Longitudinal Targeted Minimum Loss-based Estimation with Temporal-Difference Heterogeneous Transformer

Toru Shirakawa [1 2]   Yi Li [2]   Yulun Wu [2]   Sky Qiu [2]   Yuxuan Li [2]
Mingduo Zhao [2]   Hiroyasu Iso [1 3]   Mark van der Laan [2]

Abstract

We propose Deep Longitudinal Targeted Minimum Loss-based Estimation (Deep LTMLE), a novel approach to estimate the mean of counterfactual outcome under dynamic treatment policies in longitudinal problem settings. Our approach utilizes a transformer architecture with heterogeneous type embedding trained using temporal-difference learning. After obtaining an initial estimate using the transformer, following the targeted minimum loss-based likelihood estimation (TMLE) framework, we statistically corrected for the bias commonly associated with machine learning algorithms. Furthermore, our method also facilitates statistical inference by enabling the provision of 95% confidence intervals grounded in asymptotic statistical theory. Simulation results demonstrate our method's superior performance over existing approaches, particularly in complex, long time-horizon scenarios. It remains effective in small-sample, short-duration contexts, matching the performance of asymptotically efficient estimators. To demonstrate our method in practice, we applied our method to estimate counterfactual mean outcomes for standard versus intensive blood pressure management strategies in a real-world cardiovascular epidemiology cohort study.

1. Introduction

In the fields of medicine and public health, researchers frequently encounter data that are both high-dimensional and longitudinal. The outcomes of interest in these settings often involve time to the incidence of some failure event, such as total mortality (van der Laan & Robins, 2003; Salerno & Li, 2023). Estimating the counterfactual probability of the event is challenging in high-dimensional longitudinal settings. Existing methods suffer computationally due to lack of scalability and have worse performance due to curse-of-dimensionality (Wyss et al., 2022). In response, we propose an estimator that is computationally scalable and simultaneously allows for robust statistical inference. Our estimator incorporates a transformer architecture for estimating the target estimand, defined as the cumulative incidence probability under dynamic interventions, where the treatment sequence depends on patients' evolving histories. The target estimand can be identified through the g-formula contingent upon suitable assumptions (Robins, 1986). However, the target functional involves integration over potentially high-dimensional time-dependent covariates across time-horizon, posing computational challenges. Our method advances the longitudinal targeted minimum loss-based estimation (LTMLE) framework (van der Laan & Gruber, 2012; Lendle et al., 2017) by leveraging the computational capabilities of the transformer, facilitating the estimation of the target estimand and relevant nuisance parameters.

A number of estimators for the target estimand were proposed since the pioneering work by Robins (Robins, 1986). These estimators first factor the target parameter as a functional of nuisance parameters given a structural assumption on the underlying variables. Then, a common strategy to construct an estimator is plug-in, where one estimate the nuisance components with some models and then plug them into the target functional. However, since the naive plug-in of the estimated nuisance components causes bias, several methods have been proposed to remove this bias using the first variation of the target functional called influence function. Examples of such debiasing techniques include one-step estimators (Klaassen, 1987; Bickel et al., 1993), estimating equations (Robins et al., 1994; Chernozhukov et al., 2022), and targeted minimum loss-based estimation (TMLE) (van der Laan & Rose, 2011). Notably, due to its plug-in property, TMLE stands out because it will respect any conditional bounds on the outcome or global bounds on the statistical model, resulting in improved finite-sample performance (Gruber & van der Laan, 2012).

[1]Osaka University Graduate School of Medicine, Suita, Japan [2]Univerity of California, Berkeley, United States [3]National Center for Global Health and Medicine, Tokyo, Japan. Correspondence to: Toru Shirakawa <shirakawa@pbhel.med.osaka-u.ac.jp>.

The first-order bias of the plug-in estimator is represented as a population mean of the influence function evaluated at the estimated nuisance distribution. Bias correction is performed by solving the empirical analogue of this term. TMLE solves this term by optimizing a loss function along a submodel starting from the initial nuisance estimate (Bang & Robins, 2005; van der Laan & Rubin, 2006; van der Laan & Rose, 2011). The loss function and the submodel are chosen so that the linear span of the derivative of the loss function along the submodel contains the efficient influence function, the influence function with minimal variance. Targeting is the term that refers to this correction by fluctuating of the initial estimate along the path.

The current LTMLE, a TMLE developed in the context of longitudinal data, relies on a sequential regression representation of the target estimand (Bang & Robins, 2005). An ensemble machine learning technique called *super learner* is then used to estimate the nuisance components of the data-generating distribution (van der Laan et al., 2007). In real-world complex longitudinal data, these nuisance components, such as the survival probability at a given time, may depend on all past histories. Therefore, the Markovian property, which states that future variable values only depends on the present variables, independent of the past, is not guaranteed to hold. In other words, every observed variable could depend on the past variables in the time ordering. Hence, we want our model for the nuisance components to be able to take variable length of history as input. Furthermore, LTMLE performs sequential regressions starting from the last time point and thus cannot pool data across time points. Therefore, the error from the first regression at the last time point, which is the hardest regression because of the least observations due to failure and censoring with the most high-demensional covariates with all the history through the observation period, propagates to the final estimate. Under the targeted learning framework, we introduce a transformer architecture tailored towards our longitudinal setting capable of pooling time points, and propose a novel method for the bias correction using a single fluctuation parameter across all time-points.

Our contribution includes: 1) Developed a general method that uses a transformer architecture to facilitate valid statistical inference in longitudinal settings concerning survival outcomes under dynamic interventions; 2) Proposed a method for bias correction using one-dimensional fluctuation for any length of time-horizon; 3) Demonstrated competetive statistical performance with asymptotically efficient estimators in simple and low-dimensional settings and superior statistical and computational performances in more complex settings; and 4) Applied our method to a real-world medical data with results presented in a format that aligns with clinical research recommendations (Hernán et al., 2022; Dahabreh & Bibbins-Domingo, 2024).

2. Related Work

In the data science literature, several methods were proposed that predict the counterfactual outcomes from patient history. The methods include G-Net (Li et al., 2021), counterfactual recurrent network (CRN) (Bica et al., 2020), and causal transformer (CT) (Melnychuk et al., 2022). However, their target parameters do not involve survival outcomes, and their methods are optimized for the mean squared error (MSE) of the individual predictions, rather than for making statistical inferences. DeepACE (Frauen et al., 2023) is closely related to the present study which uses deep neural networks to estimate the whole propensity scores and outcome regressions simultaneously. Furthermore, it has an additional layer for targeting implementing the one-dimensional submodel proposed by van der Laan (van der Laan & Rose, 2018). Our method differs from theirs in the following three aspects. First, DeepACE incorporates the targeting mechanism within their loss function, which requires an additional hyperparameter. However, there is a lack of justification for the chosen value of this hyperparameter and guidance on its tuning in practical applications. Our approach, in contrast, separates the targeting step, aligning more closely with the TMLE literature. Second, DeepACE does not address survival outcomes, specifically failing to consider the process degeneracy following a patient's event occurrence. Third, while DeepACE utilizes the long short-term memory (LSTM) architecture, our method employs transformers. Transformers are superior in capturing long-term dependencies and offer greater computational efficiency during training than LSTM. Moreover, DeepACE does not provide uncertainty measures, such as confidence intervals, limiting its utility for statistical inference.

Our problem of estimating mean of counterfactual outcomes from longitudinal observational data under dynamic interventions has been extensively investigated as an off-policy evaluation problem in the bandit algorithm and reinforcement learning literature (Levine et al., 2020). Methods of bias correction after plugging in the initial estimate with influence function were also introduced in this context (Jiang & Li, 2016; Farajtabar et al., 2018; Narita et al., 2021). However, they did not provide tools for inference. Double reinforcement learning (Kallus & Uehara, 2020) utilized the efficient influence functions in the spirit of double machine learning (Chernozhukov et al., 2018), which is a closed form of a more general debiased estimating equation framework (Chernozhukov et al., 2022), to correct plug-in bias and proved efficiency. TMLE deform the distribution itself to correct bias before plugged-in to the the target functional, thereby the values are contained the domain of the functional. Since the estimator based on estimating equation was reported to have inferrior performance to LTMLE even in short time-horizons (Tran et al., 2019), we focus on plug-in estimators in the present study.

3. Problem Formulation

In this section, follwing the roadmap of causal inference (Petersen & van der Laan, 2014; van der Laan & Rose, 2018; Dang et al., 2023), we first describe the causal structure that generated the observed data and the statistical model that contains the data-generating distribution. Next, we define our causal target parameter. Then, we discuss assumptions needed to identify our target parameter from the observed data. Finally, we describe the idea of statistical method for constructing estimator and correcting bias.

3.1. Data

We consider the general longitudinal setting involving repeated measurements of a set of variables for a group of n patients over a period of time. In particular, our observed data contains n independent and identically distributed copies of random vector

$$O = (W_0 = W, L_1, A_1, Y_1, \ldots, L_T, A_T, Y_T = Y) \quad (1)$$

with baseline covariates W, time-dependent covariates L_t, treatments A_t, and outcome Y_t. We use P_0 to denote the true probability distribution of O that generated the data, and P_0 is in some statistical model $\mathcal{M}$. Stopping time T is a random variable (e.g. time of death in the case of survival analysis) and we use τ to denote the maximum time. We make the remark that in real-world data, patients are often subject to censoring. For a formulation of the data structure involving censoring nodes, see Appendix I.

3.2. Target Parameter

To define the target parameter, we introduce a structural causal model (SCM). In brief, SCM assumes each observed random variable X is generated from the parent nodes $pa(X)$ and the external noise U_X by a production function f_X as $X = f_X(pa(X), U_X)$. By abusing notation, we also denote the induced probability measure of X by the same symbol f_X. See Appendix C.1 for details.

Our target parameter is the counterfactual mean of the final outcome Y under a user-specified dynamic treatment policy $g = [g_t]_{t=1}^{\tau}$ where g_t is a probability measure on the treatment space conditioned on the whole history, $pa(A_t) = (L_{1:t}, A_{1:t-1}, Y_{1:t-1})$ up until A_t (not including A_t). Specifically, our target parameter is given by

$$\psi(P) = \mathbb{E}Y^g, \quad (2)$$

which is the mean of the counterfactual outcome produced by replacing π, defined as the behavior treatment policy observed on the data, with g in the structural causal model.

Identification   Under the positivity assumption:

$$g \ll \pi, \quad (3)$$

and the sequential randomization assumption:

$$Y^g \perp A_t \mid pa(A_t) \text{ for } t = 1, \ldots, \tau, \quad (4)$$

we can identify our target causal parameter through g-formula as the mean of Y under the counterfactual distribution which is given by replacing distributions π with g (Robins, 1986):

$$\mathbb{E}Y^g = \mathbb{E}_g Y. \quad (5)$$

Note that the consistency assumption $Y = Y^\pi$, usually stated in causal inference literature, is a consequence of the definition of counterfactual outcome in our SCM. Now the problem is reduced to the estimation of the statistical parameter:

$$\psi(P) = \mathbb{E}_g Y. \quad (6)$$

3.3. Targeted Minimum Loss-based Estimation

Given we have an estimator $\hat{P}_n$ of the data-generating distribution P_0, a natural estimator of the target functional is the plug-in estimator $\hat{\psi}_n = \psi(\hat{P}_n)$. Under a regularity condition, ψ admits the following first-order expansion

$$\psi(\hat{P}_n) - \psi(P_0) = -\int_{\mathcal{O}} D^\star(\hat{P}_n) dP_0 + R_2(\hat{P}_n, P_0), \quad (7)$$

where $D^\star$ is the efficient influence function of ψ, and $R_2(\hat{P}_n, P_0)$ is the exact remainder. Influence functions quantifies the amount of changes of an estimator under small perturbations of the sample. The efficient influence function is the influence function with minimal variance. The idea of TMLE is to eliminate the empirical analogue of the first term of the right hand side by fluctuating $\hat{P}_n$ to find a distribution $\hat{P}_n^\star$ with $P_n D^\star(\hat{P}_n^\star) = 0$, where $Pf = \int f dP$ is a shorthand for the expectation of a measurable function f with respect to a probability measure P. Our problem is to obtain an initial estimate $\hat{P}_n$ with a potentially large scale and high dimensional longitudinal data, and correct bias of the plug-in estimator $\psi(\hat{P}_n)$ by fluctuating $\hat{P}_n$.

4. Proposed Method

In this section, we describe our proposed method, Deep Longitudinal Targeted Minimum Loss-based Estimation (Deep LTMLE). Let

$$Q_t(pa(Y_t)) = \mathbb{E}_g[Y_\tau \mid L_{1:t}, A_{1:t}, Y_{1:t-1}] \quad (8)$$

be the mean outcome at the end of the observation period τ given the history before node Y_t for $t = 1, \ldots, \tau$, where future treatments $A_{t+1}, \ldots, A_\tau$ follow treatment assignment policy g. Similarly,

$$V_t(pa(A_t)) = \mathbb{E}_g[Y_\tau \mid L_{1:t}, A_{1:t-1}, Y_{1:t-1}] \quad (9)$$

is the mean outcome at the time τ given the history before node A_t, for $t = 1, \ldots, \tau$. We abbreviate Q_t for $Q_t(pa(Y_t))$

Algorithm 1 Temporal Difference Learning of Conditional Counterfactual Mean Outcomes

1: **for** $b = 1$ **to** B **do**
2:     $\hat{Q}_T(pa(Y_T)) \leftarrow \hat{Q}_T(pa(Y_T)) - \alpha \cdot \partial_{\hat{Q}_T(pa(Y_T))} \mathcal{L}(\hat{Q}_T(pa(Y_T)),\ Y_T)$
3:     $\hat{\pi}_T \leftarrow \hat{\pi}_T - \alpha \cdot \partial_{\hat{\pi}_T(pa(A_T))} \mathcal{L}(\hat{\pi}_T(pa(A_T)),\ A_T)$
4:     **for** $t = T - 1$ **to** 1 **do**
5:         $\hat{V}_{t+1}(pa(A_{t+1})) \leftarrow \mathbb{E}_{A_{t+1} \sim g_{t+1}(pa(A_{t+1}))} \big[\hat{Q}_{t+1}(pa(A_{t+1}), A_{t+1}) \big]$
6:         $\hat{Q}_t(pa(Y_t)) \leftarrow \hat{Q}_t(pa(Y_t)) - \alpha \cdot \partial_{\hat{Q}_t(pa(Y_t))} \mathcal{L}(\hat{Q}_t(pa(Y_t)),\ \hat{V}_{t+1}(pa(A_{t+1})))$
7:         $\hat{\pi}_t \leftarrow \hat{\pi}_t - \alpha \cdot \partial_{\hat{\pi}_t(pa(A_t))} \mathcal{L}(\hat{\pi}_t(pa(A_t)),\ A_t)$
8:     **end for**
9: **end for**
10: Output $(\hat{Q}_t, \hat{V}_t, \hat{\pi}_t)_{t=1}^T$

---

if it is clear from the context, similarly for V_t. Our goal is to estimate $\psi(P)$ by

$$
\begin{aligned}
\hat{\psi}_n^\star &= P_n \hat{V}_{1,\varepsilon^\star}(pa(A_1)) \\
&= P_n \mathbb{E}_{A_1 \sim g_1(pa(A_1))}[\hat{Q}_{1,\varepsilon^\star}(pa(A_1), A_1)] \quad (10)
\end{aligned}
$$

where $\hat{Q}_{1,\varepsilon^\star}$ is an estimation of Q_1 such that $\hat{\psi}_n^\star$ is asymptotically efficient. We achieve this by proposing a temporal-difference heterogeneous transformer to yield an initial estimation $\hat{Q}_1$, then update this estimation to get $\hat{Q}_{1,\varepsilon^\star}$ via Targeted Minimum Loss-based Estimation (TMLE). We denote the stopping time by T when we observe failure or censoring, or the trajectory reached the final time τ.

4.1. Temporal-Difference Heterogeneous Transformer

To learn the initial model $\hat{V}_1$, we use temporal-difference loss as the objective to learn underlying models $\hat{Q}_t$ for $t = 1, \ldots, \tau$ via stochastic gradient descent (SGD). The principle of temporal difference learning (Sutton, 1988; Mnih et al., 2013) is to supervise $\hat{Q}_t$ to obey the temporal equality of Q_t:

$$
\begin{aligned}
Q_t &= \mathbb{E}_{p(Y_t, L_{t+1} | pa(Y_t))} V_{t+1} \\
&= \mathbb{E}_{p(Y_t, L_{t+1} | pa(Y_t)),\ g_{t+1}(pa(A_{t+1}))} Q_{t+1} \quad (11)
\end{aligned}
$$

for $t = 1, \ldots, \tau - 1$ and $Q_\tau = \mathbb{E}_{p(Y_\tau | pa(Y_\tau))} Y_\tau$. The temporal difference loss on a sample trajectory is thus given by

$$
\mathcal{L}_t^Q = \mathcal{L}(\hat{Q}_t, \hat{V}_{t+1}) = \mathcal{L}(\hat{Q}_t, \mathbb{E}_{g_{t+1}(pa(A_{t+1}))} \big[\hat{Q}_{t+1} \big]) \quad (12)
$$

for $t = 1, \ldots, T - 1$, where $\mathbb{E}_{g_{t+1}(pa(A_{t+1}))} \big[\hat{Q}_{t+1} \big]$ can be computed by Monte-Carlo estimation if g is a density function, and $\mathcal{L}_T^Q = \mathcal{L}(\hat{Q}_T, Y_T)$. In the case of survival analysis, the components for $t = 1, \ldots, \tau - 1$ are defined as $\mathcal{L}_t^Q = \mathcal{L}_{\text{bce}}(\hat{Q}_t, \hat{V}_{t+1})$, where $\mathcal{L}_{\text{bce}}(\hat{y}, y) = -\big[y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \big]$ is the binary cross entropy loss. To yield the updated model $\hat{Q}_{t,\varepsilon}$, we need to adjust $\hat{Q}_t$ post

training to factor in estimations of propensity scores $e_t(a) = \pi_t(a | pa(A_t))$, which we will describe in detail in the next section. Hence, the loss function also includes

$$
\mathcal{L}_t^e = \mathcal{L}_{\text{bce}}(\hat{\pi}_t, A_t) \quad (13)
$$

for estimating model $\hat{\pi}_t$ and is thus given by

$$
\mathcal{L} = \sum_{t=1}^\tau \mathbb{1}\{t \le T\} \mathcal{L}_t = \sum_{t=1}^T \mathcal{L}_t^Q + \alpha \mathcal{L}_t^e \quad (14)
$$

where α is a scaling hyperparameter. See Algorithm 1 for the optimization workflow. Convergence of the algorithm can be found in Appendix D.

For the estimation of Q_t and π_t, we propose a unified model architecture to simultaneously optimize $\hat{Q}_t$ and $\hat{\pi}_t$ in an efficient, non-sequential manner by adapting a decoder-only Transformer (Vaswani et al., 2017; Brown et al., 2020) to longitudinal data with heterogeneous tokens. An overview of the model architecture is given in Figure 1. For each sampled sequence in the training set, we feed each token in the sequence to an embedding layer $h_\bullet$, which can be type-specific linear layers as in h_W, h_L, h_A, h_Y, or simply a shared layer $h_\bullet = h$ padding each token to the same length for a more efficient and parallelizable embedding operation. Then, each embedding is aggregated with its positional encoding $H_0, \ldots, H_\tau$ and type encoding H_W, H_L, H_A, and H_Y that represent its timestamp and variable type information:

$$
H(\bullet_t) = \texttt{concat}\ \{h_\bullet(\bullet_t), H_\bullet, H_t\} \quad (15)
$$

for $\bullet_t \in (W_0, L_1, A_1, Y_1, \ldots, L_\tau, A_\tau, Y_\tau)$. Note that we include type embedding $H_\bullet$ because $h_\bullet$ need not necessarily be type-specific. Then, the embedded sequence is fed into the transformer and produce $\hat{\pi}$ and $\hat{Q}$ through output layers f_e and f_Q at each position that corresponds to token type L and A respectively:

$$
\hat{\pi}_t = f_e(\text{transformer}\ \{H(W_0),\ \ldots,\ H(L_t)\}) \quad (16)
$$

$$
\hat{Q}_t = f_Q(\text{transformer}\ \{H(W_0),\ \ldots,\ H(A_t)\}). \quad (17)
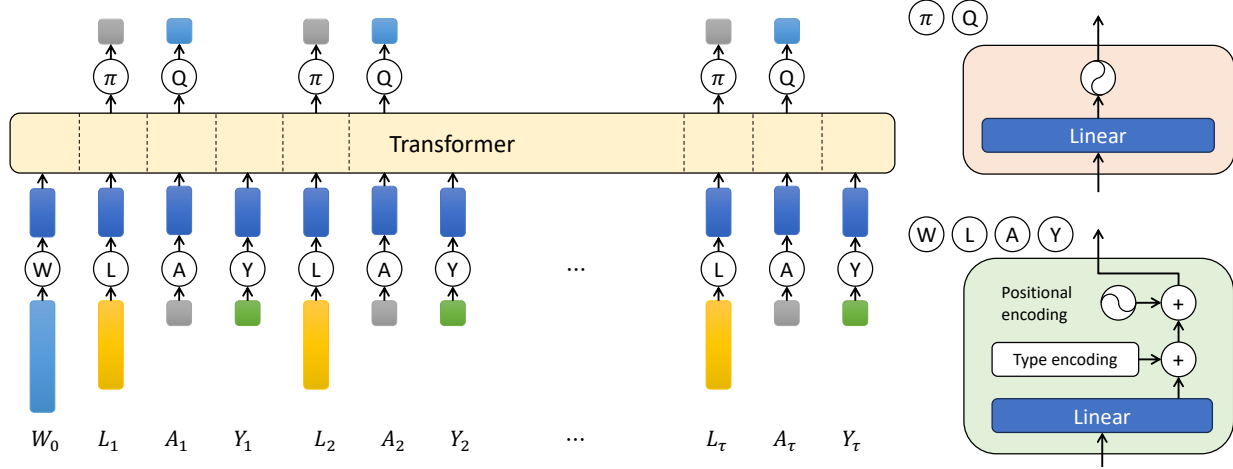$$

Figure 1. **Architecture of temporal-difference heterogeneous-token transformer (TDHT)**. Observed variables are fed into transformer after embedding layers depending on the variable types. Embedding layers aggregate linear transform with learnable type encoding and learnable positional encoding. Outputs of the transformer are π after L and Q after A. Each output head consists of a linear layer and the final activation function depending on variable distribution (sigmoid for binary, softmax for categorical and none for continuous). The outputs of π heads are used to learn propensity scores and the outputs of Q heads are used for temporal-difference learning after integration with respect to the counterfactual treatment policy.

In practice, we can use a joint output layer f for f_e and f_Q for more efficient and parallelizable output generation, where the output number of dimensions is the sum of the number of dimensions $\dim(A)$ for treatment A and $\dim(Y)$ for outcome Y. Then, we compute softmax probabilities masking out the last $\dim(Y)$ dimensions for $\hat{\pi}_t$ and first $\dim(A)$ dimensions for $\hat{Q}_t$.

Our proposed architecture does not entail concatenation of variables at the same timestamp or sequential decoding of outputs following the transformer blocks like prior work Melnychuk et al. (2022), which 1) allows us to handle different types of and different number of variables at different timestamps (e.g. starting from W_0, ending at L_8, while A_3 and Y_3 are missing), and 2) is fully parallelizable when we use padding instead of learnable linear mapping for the embedding layer $h_\bullet$ and use the joint output layer f.

4.2. Targeted Minimum Loss-based Estimation

Efficient Influence Function Since our target parameter is the counterfactual mean outcome at the final τ, the relevant part of P_0 of interest are $Q_{t,0}$ for $t = 1, 2, ..., \tau$.

Theorem 4.1. *In our counterfactual mean case, the efficient influence function $D^\star(P)(O) = D^\star(\{Q_t, \pi_t\}_{t=1}^\tau)(O)$ is given by*

$$D^\star(P)(O) = (V_1 - \psi_0) + \sum_{t=1}^{T} I_t(V_{t+1} - Q_t) \quad (18)$$

where $I_t = \prod_{s=1}^{t} dg_s/d\pi_s$ and $V_{T+1} = Y_T$.

This is given in (van der Laan & Gruber, 2012).

4.2.1. TEMPORAL DIFFERENCE TARGETING

Submodel We update the initial estimate $\hat{Q}_t$ for $Q_{t,0}$ to $\hat{Q}_t^{g\star}$ such that $P_n D^\star(\{\hat{Q}_t^{g\star}, \hat{\pi}_t\}_{t=1}^\tau) = 0$. We realize this by fluctuating $\hat{Q}_t$ along a one-dimensional submodel through the initial fit $\hat{Q} = [\hat{Q}_t]_{t=1}^\tau$ given by, $\hat{Q}_\varepsilon = [\hat{Q}_{t,\varepsilon}]_{t=1}^\tau$, where

$$\text{logit} \, \hat{Q}_{t,\varepsilon} = \text{logit} \, \hat{Q}_t + \varepsilon \quad (19)$$

with a common fluctuation parameter ε across t. If the outcome is survival, then we automatically have $Y_t \in [0, 1]$. In a general longitudinal setting for bounded Y_t's, we can re-scale both Y_t and $\hat{Q}_t$ to $[0, 1]$ and use the same one-dimensional submodel.

Partial Loss function We search for the optimal fluctuation $\varepsilon^\star$ with respect to the partial loss function

$$\mathcal{L}^\star(\hat{Q}_\varepsilon, \hat{V}_{\varepsilon'}; \hat{\pi}) = \sum_{t=1}^{T} I_t(\hat{\pi}) \mathcal{L}_{\text{bce}}(\hat{Q}_{t,\varepsilon}, \hat{V}_{t+1,\varepsilon'}), \quad (20)$$

where $\hat{V}_{T+1,\varepsilon} = Y_T$ and $\hat{V}_{t,\varepsilon} = \mathbb{E}_{A_t \sim g_t}[\hat{Q}_{t,\varepsilon}]$, such that $\mathcal{L}^\star(\hat{Q}_\varepsilon, \hat{V}_{\varepsilon'})$ satisfies the following theorem:

Theorem 4.2. *For any $\varepsilon^\star$, we have*

$$\partial_\varepsilon|_{\varepsilon^\star} \mathcal{L}^\star(\hat{Q}_\varepsilon, \hat{V}_{\varepsilon^\star}) = D^\star(\hat{Q}_{\varepsilon^\star}, \hat{\pi}). \quad (21)$$

See Section B for the proof.

Corollary 4.3. *Suppose that we found an $\varepsilon^\star$ satisfying*

$$\partial_\varepsilon|_{\varepsilon^\star} P_n \mathcal{L}^\star(\hat{Q}_\varepsilon, \hat{V}_{\varepsilon^\star}) = 0, \tag{22}$$

then $\hat{Q}_{\varepsilon^\star}$ solves the efficient influence function.

In practice, for the finite sample performance, we only need to solve it to the order of standard error with a $\sigma_n / \log n$ factor (van der Laan & Rose, 2011) (Algorithm 2).

Algorithm 2 Temporal-Difference Targeting

1: $\varepsilon \leftarrow 0$
2: **repeat**
3:     $\varepsilon \leftarrow \text{argmin}_{\varepsilon'} P_n \mathcal{L}^\star(\hat{Q}_{\varepsilon'}, \hat{V}_\varepsilon)$.
4:     $\hat{\sigma}_n \leftarrow \sqrt{n^{-1} P_n D^{\star 2}(\hat{Q}_\varepsilon, \hat{\pi})}$
5: **until** $P_n D^\star(\hat{Q}_\varepsilon^g, \hat{\pi}) < \hat{\sigma}_n / \log n$
6: $\hat{\psi}_n^\star \leftarrow P_n \hat{V}_{1,\varepsilon}(pa(A_1))$
7: 95% CI as $\hat{\psi}_n^\star \pm 1.96 \cdot \hat{\sigma}_n$

Convergence of Algorithm 2 The investigation of $\mathcal{L}_{\text{bce}}(\hat{Q}_{t,\varepsilon}, \hat{V}_{t,\varepsilon})(O)$ for different t and O's as a function of ε suggests that they admit different bell curve shapes concentrating at different ε's and have different spread out levels. Thus, by summing up $\mathcal{L}_{\text{bce}}(\hat{Q}_{t,\varepsilon}, \hat{V}_{t,\varepsilon})(O)$ across t and across O's as $P_n \mathcal{L}^\star(\hat{Q}_\varepsilon, \hat{V}_\varepsilon)$ as a function of ε will fluctuate a lot and we expect a local minima and local maxima around the neighborhood of $\varepsilon = 0$. And thus the convergence of the algorithm is highly probable and we don't discover any issue in our simulations.

Comparison to LTMLE In the LTMLE, we only need a good estimate of Q_τ and then do backward sequential regression and targeting as mentioned in (van der Laan & Gruber, 2012). However, the problem is the error in the estimation of Q_τ can propagate as we progress back to get $Q_{\tau-1}^*, ..., Q_1^*$. Nonetheless, after our initial transformer step, we have good initial estimates for all $Q_1, ..., Q_\tau$. So, instead of only relying on a good estimate of Q_τ, our algorithm makes uses of all of them. and doing targeting across t with $o(\varepsilon)$ fluctuation at each t level. Thus, we are able to pool information across time when doing the targeting step.

4.2.2. SEQUENTIAL TARGETING

Alternatively, one could apply a sequential targeting procedure that is very similar to LTMLE but with given initials generated from the transformer step.

Submodel We fluctuate each component of the initial fit $\hat{Q} = [\hat{Q}_t]_{t=1}^\tau$ along a model as

$$\text{logit } \hat{Q}_{t,\varepsilon_t} = \text{logit } \hat{Q}_t + \varepsilon_t. \tag{23}$$

Loss Function Starting from $t = \tau$, given we have found $\varepsilon_{t+1}^\star$, among individuals whose $T > t - 1$, we search for empirical loss minimizer $\varepsilon_t^\star$ with respect to the loss function $\mathcal{L}_t^\star$ as,

$$\mathcal{L}_t^\star(\hat{Q}_{t,\varepsilon_t}, \hat{V}_{t+1,\varepsilon_{t+1}^\star}) = I_t(\hat{\pi}) \mathcal{L}_{\text{bce}}(\hat{Q}_{t,\varepsilon_t}, \hat{V}_{t+1,\varepsilon_{t+1}^\star}), \tag{24}$$

where $\hat{V}_{t+1,\varepsilon_{t+1}^\star} = \mathbb{E}_{A_{t+1} \sim g_{t+1}}[\hat{Q}_{t+1,\varepsilon_{t+1}^\star}]$ when $T > t$ and $\hat{V}_{t+1,\varepsilon_{t+1}^\star} = Y_T$ when $T = t$. To initialize, we set $\varepsilon_{\tau+1}^\star = 0$.

Lemma 4.4. *Suppose that we found $\varepsilon_\tau^\star, ...\varepsilon_1^\star$ sequentially as mentioned above, then $\{\hat{Q}_{\varepsilon_t^\star}\}_{t=1}^\tau$ solves the efficient influence function.*

Algorithm 3 Sequential Targeting

1: $\varepsilon_{\tau+1}^\star \leftarrow 0$
2: **for** $t = \tau$ **to** 1 **do**
3:     $\varepsilon_t^\star \leftarrow \text{argmin}_{\varepsilon_t} P_n \mathbb{1}\{T \geq t\} \mathcal{L}^\star(\hat{Q}_{t,\varepsilon_t}, \hat{V}_{t+1,\varepsilon_{t+1}^\star})$.
4: **end for**
5: $\hat{\psi}_n^\dagger \leftarrow P_n \hat{V}_{1,\varepsilon_1^\star}(pa(A_1))$
6: $\hat{\sigma}_n \leftarrow \sqrt{n^{-1} P_n D^{\star 2}(\{\hat{Q}_{\varepsilon_t^\star}\}_{t=1}^\tau, \hat{\pi})}$
7: 95% CI as $\hat{\psi}_n^\dagger \pm 1.96 \cdot \hat{\sigma}_n$

Comparaison to LTMLE While the error can still propagate as we move back in time, the error propagates only through the targeting steps whereas in LTMLE the error can also propagate through regressions. At each time step t, LTMLE needs to first regress $\hat{V}_{t+1,\varepsilon_{t+1}^\star}$ on $pa(Y_t)$ to get an estimate $\hat{Q}_t$ and then perform the targeting through the submodel in(23). However, we only use initial estimate $\hat{Q}_t$ from our transformer fit and it does not depend on $\varepsilon_{t+1}^\star, \ldots, \varepsilon_\tau^\star$.

Why not targeting through additional loss function As in DeepACE, the targeting can be performed through introducing additional loss components to further train the transformer we have build in the first step. This additional loss function will have its derivative equal to the efficient influence function. However, we find that the penalty factor before this loss function is hard to tune and in near all cases, it is hard to guarantee the EIF is solved and most of the time we will hurt our initial fits as shown in Appendix 5.1.

5. Experiments

We conducted two experiments. In the first experiment, we compare the bias, root-mean-squared-error (RMSE), and coverage probability, of our estimator with existing estimators based on 100 times of estimations for both continuous and survival outcomes. The second experiment is an application of our proposed method to a real-world data.
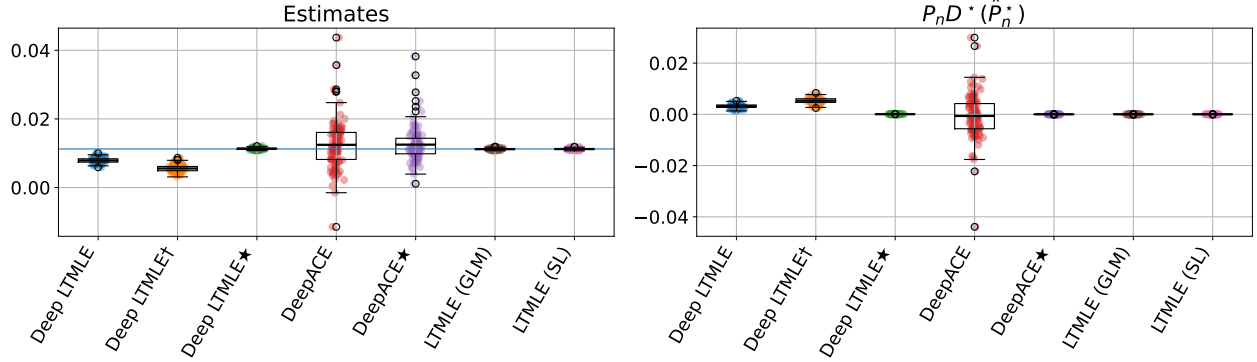
Figure 2. **Results from simple synthetic data with continuous outcome.** Left: Sampling distributions of estimates. Right: Sampling distributions of empirical means of estimated efficient influence functions. Deep LTMLE: initial estimate with TDHT; Deep LTMLE†: TDHT with sequential targeting; Deep LTMLE⋆: TDHT with temporal-difference targeting; DeepACE⋆: DeepACE with temporal-difference targeting; LTMLE (GLM): LTMLE with GLM; LTMLE (SL): LTMLE with super learner of GLM, MARS, and XGBoost.

Table 1. Results from complex synthetic data.

Model	Bias			RMSE			Coverage			Mean $\hat{\sigma}_n$		
	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$
LTMLE (GLM)	0.0230	0.0766	0.1344	0.0265	0.0796	0.1381	1.00	1.00	1.00	0.43	0.69	0.76
LTMLE (SL)	0.0144	0.0297	0.0477	0.0185	0.0344	0.0545	1.00	1.00	1.00	0.31	0.40	0.45
DeepACE	0.0055	-0.0211	-0.0714	0.0397	0.0672	0.0962	1.00	1.00	1.00	0.69	0.76	0.73
Deep LTMLE	0.0181	0.0292	0.0503	0.0263	0.0332	0.0533	0.99	0.96	0.75	0.17	0.10	0.06
Deep LTMLE†	0.0319	0.0327	0.0535	0.0571	0.0379	0.0571	0.99	0.97	0.78	0.19	0.10	0.07
Deep LTMLE⋆	0.0156	0.0307	0.0496	0.0218	0.0344	0.0541	0.99	0.95	0.78	0.17	0.09	0.07

5.1. Synthetic Data with Continuous Outcome

First, we start our experiment with a very simple data generating process with continuous outcome, $n = 500$, and $\tau = 10$. The data generating proccess is described in the Section G.1. After fitting DeepACE, we additionally performed our targeting precedures on the fit.

The results were shown in Figure 2. Initial fits of Deep LTMLE and DeepACE had comparable bias. Even with the targeting loss, DeepACE failed to solve the efficient influence function. On the other hand, due to the separation of the targeting step in our method, we managed to solve it completely and succeeded in correcting bias.

5.2. Synthetic Data with Survival Outcome

Next, we evaluated Deep LTMLE under a highly complex data-generating process with survival outcomes, five-dimensional time-dependent covariates, non-Markovian dependencies, $n = 1000$, and $\tau = 10, 20, 30$, imitating the setups from previous studies (Bica et al., 2020; Frauen et al., 2023). See Section G.3 for details.

Results are presented in Table 1. We observe that Deep LTMLE and LTMLE with a super learner on average

achieves a lower RMSE compared to other methods, particularly in scenarios with larger τ, indicating its robustness in complex and realistic scenarios without Markovian dependencies. Benefits by our temporal difference targeting procedures are obvious for $\tau = 10$. While Deep LTMLE's coverage probability diminished at $\tau = 30$, the confidence intervals generated by LTMLE and DeepACE were notably over-conservative with large estimated standard errors.

The pronounced bias of DeepACE can likely be attributed to three factors. First, DeepACE's use of the squared-error-loss for the outcome is known to induce greater bias in sparse outcomes, a common scenario in survival analysis, as opposed to the logistic loss used in our approach (Gruber & van der Laan, 2010). Second, DeepACE failed to solve the efficient influence function. Third, DeepACE does not account for the degeneration of the survival outcome.

Simple Synthetic Data with Survival Outcome We also conducted an eperiment with a very simple survival synthetic data with one-dimensional time-dependent covariates, $n = 1000$, and $\tau = 10, 20, 30$. Although LTMLE with GLM is expected to have strong performance in this experiment, Deep LTMLE remains highly competitive in this context, equalling LTMLE's performance (Section H).

Table 2. Results from semi-synthetic data with unmeasured confounding

Model	Bias			RMSE			Coverage			Mean $\hat{\sigma}_n$		
	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$
LTMLE (SL)	0.0075	0.0341	0.0574	0.0138	0.0491	0.0786	0.70	0.45	0.25	0.09	0.12	0.14
DeepACE	-0.0174	-0.0434	-0.0770	0.0788	0.1154	0.1341	1.00	1.00	1.00	0.67	0.78	0.86
Deep LTMLE	-0.0002	0.0108	0.0041	0.0162	0.0720	0.0772	1.00	0.95	1.00	0.18	0.27	0.32

5.3. Semi-Synthetic Data

To evaluate the performance of the proposed methods, we generated realistic data from Circulatory Risk in Communities Study (CIRCS) (Yamagishi et al., 2019), a long-term on-going cardiovascular epidemiological cohort study, lasting over a half century. See Section H.1 for the detail.

Table 2 shows the results with semi-synthetic data with unmeasured confounding, which reflects a real world setting. Deep LTMLE performed best in terms of bias for all time horizons. Furthermore, as the time horizon increases from 10 to 30, LTMLE's coverage probability drops as low as 0.25. On the other hand, Deep LTMLE has nominal coverage even in the longest time-horizon setting.

5.4. Real World Data

We applied Deep LTMLE to real world data from CIRCS. We estimated the counterfactual mean outcomes under the standard blood pressure (SBP) management strategy that controls SBP less than 140 mmHg and the intensive blood pressure management strategy with SBP less than 120 mmHg after the 30 years of sustained management.

In real world applications, we often encounter with practical problems of censoring, that is loss of follow-up for some reasons. Our model can be easily generalized to cover this setting with a slight modification by adding censoring nodes. Details are described in Section I of Appendix.

The results were shown in Figure 3. The average treatment effect (ATE) of the intensive management strategy over the standard management strategy first increased with a peak at 20 years after baseline and then decreased with a fluctuation. The direction and trend of ATE is consistent with the difference of empirical means of cumulative outcomes between two groups followed the two strategies.

5.5. Computation Details

DeepACE and Deep LTMLE were run on a GPU (Tesla T4) with 16 GB memory and LTMLE on CPU (Intel Xeon Skylake 6230 @ 2.1 GHz) with 40 cores and 96 GB memory. We used the R package `ltmle` with GLM and a super learner (SL) library consisting of GLM, maltivariate adaptive regression spline with `earth` package, and `xgboost` for the simple synthetic data and the real world data (Lendle
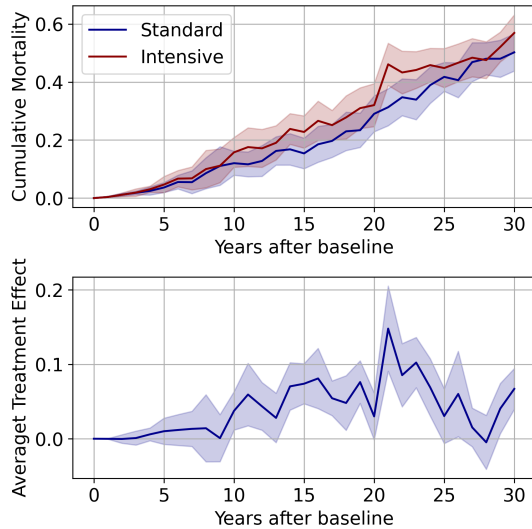

Figure 3. **Counterfactual mean outcomes and 95% simultaneous confidence intervals according to standard and intensive treatment policies among 13,485 participants of the CIRCS.**

et al., 2017; Polley et al., 2021; Milborrow, 2023; Chen et al., 2022). Confidence intervals for LTMLE was constructed based on its estimate of the efficient influence function.

Table 3. Running time with complex synthetic data

Model	Time, sec		
	$\tau = 10$	$\tau = 20$	$\tau = 30$
LTMLE (SL)	271	958	2122
DeepACE	53	54	133
Deep LTMLE	38	39	116

As shown in Table 3, Deep LTMLE leverages GPU acceleration to achieve significantly faster processing times than LTMLE, presenting a substantial computational benefit for analyses involving extensive time horizons and high-dimensional time-dependent covariates. Code for experimens with synthetic data is available at https://github.com/shirakawatoru/dltmle-icml-2024.

6. Discussion

Our method assumes the sequential randomization and the positivity assumption on the intervention mechanism to identify the counterfactual outcome from observational data. However, to our surprise, in semi-synthetic data simulations, we found that when there is unmeasured confounding violating the sequential randomization assumption rely on, our method is very robust and could even provide robust inference. Furthermore, our proposed model does not currently address several complexities often found in real-world data, such as visiting processes, competing risks, and continuous time horizons. These challenges will be the focus of our future research efforts.

7. Conclusion

In this paper, we propose a variant of LTMLE that leverages the sequential learning capabilities of transformers. This approach enables simultaneous fitting of the entire LTMLE, allowing us to target the mean survival under dynamic interventions directly through weighting the loss function with cumulative inverse probabilities of intervention. The proposed method performs competitively with asymptotically efficient estimators in low-dimensional settings and exceeds the performance of existing models in high-dimensional scenarios. Scalability of our model to larger and longer datasets was implied. We applied our method to real world data and demonstrated a causal inference on the effect of sustained blood pressure management strategies on total mortality.

Acknowledgement

Impact Statement

The present study may have an impact on generating evidence for medical practice. Since the majority of medical evidence on the effect of treatments such as blood pressure control on the incidence of diseases or mortality are based on a single time point intervention study, the present study which takes multiple point intervention and long past patient histories into account might trigger the rethinking of the methodology for evidence synthesis and potentially affect the medical practice in the future after thorough statistically rigorous evaluation of the method and analyses of both synthetic and real world data.

References

Bang, H. and Robins, J. M. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61 (4):962–973, 2005.

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

Bickel, P., Klaassen, C., Ritov, Y., and Wellner, J. *Efficient and Adaptive Estimation for Semiparametric Models*. Johns Hopkins Series in the Mathematical Sciences. Springer New York, 1993. ISBN 978-0-387-98473-5.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., and Yuan, J. *xgboost: Extreme Gradient Boosting*, 2022. URL `https://CRAN.R-project.org/package=xgboost`. R package version 1.7.6.1.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Chernozhukov, V., Escanciano, J. C., Ichimura, H., Newey, W. K., and Robins, J. M. Locally robust semiparametric estimation. *Econometrica*, 90(4):1501–1535, 2022.

Dahabreh, I. J. and Bibbins-Domingo, K. Causal Inference About the Effects of Interventions From Observational Studies in Medical Journals. *JAMA*, May 2024. ISSN 1538-3598. doi: 10.1001/jama.2024.7741.

Dang, L. E., Gruber, S., Lee, H., Dahabreh, I. J., Stuart, E. A., Williamson, B. D., Wyss, R., Díaz, I., Ghosh, D., Kıcıman, E., Alemayehu, D., Hoffman, K. L., Vossen, C. Y., Huml, R. A., Ravn, H., Kvist, K., Pratley, R., Shih, M.-C., Pennello, G., Martin, D., Waddy, S. P., Barr, C. E.,

Akacha, M., Buse, J. B., van der Laan, M., and Petersen, M. A causal roadmap for generating high-quality real-world evidence. *Journal of Clinical and Translational Science*, 7(1):e212, 2023.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. More robust doubly robust off-policy evaluation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1447–1456. PMLR, 10–15 Jul 2018.

Frauen, D., Hatt, T., Melnychuk, V., and Feuerriegel, S. Estimating average causal effects from patient trajectories. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7586–7594, 2023.

Gruber, S. and van der Laan, M. J. A targeted maximum likelihood estimator of a causal effect on a bounded continuous outcome. *The International Journal of Biostatistics*, 6(1):Article 26, 2010. ISSN 1557-4679. doi: 10.2202/1557-4679.1260.

Gruber, S. and van der Laan, M. J. Targeted minimum loss based estimation of a causal effect on an outcome with known conditional bounds. *The international journal of biostatistics*, 8(1):21–21, 2012. ISSN 1557-4679.

Hernán, M. A., Wang, W., and Leaf, D. E. Target Trial Emulation: A Framework for Causal Inference From Observational Data. *JAMA*, 328(24):2446–2447, December 2022. ISSN 1538-3598. doi: 10.1001/jama.2022.21383.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 652–661, New York, New York, USA, 20–22 Jun 2016. PMLR.

Kallus, N. and Uehara, M. Double reinforcement learning for efficient and robust off-policy evaluation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5078–5088. PMLR, 13–18 Jul 2020.

Kennedy, E. H. Semiparametric doubly robust targeted double machine learning: A review. *arXiv preprint arXiv:2203.06469*, 2022.

Klaassen, C. A. J. Consistent estimation of the influence function of locally asymptotically linear estimators. *The Annals of Statistics*, 15(4):1548–1562, 1987.

Lendle, S. D., Schwab, J., Petersen, M. L., and van der Laan, M. J. ltmle: An R package implementing targeted minimum loss-based estimation for longitudinal data. *Journal of Statistical Software*, 81(1):1–21, 2017. doi: 10.18637/jss.v081.i01.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643*, 2020.

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. G-net: A recurrent network approach to G-computation for counterfactual prediction under a dynamic treatment regime. In *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pp. 282–299. PMLR, 2021.

Melnychuk, V., Frauen, D., and Feuerriegel, S. Causal transformer for estimating counterfactual outcomes. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15293–15329. PMLR, 2022.

Milborrow, S. *earth: Multivariate Adaptive Regression Splines*, 2023. URL https://CRAN.R-project.org/package=earth. R package version 5.3.2.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Narita, Y., Yasui, S., and Yata, K. Debiased off-policy evaluation for recommendation systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, pp. 372–379, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384582.

Petersen, M. L. and van der Laan, M. J. Causal models and learning from data: Integrating causal modeling and statistical estimation. *Epidemiology (Cambridge, Mass.)*, 25(3):418–426, 2014.

Polley, E., LeDell, E., Kennedy, C., and van der Laan, M. *SuperLearner: Super Learner Prediction*, 2021. URL https://CRAN.R-project.org/package=SuperLearner. R package version 2.0-28.1.

Robins, J. 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, 1986.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. Estimation of Regression Coefficients When Some Regressors Are Not Always Observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.

Salerno, S. and Li, Y. High-dimensional survival analysis: Methods and applications. *Annual review of statistics and its application*, 10:25–49, 2023.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

Tran, L., Yiannoutsos, C., Wools-Kaloustian, K., Siika, A., van der Laan, M., and Petersen, M. Double robust efficient estimators of longitudinal treatment effects: Comparative performance in simulations and a case study. *The International Journal of Biostatistics*, 15(2), 2019. ISSN 1557-4679. doi: 10.1515/ijb-2017-0054.

van der Laan, M. and Rubin, D. Targeted Maximum Likelihood Learning. *The International Journal of Biostatistics*, 2(1), 2006.

van der Laan, M. J. and Gruber, S. Targeted Minimum Loss Based Estimation of Causal Effects of Multiple Time Point Interventions. *The International Journal of Biostatistics*, 8(1), 2012.

van der Laan, M. J. and Robins, J. *Unified Methods for Censored Longitudinal Data and Causality*. Springer Series in Statistics. Springer New York, 2003. ISBN 978-0-387-21700-0.

van der Laan, M. J. and Rose, S. *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Series in Statistics. Springer, 2011. ISBN 978-1-4419-9781-4 978-1-4419-9782-1.

van der Laan, M. J. and Rose, S. *Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies*. Springer Series in Statistics. Springer International Publishing, 2018.

van der Laan, M. J., Polley, E. C., and Hubbard, A. E. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1):1309–1309, 2007. ISSN 1544-6115.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wyss, R., Yanover, C., El-Hay, T., Bennett, D., Platt, R. W., Zullo, A. R., Sari, G., Wen, X., Ye, Y., Yuan, H., Gokhale, M., Patorno, E., and Lin, K. J. Machine learning for improving high-dimensional proxy confounder adjustment in healthcare database studies: An overview of the current literature. *Pharmacoepidemiology and drug safety*, 31(9): 932–943, 2022. ISSN 1053-8569.

Yamagishi, K., Muraki, I., Kubota, Y., Hayama-Terada, M., Imano, H., Cui, R., Umesawa, M., Shimizu, Y., Sankai, T., Okada, T., Sato, S., Kitamura, A., Kiyama, M., and Iso, H. The Circulatory Risk in Communities Study (CIRCS): A Long-Term Epidemiological Study for Lifestyle-Related Disease Among Japanese Men and Women Living in Communities. *Journal of Epidemiology*, 29(3):83–91, 2019.

A. Notation

Here we list notations used in the article.

O	Observed variables $O = (W = W_0, L_1, A_1, Y_1, L_2, A_2, Y_2, \ldots, L_\tau, A_\tau, Y_\tau)$
τ	Maximum length of time-horizon
T	Stopping time
W	Baseline covariates
L_t	Time-dependent covariates (states)
A_t	Time-dependent treatments (controls)
Y_t	Outcomes. In survival case, binary failure indicator defined as $Y_t = \mathbb{1}\{T \le t\}$
Y	Outcome at the end of the trajectory: $Y = Y_T$
$pa(\bullet)$	Parent nodes of a node $\bullet$. For example, $pa(L_t) = (W, L_{1:t-1}, A_{1:t-1}, Y_{1:t-1})$
P_0	The true distribution of the observed variable
$\hat{P}_n$	Estimator of P_0
π	Propensity scores $\pi = [\pi_t]_1^\tau$ with $\pi(da_t\|pa(a_t)) = \mathbb{P}(da_t\|pa(a_t))$
g	User-specified treatment policies $g = [g_t]_{t=1}^\tau$
ψ	Target functional $\psi(P) = \mathbb{E}_g Y$
ψ_0	True parameter $\psi_0 = \psi(P_0)$
$\hat{\psi}_n$	Estimator of ψ_0
$\hat{\sigma}_n$	Estimator of the standard error of the estimator $\hat{\psi}_n$
Q	State-action value functions $Q = [Q_t]_{t=1}^\tau$
Q_t	$= Q_t(pa(Y_t)) = \mathbb{E}_g[Y\|pa(Y_t)]$
V	Value functions $V = [V_t]_{t=1}^{\tau+1}$
V_t	$= V_t(pa(A_t)) = \mathbb{E}_g[Y\|pa(A_t)]$ for $t = 1, \ldots, \tau$ and $V_{\tau+1} = Y_\tau$
I_t	Clever covariates (importance weights) $I_t = \prod_{s=1}^t dg_s/d\pi_s(O)$
$D^\star$	Efficient influence function of ψ: $D^\star(Q, V; \pi) = V_1 - \psi_0 + \sum_{t=1}^\tau (V_{t+1} - Q_t)$
Q_ε	Local least favorable submodel $Q_\varepsilon = [Q_{t,\varepsilon}]_{t=1}^\tau$
Q_t	$\text{logit } Q_{t,\varepsilon} = \text{logit } Q_t + \varepsilon$
V_ε	Local least favorable submodel $V_\varepsilon = [V_{t,\varepsilon}]_{t=1}^{\tau+1}$
V_t	$\text{logit } V_{t,\varepsilon} = \text{logit } V_t + \varepsilon$ for $t = 1, \ldots, \tau$ and $V_{\tau+1,\varepsilon} = V_{\tau+1}$
$\mathcal{L}(Q, V)$	Loss function for temporal difference learning
$\mathcal{L}^\star$	Loss function for targeting
α	Weight for the propensity loss (hyperparameter)
Pf	Mean of a function f under the distribution P: $Pf = \int f dP$
$H(\bullet_t)$	Embedding of a node $\bullet_t$
$h_\bullet(\bullet_t)$	Type embedding of a node $\bullet_t$
E_t	Positional encoding at time t
f_X	production function of a node X

B. Proof

Proof of Theorem 4.2. A direct calculation shows

$$\partial_\varepsilon \mathcal{L}_t^\star(Q_{t,\varepsilon}, V_{t+1,\varepsilon^\star}) = I_t(\pi)[\partial_\varepsilon Q_{t,\varepsilon}][\partial_{Q_{t,\varepsilon}} \mathcal{L}_{\text{bce}}(Q_{t,\varepsilon}, V_{t+1,\varepsilon^\star})] = I_t(\pi)(V_{t+1,\varepsilon^\star} - Q_{t,\varepsilon}).$$

Substitution of $\varepsilon^\star$ to ε yields the t-th component of the efficient influence function (35) at $(Q_{\varepsilon^\star}, V_{\varepsilon^\star}, \pi)$. $\square$

C. Review of TMLE

C.1. Structural Causal Model

We assume each node depends on the all previous nodes in the trajectory, that is, we do not assume the Markovian property. And each node X is produced from the parent nodes $pa(X)$ and independent noise random variables U_X by a measurable function f_X: $X = f_X(pa(X), U_X)$. This production function f_X induces a conditional distribution $P_{X|H}$ of X given $H = pa(X)$ by pushing forward the distribution of noise variable: $P_{X|H}(A|h) = (P_{U_X} \circ f_X^{-1}(h, \cdot))(A)$ for all measurable

$A \subset \mathcal{X}$, where $\mathcal{X}$ is a domain of random vector X. Starting from nodes without parents including noise nodes and their distributions, production functions and their causal structure, which can be described by a directed acyclic graph over the ovservables, generate the joint distribution of the observed random variables. With our particular data in longitudinal setting, we define the propensity score $\pi_t = P_{A_t|pa(A_t)}$, where $pa(A_t) = pa(A_t)$ is the patient history before the node A_t. We use the same symbol for the production function if the treatment assignment is deterministics, that is, there is no noise variable in generating the treatment node: $d\pi_t(A_t|pa(A_t)) = 1$ if $A_t = a_t$ for some specific a_t.

C.2. Causal Target Parameter and Identification

Our target parameter is the counterfactual mean of the final outcome Y under the user-specified dynamic treatment policy $g = (g_t)$. This is the mean of counterfactual outcome which is produced by replacing π_t with g_t in the structural causal model:

$$\psi^g(P) = EY^g. \tag{25}$$

To identify this causal target paratmer from observatoinal data, we assume the following conditions of the positiviy:

$$g \ll \pi, \tag{26}$$

and the sequential randomization:

$$Y^g \perp A_t \mid pa(A_t) \text{ for } t = 1, \ldots, \tau. \tag{27}$$

Note that the consistency $Y = Y^\pi$ usually stated in the causal inference literature is a consequence of the definition of counterfactual outcome in our structural causal model. Under these identifiability conditions, this parameter is identified through g-formula that is the mean of Y under the counterfactual distribution which is given by replacing distributions π_t with g_t:

$$EY^g = E_g Y. \tag{28}$$

Then the problem reduced to the estimation of the statistical parameter:

$$\psi(P) = E_g Y. \tag{29}$$

C.3. TMLE

Bias correction by TMLE is based on the following first order approximation of the target functional around the true distribution P_0 (van der Laan & Rubin, 2006; van der Laan & Rose, 2011; Kennedy, 2022):

$$\psi(\hat{P}_n) - \psi(P_0) = -P_0 D^\star(\hat{P}_n) + R_2(\hat{P}_n, P_0), \tag{30}$$

where $D^\star$ is called influence function and R_2 is the second order remainder. This equation is the infinite dimensional extension of Taylor expansion.

The right hand side of this equation can be further written as:

$$-P_n D^\star(\hat{P}_n) + (P_n - P_0)\left[D^\star(\hat{P}_n) - D^\star(P_0)\right] + (P_n - P_0)D^\star(P_0) + R_2(\hat{P}_n, P_0), \tag{31}$$

whose second term called empirical process term converges to zero in the rate of square root of n if $D^\star(\hat{P}_n), D^\star(P_0)$ belong to the Donsker class and the $L^2(P_0)$-distance between $D^\star(\hat{P}_n)$ and $D^\star(P_0)$ converges to zero in probability. Given a good initial fit $\hat{P}_n$ of P_0, above conditions are usually satisfied and, in addition, we require $R_2(\hat{P}_n, P_0) = o_{P_0}(n^{-1/2})$ which we will look at closely in Section C.5. By further using the fact about the influence function that $P_0 D^\star(P_0) = 0$, the right hand side reduced to

$$-P_n D^\star(\hat{P}_n) + P_n D^\star(P_0) + o_{P_0}(n^{-1/2}). \tag{32}$$

Now, the idea is to find $\hat{P}_n^\star$ in the close neighborhood of $\hat{P}_n$ that solves the empirical analog of the first term:

$$P_n D^\star(\hat{P}_n^\star) = 0. \tag{33}$$

By doing so, using similar arguments as above for $\hat{P}_n^\star$ instead of $\hat{P}_n$, we have the following.

$$\psi(\hat{P}_n^\star) - \psi(P_0) = P_n D^\star(P_0) + o_{P_0}(n^{-1/2}). \tag{34}$$

Thus, our estimator $\psi(\hat{P}_n^\star)$ is a plug in estimator and attains the efficiency bound among the asymptotically linear and regular estimators.

C.4. Efficient Influence Curve

Then the efficient influence function of our target parameter (6) is computed as follows (van der Laan & Gruber, 2012)

$$D^\star(P) = \sum_{t=1}^{\tau} D_t^\star(P) = (V_1 - \psi_0) + \sum_{t=1}^{\tau} I_t(V_{t+1} - Q_t) = (V_1 - \psi_0) + \sum_{t=1}^{T} I_t(V_{t+1} - Q_t). \tag{35}$$

The last equality follows since $V_{t+1} = Q_t$ if $t > T$.

C.5. Second-order Remainder and Double Robustness

In this subsection, we derive the exact remainder (Theorem C.1) and see its double robustness (Theorem C.2).

Theorem C.1. *The second-order remainder of the target functional* (6) *is given by:*

$$R_2(\hat{P}, P) = \sum_{t=1}^{\tau} \int_{\mathcal{O}_{1:t}} \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \Big[\int V_{t+1}(\hat{\eta}) d\eta_{t+1} - Q_t(\hat{\eta}) \Big] d(\pi_{1:t} - \hat{\pi}_{1:t}) d\eta_{1:t}. \tag{36}$$

Proof. First, we could consider the TDHD (Figure 1) implicitly learns conditional distributions $\eta_t = P_{Y_{t-1}, L_t | pa(Y_{t-1})}$ for $t = 2, \ldots, \tau$ and outputs

$$\hat{Q}_t = Q_t(\hat{\eta}) = (g_{t+1:\tau} \otimes \hat{\eta}_{t+1:\tau+1})[Y] = \int_{\mathcal{O}_{t+1:\tau}} Y d(g_{t+1:\tau} \otimes \hat{\eta}_{t+1:\tau+1}) \tag{37}$$

$$\hat{V}_t = V_t(\hat{\eta}) = (g_{t:\tau} \otimes \hat{\eta}_{t+1:\tau+1})[Y] = \int_{\mathcal{O}_{t+1:\tau}} Y d(g_{t:\tau} \otimes \hat{\eta}_{t+1:\tau+1}) \tag{38}$$

for $t = 1, \ldots, \tau$. Here we used a notation $\mu\nu f = (\mu \otimes \nu)[f] = \int f d(\mu \otimes \nu)$ for disintegrations of measures $\mu \otimes \nu$ and functions f, identifying the measure $\mu \otimes \nu$ with a linear functional from some functional class $\mathcal{F}$ to the real numbers. Let $\hat{\eta}_1$ be the empirical distribution of W and L_1, $P = \eta \otimes \pi$ be the true and $\hat{P} = \hat{\eta} \otimes \hat{\pi}$ be the initial estimate of the joint density of the observables. By definition (30), the second-order remainder is

$$R_2(\hat{P}, P) = \psi(\hat{P}) - \psi(P) + PD^\star(\hat{\eta} \otimes \hat{\pi})$$

$$= \Big[g(\hat{\eta} - \eta) + g(\eta_1 \hat{\eta}_{2:\tau+1} - \hat{\eta}_{1:\tau+1}) + \sum_{t=1}^{\tau} \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \pi_{1:t} g_{t+1:\tau} \eta_{1:t}(\eta_{t+1} - \hat{\eta}_{t+1}) \hat{\eta}_{t+2:\tau+1} \Big][Y]$$

$$= \Big[g \sum_{t=1}^{\tau} \big(\eta_{1:t} \hat{\eta}_{t+1:\tau+1} - \eta_{1:t+1} \hat{\eta}_{t+2:\tau+1} \big) + \sum_{t=1}^{\tau} \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \pi_{1:t} g_{t+1:\tau} \eta_{1:t}(\eta_{t+1} - \hat{\eta}_{t+1}) \hat{\eta}_{t+2:\tau+1} \Big][Y] \tag{39}$$

$$= \Big[\sum_{t=1}^{\tau} \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} (\pi_{1:t} - \hat{\pi}_{1:t}) g_{t+1:\tau} \eta_{1:t}(\eta_{t+1} - \hat{\eta}_{t+1}) \hat{\eta}_{t+2:\tau+1} \Big][Y]$$

$$= \sum_{t=1}^{\tau} \int_{\mathcal{O}_{1:t}} \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \Big[\int V_{t+1}(\hat{\eta}) d\eta_{t+1} - Q_t(\hat{\eta}) \Big] d(\pi_{1:t} - \hat{\pi}_{1:t}) d\eta_{1:t}.$$

$\square$

Theorem C.2. *Suppose* $\delta = \sup_{t,o} dg_{1:t}/d\hat{\pi}_{1:t}(o)$ *exists and* $\big\|\hat{Q}_t - Q_t\big\|_{\mathbb{L}^2(\nu_{1:t} \otimes \eta_{1:t})} \big\|\hat{e}_s - e_s\big\|_{\mathbb{L}^2(\nu_{1:s} \otimes \eta_{1:s})} = o_P(n^{-1/2})$ *for all* t *and* $s \leq t$, *then* $R_2(\hat{P}, P) = o_P(n^{-1/2})$.

Proof. Noting that

$$\int V_{t+1}(\eta) d\eta_{t+1} - Q_t(\eta) = 0, \tag{40}$$

since the first term in the left side is the definition of Q_t, we have

$$\sum_{t=1}^{\tau} \int_{\mathcal{O}_{1:t}} \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \Big[\int V_{t+1}(\eta) d\eta_{t+1} - Q_t(\eta) \Big] d(\pi_{1:t} - \hat{\pi}_{1:t}) d\eta_{1:t} = 0. \tag{41}$$

Let $e_t = d\pi_t/d\nu_t$ and $\hat{e}_t = d\hat{\pi}_t/d\nu_t$ be the propensity scores with respect to the counting measures ν_t over the action space. By subtracting (41) from the second-order remainder (36), we have

$$R_2(\hat{P}, P) = \sum_{t=1}^{\tau} \int \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \left(V_{t+1}(\hat{\eta}) - V_{t+1}(\eta) \right) \left[\sum_{s=1}^{t} e_{1:s-1}(e_s - \hat{e}_s)\hat{e}_{s+1:t} \right] d\nu_{1:t} d\eta_{1:t+1} \tag{42}$$

$$- \sum_{t=1}^{\tau} \int \frac{dg_{1:t}}{d\hat{\pi}_{1:t}} \left(Q_t(\hat{\eta}) - Q_t(\eta) \right) \left[\sum_{s=1}^{t} e_{1:s-1}(e_s - \hat{e}_s)\hat{e}_{s+1:t} \right] d\nu_{1:t} d\eta_{1:t}. \tag{43}$$

Applying the Cauchy-Schwartz inequality, we have

$$\left| R_2(\hat{P}, P) \right| \leq 2\delta \sum_{t=1}^{\tau} \sum_{s=1}^{t} \left\| Q_t(\hat{\eta}) - Q_t(\eta) \right\|_{\mathbb{L}^2(\nu_{1:t} \otimes \eta_{1:t})} \left\| e_s - \hat{e}_s \right\|_{\mathbb{L}^2(\nu_{1:s} \otimes \eta_{1:s})}. \tag{44}$$

$\square$

D. Convergence of Temporal Difference Learning

First, consider a flexible model Q_θ and corresponding $V_{t,\theta} = \mathbb{E}_{A_t \sim g_t} Q_{t,\theta}$. Initiate θ^0 and then iteratively update by $\theta^{k+1} = \arg\min_\theta PL(Q_\theta, V_{\theta^k})$ for $k = 2, \ldots$, till convergence. Our proof below shows that if we use a variation independent parameter space for each $Q_{t,\theta}$ and the parameter spaces contain the true $Q_{t,P}$, then in $K+2$-steps this algorithm will have converged to the true solution Q_P.

Ignoring the parameterization, but just thinking in terms of optimizing over parameter spaces, this algorithm corresponds with: initiate V^0, and then for $k = 0, \ldots$, compute $Q^{k+1} = \arg\min_Q PL(Q, V^k)$ and set V^{k+1} as the one implied by the intervention g and Q^{k+1}; and set $k = k + 1$.

Firstly, we claim that in a nonparametric model the t-specific parameters Q_t are variation independent across t. Consider a given V (misspecified). This implies a parameter space $\left\{ \mathbb{E}_{L_t|pa(L_t) \sim \mu_t(\cdot|pa(L_t))} V_t : \mu_t \right\}$ for the regressions Q. The parameter space of the free parameter μ_t is even larger than the parameter space of functions of $pa(L_t)$. Therefore this appears indeed a reasonable condition. Then we can state that the parameter space over which we optimize at step k of the algorithm is the cartesian product of the parameter spaces $\mathcal{Q}_t$ for Q_t across $t = \tau, \ldots, 1$. Consider the $k = 1$-step of the algorithm in which the outcomes are V^0 and we optimize over all the $Q \in \prod_{t=\tau}^{1} \mathcal{Q}_t$. Then, Q^1 is the minimizer of $Q \to PL(Q, V^0)$. That means that the derivative w.r.t. ε_t along a path $Q_t^1 + \varepsilon_t h_t$ through Q_t^1 in any direction h_t at $\epsilon_t = 0$ should be equal to zero, across all $t = \tau, \ldots, 1$. Thus, at $\varepsilon = 0$, we have

$$\frac{d}{d\varepsilon} \sum_{t=1}^{\tau} \mathcal{L}_t(Q_t^1 + \varepsilon_t h_t \mid V_{t+1}^0) = 0$$

Consider the derivative w.r.t. ε_Y. This yields the score equation $Ph_{Q_\tau}(V_{\tau+1} - Q_\tau^1) = 0$ for all h_{Q_τ}. This implies that $Q_\tau^1 = Q_{\tau,P}$. The others are some optimizer. Now, we go to step $k = 2$. We now know that $V_\tau^1 = V_{\tau,P}$ due to $Q_\tau^1 = Q_{\tau,P}$. Therefore, at the next step, due to the derivative w.r.t. $\varepsilon_{\tau-1}$, it follows that $Q_{\tau-1}^2 = Q_{\tau-1,P}$, while it again $Q_\tau^2 = Q_\tau^1 = Q_{\tau,P}$. Then, at step $k = 3$, we also obtain $Q_{\tau-2}^3 = Q_{\tau-2,P}$. In this manner, it follows that after $K+2$ steps we have $Q^{K+2} = Q_P$.

E. Evaluation of V

To evaluate V_t using a fit $\hat{Q}_t$, we integrate A_t out under the counterfactual intervention g_t. Since we assume deterministic intervention $A_t^g = g(pa(A_t))$, we computed it as $\hat{V}_t = \hat{Q}_t(L_{1:t}, a_{1:t})$ where a_s is defined by $a_s = g_s(L_{1:s}, A_{1:s-1})$ for $s = 1, \ldots, t$. This is based on the almost sure equality $\mathbb{E}_g[Y|L_{1:t}, A_{1:t-1}] = \mathbb{E}_g[Y|L_{1:t}, a_{1:t-1}]$ as random variables. To see this, note that for general random variables X and Y the conditional expectation $\mathbb{E}[Y|X = x]$ can take any value whenever $\mathbb{P}(X = x) = 0$. An alternative approach is computing $\hat{V}$ values as $\hat{V}_t = \hat{Q}_t(L_{1:t}, A_{1:t-1}, a_t)$ and both are almost surely equivalent. Here we supressed Y values from the history assuming all those values were zero without loss of generality.

F. Hyperparameter Tuning

We selected hyperparameters shown in Table 4 which optimized the empiricall loss $\mathcal{L}^Q + \mathcal{L}^e$ in the validation set which is the 30% of the entire dataset. The parameter α and β for censoring mechanism balances the learning rate of π-parts and Q-parts because the complexity of π-parts would be simpler than Q-parts which involves prediction in the long-range.

Table 4. Selected hyperparameters.

Data	Simple Synthetic Data						Complex Synthetic Data						Real World
Model	Deep LTMLE			DeepACE			Deep LTMLE			DeepACE			Deep LTMLE
τ	10	20	30	10	20	30	10	20	30	10	20	30	30
Embedding dimension	16	16	32	—	—	—	16	32	32	—	—	—	32
Dropout rate	0.1	0.2	0	0.1	0.1	0.2	0	0	0	0.3	0.3	0.3	0
Hidden size	32	32	16	16	32	32	64	32	16	16	16	16	16
Number of Layers	8	4	8	4	4	4	4	4	4	8	8	4	8
Number of heads	4	4	4	—	—	—	8	8	8	—	—	—	4
Learning rate	1e-3	5e-4	1e-4	5e-3	5e-4	1e-3	1e-3	5e-4	1e-4	5e-4	5e-4	5e-4	5e-4
α	0.1	0.1	0.05	0.1	0.1	0.05	0.01	0.05	0.05	0.1	0.1	0.1	0.1
β	—	—	—	0.05	0.05	0.05	—	—	—	0.05	0.05	0.05	0.01
Number of epochs	100	100	100	100	100	100	100	100	400	100	100	100	100

G. Synthetic Data

G.1. Simple Synthetic Data with Continuous Outcome

The process iteratively generates variables W, L_t, A_t, and Y_t over time steps t, for $t = 0, \ldots, \tau - 1$. $W \sim \text{Normal}(0, 1)$. At $t = 0$, $L_0 \sim \text{Normal}(0.1W, 1)$, $A_0 \sim \text{Ber}(\sigma(-0.5W + L_0))$, $Y_0 \sim \text{Ber}(\sigma(-3 + 0.2W + 0.2L_0 - 2A_0))$. For $t > 0$, $L_t \sim \text{Normal}(0.1W - 0.1A_{t-1}, 1)$, $A_t \sim \text{Ber}(\sigma(-0.5 + 0.3W + 0.3L_t + 2A_{t-1}))$, $Y_t = \sigma(-3 + 0.2W + 0.2L_t - 2A_t)$, $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. We set the counterfactual treatment at all time-points to 1 and and evaluated the counterfactual mean of survival under this treatment policy.

G.2. Simple Synthetic Data with Survival Outcome

The process iteratively generates variables W, L_t, A_t, and Y_t over time steps t, for $t = 0, \ldots, \tau - 1$. $W \sim \text{Normal}(0, 1)$. At $t = 0$, $L_0 \sim \text{Normal}(0.1W, 1)$, $A_0 \sim \text{Ber}(\sigma(-0.5W + L_0))$, $Y_0 \sim \text{Ber}(\sigma(-3 + 0.2W + 0.2L_0 - 2A_0))$. For $t > 0$, $L_t \sim \text{Normal}(0.1W - 0.1A_{t-1}, 1)$, $A_t \sim \text{Ber}(\sigma(-0.5 + 0.3W + 0.3L_t + 2A_{t-1}))$, $Y_t \sim \text{Ber}(\sigma(-3 + 0.2W + 0.2L_t - 2A_t))$, with $Y_{t-1} = 1$ implying $Y_t = 1$. Here $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. We set the counterfactual treatment at all time-points to 1 and and evaluated the counterfactual mean of survival under this treatment policy.

G.3. Complex Synthetic Data with Survival Outcome

First draw parameters $\alpha_i, \beta_i \sim \text{Normal}((i + 1)^{-1}, 0.02)$ and $\gamma_i \sim 2 * \text{Binom}(0.5) - 1$ for $i \in [ht]$, where h is the length of time-dependency with $h = 1$ corresponding to Markovian process. Then, draw error in time-dependet variables $\varepsilon_{tj}^\ell \sim \text{Normal}(0, 0.1)$ for $t \in [\tau]$ and $j \in [p]$, errors in treatment $\varepsilon_{t1}^A \sim \text{Normal}(0, 0.2)$, $\varepsilon_{t2}^A \sim \text{Normal}(0, 0.05)$ for $t \in [\tau]$. For each $t \in [\tau]$, $L_t = \tanh\left(\sum_{k \in [ht]} \alpha_k L_{t-k} + \beta_k \gamma_k (2A_{t-k} - 1)\right) + \varepsilon_{tj}^\ell$, then draw A_t from an indicator function $\mathbb{1}\{(\sigma(s_t+) + \varepsilon_{t2}^A) > 0.5\}$, with $s_t = \tan\left(\prod_{j \in [p]} \bar{L}_j + \bar{A}\right) + \varepsilon_{t1}^A$. The outcome Y_t is drawn from a Bernoulli distribution of a probability $\sigma(p_t)$ with $p_t = \tan\left(\prod_{j \in [p]} \bar{L}_j - 0.7 * (\bar{A} - 0.5)\right) - 4.5$. $Y_t = 1$ if $Y_{t-1} = 1$ for $t > 0$. We set the counterfactual treatment policy as $\mathbb{1}\{\sigma(s_t) > 0.5\}$ for $t \in [\tau]$ and evaluated the counterfactual mean of survival under this policy.

H. Results with Simple Synthetic Data with Survival Outcome

Results of an experiment with the simple synthetic data described in Section G.2 was shown in Table 5. Although LTMLE's strong performance on simple synthetic data is anticipated due to reduced burden in estimating nuisance parameters from

Markovian dependencies, Deep LTMLE remains highly competitive in this context, equalling LTMLE's performance. Our two targeting approaches demonstrated better bias variance trade off for the estimation of the target parameter compared to the untargeted approach. Both bias and standard deviation get improved a lot for all τ's considered. The targeting step made a marked difference in terms of coverage probability, getting much closer to a nominal 95% coverage probability compared to the one without targeting.

Table 5. Results from simple synthetic data

Model	Bias			RMSE			Coverage			Mean $\hat{\sigma}_n$		
	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 10$	$\tau = 20$	$\tau = 30$
LTMLE (GLM)	0.0052	0.0045	0.0021	0.0202	0.0268	0.0308	0.95	0.94	0.93	0.02	0.03	0.03
LTMLE (SL)	0.0056	0.0058	0.0061	0.0203	0.0263	0.0311	0.91	0.93	0.91	0.02	0.02	0.03
DeepACE	0.0260	-0.0061	-0.0976	0.1558	0.0380	0.1232	0.46	0.93	0.30	0.06	0.04	0.04
Deep LTMLE	0.0045	0.0176	0.0257	0.0280	0.0328	0.0710	0.82	0.87	0.73	0.02	0.03	0.04
Deep LTMLE†	0.0065	0.0068	0.0009	0.0225	0.0295	0.0347	0.91	0.91	0.92	0.02	0.03	0.04
Deep LTMLE⋆	0.0059	0.0106	0.0036	0.0226	0.0296	0.0360	0.91	0.92	0.90	0.02	0.03	0.04

H.1. Semi-Synthetic Data

As a compromise, we conducted several additional experiments with semi-synthetic data from the real world data as used in previous studies (Bica et al., 2020; Frauen et al., 2023). For this experiment, we used covariates from the Circulatory Risk in Communities Study (CIRCS) and fit outcome regression given the history through each time point using XGBoost with early stopping. Outcomes were then generated using this fitted regression model. For the experiment, we sample 1000 observations from the empirical dstribution of covariates W, L_t, A_t and generate Y_t for $t = 1, \ldots, \tau$ with $\tau \in \{10, 20.30\}$.

I. Extension to Survival Analysis with Censoring

In this section, we describe the extended LTMLE model with censoring for the real world application in Section 5.4. We assume the following order of observed nodes $O = (W = W_0, L_1, A_1, C_1, Y_1, L_2, A_2, C_2, Y_2, \ldots, L_\tau, A_\tau, C_\tau, Y_\tau = Y)$, where C_t are binary censoring nodes with $C_t = 1$ indicating one being censord. Our interest is to estimate the risk of our outcome Y_τ, the mortality of the individual. However, our observation period spans long-term, individuals are at risk of being censored. Censoring C_t is loss of follow-up from administrative reasons, for example, move to other areas or denial of participation in the survey. We assume degenerations of nodes. When we observe a jump in Y or C nodes, the process halts and all nodes after the jump remain constant with the last observed values. For example, if $Y_t = 1$, then $Y_s = 1$, $C_s = 0$, $A_{s-1} = A_{t-1}$, and $L_s = L_t$ for all $s > t$.

We constructed a Deep LTMLE similar to the one describe in Section 4 with this structure. The only difference is an additional component of censoring mechanism λ^c which is involved in the clever covariate I_t and the loss function:

$$I_t(\pi, \lambda^c) = \prod_{s=1}^{t} \frac{d(g \otimes \mathbb{1}(C_s = 0))}{d(\pi_t \otimes \lambda_t^c)}(O), \text{ and} \tag{45}$$

$$\mathcal{L}(Q, V, \pi, \lambda^c) = \mathcal{L}^Q(Q, V) + \alpha \mathcal{L}^e(\pi, A) + \beta \mathcal{L}^c(\lambda^c, C), \tag{46}$$

where β is an additional scaling hyperparameter. The counterfactual treatment on the censoring process is $\mathbb{1}(C_t = 0)$ meaning supression of censoring. Estimates of the target parameter and the efficient influence curve for different treatment strategies are computed using Deep LTMLE, and average treatment effects (ATEs) and its EIC were computed using the delta method. Based on the estimated EICs of the target parameters at each time point t, we constructed a simultaneous confidence intervals.