-2 architecture and modify it in the following manner. Each token is a concatenation of embeddings of time $t_i$, value $\mathbf{z}_i$ and a type of event $e_i \in \{A, X, Y, \Delta T\}$. The event types $A, X, Y$ correspond to actions, features, and outcomes, while the $\Delta T$ event is introduced for convenience, as we wish to represent trajectories where time has passed but no event has yet occured. For example, this allows our model to represent quantities such as $\mathbb{E}_P[Y|\mathcal{H}_{t+\delta} = \mathcal{H}_t]$, which may appear in eq. (2). Whereas $t_i$ represents the absolute time passed until a certain event, the value of timestep tokens represents time gaps, yet both are embedded with a continuous time positional embedding: $\sin\left(tC_k/d_{\text{time}}\right)$ for even $k$, and $\cos\left(tC_{k-1}/d_{\text{time}}\right)$ for odd $k$. Here $C = 10^5$ and $d_{\text{time}}$ is the embedding dimension. We also keep a target network as and update it with soft-Q updates, as is common in Deep Q-Networks, e.g. Van Hasselt et al. (2016). We implement two methods as baselines in our experiment.

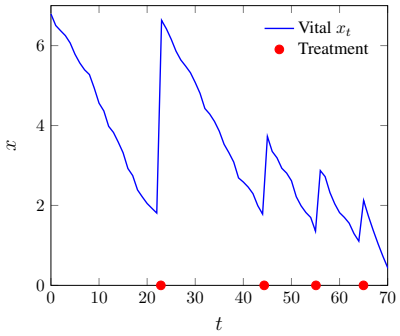
### 5.1 Baselines

Since we are unaware of algorithms that perform effect estimation on treatment timing with high-dimensional or long sequence data, we implement two baselines that let us glean some important aspects of EDQ.

**ERM / MC** is an Empirical Risk Minimizer (ERM) that is trained to predict observed outcomes, which, in the context of reinforcement learning and policy evaluation is also called Monte-Carlo prediction (MC). We use the same GPT-2 architecture and data representation as EDQ, but instead of running algorithm 2, we simply seek to learn $f_{\boldsymbol{\theta}}(\mathcal{H}_t)$ that minimizes prediction loss on observed data. Assuming each training trajectory $\mathcal{H}_i$ comes with a label $y_i$ of its outcome, we solve $\min_\theta m^{-1} \sum_{i,(t_i,\mathbf{z}_i)\in\mathcal{H}_i} (\ell(f_{\boldsymbol{\theta}}(\mathcal{H}_{i,t}), y_i)$ with gradient updates, where $\ell(\cdot,\cdot)$ is set as the squared loss. Since this method estimates outcomes under the observed policy $\lambda_{\text{obs}}$, we expect it to perform as well as, or better than, off-policy evaluation methods, including EDQ, when $\lambda = \lambda_{\text{obs}}$, and to suffer a drop otherwise.

**FQE** Is implemented as described in section 3, but with discretized time and Q-updates using one timestep forward in time. That is, at each iteration, we draw a training example $i \in [m]$ and time $t \in [0, T]$, define $\hat{y}_i = y_{t+1} + Q_{t+1}(\widetilde{\mathcal{H}}_{t+1}; \boldsymbol{\theta}))$, and perform a gradient step on the loss $\ell(Q_t(\mathcal{H}_{i,t}), \hat{y}_i)$. Similarly, we define discrete-time approximations of our policies of interest, which will be described later. In terms of implementation, the positional embeddings now correspond to discrete times, and the representations of actions, features and outcomes at each timestep are concatenated. Other than this, we use the exact same architecture and hyperparameters of EDQ. This baseline examines the effects of time discretization on estimation quality and optimization.

*A comment on computational complexity:* The per-iteration runtime of EDQ is similar to that of FQE, which is a common tool in large-scale offline RL problems; for example, Paine et al. (2020); Voloshin et al. (2021) use it in benchmarks and evaluations. The difference in computation times between EDQ and FQE is due to sampling from the target policy, in order to draw the treatments used in the $Q$-update. We expand on this discussion in appendix A.




| | ERM / MC | FQE | EDQ |
|---|---|---|---|
| | | $\lambda_{\text{int}} = 0.1$ | |
| $\lambda_{\text{obs}} = 0.1$ | $0.20 \pm 0.007$ | $0.21 \pm 0.02$ | $0.20 \pm 0.005$ |
| $\lambda_{\text{obs}} = 0.5$ | $0.38 \pm 0.011$ | $0.23 \pm 0.04$ | $0.20 \pm 0.006$ |
| | | $\lambda_{\text{int}} = 0.5$ | |
| $\lambda_{\text{obs}} = 0.5$ | $0.11 \pm 0.004$ | $0.197 \pm 0.013$ | $0.10 \pm 0.003$ |
| $\lambda_{\text{obs}} = 0.1$ | $0.28 \pm 0.004$ | $0.31 \pm 0.01$ | $0.11 \pm 0.006$ |

Figure 2: **Left.** An example of a trajectory from our simulation. Blue curve denotes the value of the vital $x_t$ and red dots mark treatment times. **Right.** Normalized RMSE of the methods under the different simulation settings. Mean is taken over all points in the history of patients in the test data. Rows colored blue are those where $\lambda_{\text{obs}} = \lambda_{\text{int}}$, and we expect all methods to perform well since training and test data are sampled from the same distribution. Red rows are those where the effect of an intervention needs to be estimated.

### 5.2 Simulations on Time to Failure and Cancer Tumor Growth Prediction

To validate the efficacy of our methods, we construct two simulated settings. In one, the task is to predict the effect of treatment timing policies on patients' time-to-event. The second setting uses a cancer tumor growth simulator from Geng et al. (2017) to form a policy evaluation problem on applications of chemotherapy and radiotherapy.

**Simulators.** We use two simulators. **(i) Time-to-failure:** In this setting, each data point simulates the vital of a patient $x_t \in \mathbb{R}_+$ measured regularly at a frequency of one time unit, and treatments $a_t \in \mathbb{R}_+$ are assigned irregularly in time according to an observational policy. Without treatment, the vital drops linearly $dx_t/dt = -(\alpha + \xi_t)$ where $\xi_t \sim \mathcal{N}(0, \sigma)$ is a noise term drawn at each time unit. Upon receiving treatment, the vital rises by an amount proportional to the number of treatments, $1 \leq k \leq m$, applied up until that time, where $m$ is the maximal number of treatments that a patient can receive. That is, the efficacy of treatment reduces with repeated applications. We also inject small noise terms into the dosage of treatment that a patient receives, which further affect the vital and add randomness to the problem. Section 5.2 shows an example of a simulated patient trajectory. **(ii) Tumor growth:** We use the experimental setting from Bica et al. (2020), which other works use to study irregular sampling (Seedat et al., 2022; Vanderschueren et al., 2023). As this is a commonly used simulator, we defer the details on its dynamics to appendix A and focus on the type of irregular sampling and policies we use. The simulator works in discrete time $t \in [T]$, and irregular sampling is induced by the features being unobserved at certain times. Namely, the covariate $x_t \in \mathbb{R}_+$ which represents tumor volume is observed with probability $\sigma((\bar{x}_{t-d:t-1}/d_{\max}) - 1.5)$, where $\bar{x}_{t-d:t-1}$ is the average tumor volume over the last $d$ timesteps, and $d_{\max}$ is the maximum considered volume.

**Outcomes and policies.** For **(i) time-to-failure**, our outcome of interest is failure time $y \in \mathbb{R}_+$, where a patient dies if the vital drops to a value of 0. [5] We focus on effect estimation for interventions on a rate parameter $\lambda^a$ that controls the timing of treatment. At each time $t$ where the observed vital crosses a threshold, i.e. $x_t < r$ for some predetermined $r \in \mathbb{R}_+$, a random time is drawn from an exponential distribution $\delta \sim \exp(\lambda^a)$ and treatment is applied at $t + \delta$. The threshold $r$ and the dosage of treatment given are also part of the policy $\pi$, yet to focus on the effects of timing we do not intervene on them in this experiment. At each experiment we observe $m$ patients treated under a policy with $\lambda^a = \lambda_{\text{obs}}$ and aim to reason about the expected failure times under the interventional $\lambda_{\text{int}}$. In **(ii) tumor-growth**, the goals is to predict $x_T$ where $T = 20$, when the policy at time $t$ assigns treatment $a_t \in [4]$ which is either no-treatment, radiotherapy, chemotherapy or a combined therapy. Therefore, the estimation here is both on "when" and "what" to do. Policies are determined by two parameters $(\gamma, \beta)$ and assign each type of treatment with probability $\sigma(\gamma(x_{\text{last}} - \beta) + t - t_{\text{last}})$. Here, $x_{\text{last}}$ is the last observed volume and $\beta$ is an intercept controlling how often treatments are

[5] Note that the vital changes outside measurement times, hence death time does not generally coincide with vital measurement times.

| | ERM / MC | FQE | EDQ |
|---|---|---|---|
| | $(\gamma, \beta)_{\text{int}} = (6, 0.75)$ | | |
| $(\gamma, \beta)_{\text{obs}} = (6, 0.75)$ | $0.034 \pm 0.001$ | $0.048 \pm 0.001$ | $0.037 \pm 0.001$ |
| $(\gamma, \beta)_{\text{obs}} = (10, 0.5)$ | $0.07 \pm 0.004$ | $0.080 \pm 0.013$ | $0.052 \pm 0.006$ |

| | ERM / MC | FQE | EDQ |
|---|---|---|---|
| | $\lambda_{\text{int}} = 0.2$ | | |
| $\lambda_{\text{obs}} = 0.2$ | $0.17 \pm 0.01$ | $0.18 \pm 0.004$ | $0.178 \pm 0.01$ |
| $\lambda_{\text{obs}} = 2$ | $0.28 \pm 0.01$ | $0.20 \pm 0.03$ | $0.178 \pm 0.01$ |
| | $\lambda_{\text{int}} = 2$ | | |
| $\lambda_{\text{obs}} = 2$ | $0.22 \pm 0.01$ | $0.197 \pm 0.013$ | $0.22 \pm 0.004$ |
| $\lambda_{\text{obs}} = 0.2$ | $0.32 \pm 0.02$ | $0.31 \pm 0.01$ | $0.22 \pm 0.007$ |

Figure 3: **Left.** Normalized RMSE on the **tumor-growth** simulation. All methods are affected by distribution shift. EDQ is the most robust out of the baselines considered. **Right.** Normalized RMSE for **time-to-failure** simulation on short trajectories.

applied, while $\gamma$ controls the dependence of treatment assignment on tumor volume. Finally $t_{\text{last}}$ is the time of the last treatment, and the term $t - t_{\text{last}}$ induces a lag between consecutive treatments.

**Experiments.** We perform two sets of experiments for the **time-to-failure** simulation. In the first set, trajectory lengths range between 10 and 100, and the number of possible treatments equals 5. In the second set (results in Figure 3, right), we change the parameters of the problem by taking a high slope $\alpha$ and capping the number of treatments at 1. This creates short trajectories of length between 3 and 10. To evaluate the performance of the estimator, we sample trajectories $(\mathcal{H}_i, y_i) \sim P_{\lambda_{\text{int}}}$ under the target policy and treat every $(\mathcal{H}_{i,t_i}, y_i)$ as a labeled data point. We then evaluate normalized RMSE between $f_{\boldsymbol{\theta}}(\mathcal{H}_{i,t_i})$ and the true labels $y_i$. For the **tumor growth** experiment we evaluate a policy that increases the likelihood of treatments (i.e increases $\beta$) and reduces $\gamma$, the correlation to the observed volume. Error is also calculated with normalized RMSE.

**Results.** The tables in fig. 2 and fig. 3 present the results of both simulations. They show that for the **time-to-failure** simulation, EDQ solves the estimation problem both when $\lambda_{\text{obs}} = \lambda_{\text{int}}$ (blue rows, no intervention performed), and when $\lambda_{\text{obs}} \neq \lambda_{\text{int}}$ (red rows). This is evident by comparing its performance with ERM under the setting where $\lambda_{\text{obs}} = \lambda_{\text{int}}$, as ERM should be nearly optimal in that setting.[6] ERM takes a significant performance drop when $\lambda_{\text{obs}} \neq \lambda_{\text{int}}$ as expected. As for FQE, while in the first set of experiments, depicted in fig. 2, discretization should not result in significant information loss, it does create a more difficult optimization problem for FQE. This is because the updates to $Q(\mathcal{H}_t)$ need to propagate backwards and most of the updates get noisy gradient signals by fitting to the $Q$ value of a trajectory sampled one time step forward $Q(\widetilde{\mathcal{H}}_{t+1})$. This challenge for FQE is most evident in the results of Figure 2, where $\lambda_{\text{int}} = 0.5$ and it incurs a significant loss both when $\lambda_{\text{obs}} = \lambda_{\text{int}}$ and when $\lambda_{\text{obs}} \neq \lambda_{\text{int}}$. The results in the right table of Figure 3, demonstrate potential effects of information loss due to time discretization. Here, since the trajectories are short, the optimization problem of losses propagating along the trajectory is likely less pronounced. However, we see that there is still a significant drop when $\lambda_{\text{int}} = 2$ and $\lambda_{\text{obs}} = 0.2$, that is, approximating a high rate of treatment when data was sampled under low rates. Taken together, these two experiments demonstrate two possible drawbacks of discretizing time. For **tumor-growth**, EDQ still outperforms the alternatives but suffers a certain decrease in performance due to the distribution shift between the observational and interventional distributions.

## 6 Limitations and Future Work

To summarize this work, we have developed a method for off-policy evaluation with irregular treatment and observation times, which facilitates interventions on treatment intensities. We connected the setting with identifiability results from the causal inference literature to highlight the conditions under which the estimates are correct, and proved the correctness of our estimator. EDQ is a "direct" method based on fitting regressions and. as demonstrated in our experiments, it is easily applicable to high-capacity sequence modeling architectures. To the best of our knowledge, it is the first available solution to this estimation problem that is applied with such architectures. There are several limitations to this work, which motivate exciting future research. Empirically, we plan to apply the method to large real-world datasets to study effects of intervening on treatment times. The method also does not handle censoring, which is required in order to reliably apply it in most survival analysis and real-world trial data. Additional potential technical developments include policy optimization in the setting we studied here and deriving bounds on errors due to unobserved confounding.

[6] this is up to numerical optimization issues, as we see FQE can outperform it in certain cases

## REFERENCES


Aalen, O., Borgan, O., and Gjessing, H. (2008). *Survival and event history analysis: a process point of view*. Springer Science & Business Media.

Aalen, O. O. (1987). Dynamic modelling and causality. *Scandinavian Actuarial Journal*, 1987(3-4):177–190.

Andersen, P. K., Borgan, O., Gill, R. D., and Keiding, N. (2012). *Statistical models based on counting processes*. Springer Science & Business Media.

Bica, I., Alaa, A. M., Jordon, J., and van der Schaar, M. (2020). Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*.

Chakraborty, B. and Moodie, E. E. (2013). Statistical methods for dynamic treatment regimes. *Springer-Verlag. doi*, 10(978-1):4–1.

Chakraborty, B. and Murphy, S. A. (2014). Dynamic treatment regimes. *Annual review of statistics and its application*, 1:447–464.

Chen, I. Y., Joshi, S., Ghassemi, M., and Ranganath, R. (2021a). Probabilistic machine learning for healthcare. *Annual review of biomedical data science*, 4(1):393–415.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021b). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

De Asis, K., Hernandez-Garcia, J., Holland, G., and Sutton, R. (2018). Multi-step reinforcement learning: A unifying algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Didelez, V. (2008). Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(1):245–264.

Fu, J., Norouzi, M., Nachum, O., Tucker, G., Wang, Z., Novikov, A., Yang, M., Zhang, M. R., Chen, Y., Kumar, A., et al. (2021). Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*.

Geng, C., Paganetti, H., and Grassberger, C. (2017). Prediction of treatment response for combined chemo-and radiation therapy for non-small cell lung cancer patients using a bio-mathematical model. *Scientific reports*, 7(1):13542.

Hallak, A., Tamar, A., Munos, R., and Mannor, S. (2016). Generalized emphatic temporal difference learning: Bias-variance analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. (2016). Q () with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pages 305–320. Springer.

Hernan, M. and Robins, J. (2023). *Causal Inference: What If*. Chapman & Hall/CRC Monographs on Statistics & Applied Probab. CRC Press.

Jacobsen, M. and Gani, J. (2006). Point process theory and applications: marked point and piecewise deterministic processes.

Jia, Y. and Zhou, X. Y. (2023). q-learning in continuous time. *Journal of Machine Learning Research*, 24(161):1–61.

Johansson, F., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029. PMLR.

Le, H., Voloshin, C., and Yue, Y. (2019). Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712. PMLR.

Lewis, P. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413.

Li, R., Hu, S., Lu, M., Utsumi, Y., Chakraborty, P., Sow, D. M., Madan, P., Li, J., Ghalwash, M., Shahn, Z., and Lehman, L.-w. (2021). G-net: a recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime. In Roy, S., Pfohl, S., Rocheteau, E., Tadesse, G. A., Oala, L., Falck, F., Zhou, Y., Shen, L., Zamzmi, G., Mugambi, P., Zirikly, A., McDermott, M. B. A., and Alsentzer, E., editors, *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pages 282–299. PMLR.

Lim, B. (2018). Forecasting treatment responses over time using recurrent marginal structural networks. *Advances in neural information processing systems*, 31.

Lin, H., Scharfstein, D. O., and Rosenheck, R. A. (2004). Analysis of longitudinal data with irregular, outcome-dependent follow-up. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(3):791–813.

Lok, J. J. (2008). Statistical modeling of causal effects in continuous time. *The Annals of Statistics*, pages 1464–1507.

Malenica, I. and Murphy, S. (2023). Causality in goal conditioned rl: Return to no future? In *NeurIPS 2023 Workshop on Goal-Conditioned Reinforcement Learning*.

McDermott, M., Nestor, B., Argaw, P., and Kohane, I. (2023). Event stream gpt: A data pre-processing and modeling library for generative, pre-trained transformers over continuous-time sequences of complex events. *arXiv preprint arXiv:2306.11547*.

Melnychuk, V., Frauen, D., and Feuerriegel, S. (2022). Causal transformer for estimating counterfactual outcomes. In *International Conference on Machine Learning*, pages 15293–15329. PMLR.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. *Advances in neural information processing systems*, 29.

Murphy, S. A. (2005). A generalization error for q-learning.

Nagpal, C., Jeanselme, V., and Dubrawski, A. (2021). Deep parametric time-to-event regression with time-varying covariates. In *Survival Prediction-Algorithms, Challenges and Applications*, pages 184–193. PMLR.

Namkoong, H., Keramati, R., Yadlowsky, S., and Brunskill, E. (2020). Off-policy policy evaluation for sequential decisions under unobserved confounding. *Advances in Neural Information Processing Systems*, 33:18819–18831.

Ogata, Y. (1981). On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31.

Paine, T. L., Paduraru, C., Michi, A., Gulcehre, C., Zolna, K., Novikov, A., Wang, Z., and de Freitas, N. (2020). Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*.

Precup, D., Sutton, R. S., and Singh, S. P. (2000). Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 759–766.

Qu, C., Tan, X., Xue, S., Shi, X., Zhang, J., and Mei, H. (2023). Bellman meets hawkes: Model-based reinforcement learning via temporal point processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9543–9551.

Robins, J. (1986). A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512.

Røysland, K. (2011). A martingale approach to continuous-time marginal structural models.

Røysland, K. (2012). Counterfactual analyses with graphical models based on local independence.

Røysland, K., Ryalen, P., Nygaard, M., and Didelez, V. (2022). Graphical criteria for the identification of marginal causal effects in continuous-time survival and event-history analyses. *arXiv preprint arXiv:2202.02311*.

Rytgaard, H. C., Gerds, T. A., and van der Laan, M. J. (2022). Continuous-time targeted minimum loss-based estimation of intervention-specific mean outcomes. *The Annals of Statistics*, 50(5):2469–2491.

Schulam, P. and Saria, S. (2017). Reliable decision support using counterfactual models. *Advances in neural information processing systems*, 30.

Schweder, T. (1970). Composable markov processes. *Journal of applied probability*, 7(2):400–410.

Seedat, N., Imrie, F., Bellot, A., Qian, Z., and van der Schaar, M. (2022). Continuous-time modeling of counterfactual outcomes using neural controlled differential equations. *arXiv preprint arXiv:2206.08311*.

Snyder, D. L. and Miller, M. I. (2012). *Random point processes in time and space*. Springer Science & Business Media.

Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.

Tennenholtz, G., Shalit, U., and Mannor, S. (2020). Off-policy evaluation in partially observable environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10276–10283.

Uehara, M., Shi, C., and Kallus, N. (2022). A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*.

Upadhyay, U., De, A., and Gomez Rodriguez, M. (2018). Deep reinforcement learning of marked temporal point processes. *Advances in neural information processing systems*, 31.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Vanderschueren, T., Curth, A., Verbeke, W., and van der Schaar, M. (2023). Accounting for informative sampling when learning to forecast treatment outcomes over time. *arXiv preprint arXiv:2306.04255*.

Voloshin, C., Le, H. M., Jiang, N., and Yue, Y. (2021). Empirical study of off-policy policy evaluation for reinforcement learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

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

Xiong, H., Wu, F., Deng, L., Su, M., and Lehman, L.-w. H. (2024). G-transformer: Counterfactual outcome prediction under dynamic and time-varying treatment regimes. *arXiv preprint arXiv:2406.05504*.

Zhang, M., Joffe, M. M., and Small, D. S. (2011). Causal inference for continuous-time processes when covariates are observed only at discrete times. *Annals of statistics*, 39(1).

Zhang, Z., Mei, H., and Xu, Y. (2023). Continuous-time decision transformer for healthcare applications. In *International Conference on Artificial Intelligence and Statistics*, pages 6245–6262. PMLR.