DeepBlip: Estimating Conditional Average Treatment Effects Over Time

Anonymous Author(s)
Affiliation
Address
email

Abstract

Estimating the conditional average treatment effect (CATE) over time is crucial for making personalized decisions in medicine. Yet, existing neural methods for this task have limitations: they either (1) do not adjust for time-varying confounding and are thus biased (e.g., causal transformer), or (2) become unstable over long time horizons because the method has to learn the full counterfactual outcome trajectories (e.g., MSNs, G-computation). To address these limitations, we propose DeepBlip, the first neural framework that leverages the blip function from structural nested mean models to break the joint effect of treatment sequences over time into localized, time-specific "blip effects". As a result, we learn a simpler estimand that does not require learning full counterfactual outcome trajectories, which is thus more stable over long horizons. Further, our DeepBlip adjusts for time-varying confounding and is thus unbiased. Our DeepBlip seamlessly integrates sequential models like LSTMs or transformers to capture complex temporal dependencies. Our DeepBlip has two further strengths for medical practice: (i) The loss is Neyman-orthogonal, meaning it is robust against model misspecification. (ii) The blip effects can be used to predict treatment effects for new treatment sequences without re-computation, which allows to identify optimal treatment sequences through offline evaluation. Finally, we evaluate our DeepBlip across various clinical datasets, where it achieves state-of-the-art performance.

1 Introduction

Predicting the effects of treatment sequences is crucial for personalized medicine to choose the best therapeutic strategy for a patient based on their history [8]. Methodologically, the **conditional average treatment effect (CATE)** ***over time*** captures the combined effect of multiple treatments in the next τ time steps (see Fig. 1). Nowadays, CATEs over time are frequently estimated from observational data with patient histories, such as the electronic health records [2, 4].

Several works aim to estimate CATE over time from observational data, but they suffer from two key limitations: ① **No proper adjustment for confounding and thus bias:** There are methods that do *not* properly adjust for time-varying confounding (e.g., CRN [3], causal transformer [20]) and that are thus *biased*. This leads to unreliable estimates, which is particularly problematic for safety-critical applications such as personalized medicine. ② **Unstable for long time horizons:** Other methods require modeling the *full* counterfactual outcome trajectories. This is the case in MSNs, which must learn long-range treatment-response mappings (e.g., as RMSNs [19]), or g-computation, which relies on modeling the full data-generating process of covariates and outcomes (e.g., G-transformer [14]). To the best of our knowledge, there is **no** method for estimating CATE over time that has addressed both challenges ① and ②.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In principle, one way to address the above limitations is through the theortical framework of **structural nested mean models** (SNMMs) [27, 28]. SNMMs provide a principled foundation for estimating CATEs over time in an *unbiased* way. For this, SNMMs decompose the time-varying CATE into a sequence of *incremental treatment effects*, formalized through so-called ***blip functions***. This decomposition yields several important advantages: (ii) It enables a divide-and-conquer approach that breaks the CATE over time into localized, time-specific causal effects. As a result, SNMMs define an estimand that is easier to learn than in many other methods (e.g., MSNs) and thus avoid the need to model full counterfactual trajectories. (ii) Because blip functions are conditionally independent across time given a patient's history, estimation errors do *not* propagate, which makes them more stable for long time horizons. *However*, SNMMs are *only* a theoretical foundation (and, therefore, *not* a model that can be directly applied). So far, one study by Lewis et al. [18] has employed SNMMs, yet only instantiated using linear models. To the best of our knowledge, no prior work has developed a neural version of SNMMs.

Here, we propose **DeepBlip**, the first *neural* framework to estimate CATE over time by leveraging the blip function from SNMMs. DeepBlip decomposes the joint effect of treatment sequences over time into localized, time-specific blip effects, which enables more tractable and stable learning. This allows our DeepBlip to overcome both of the two limitations from above: (1) Our DeepBlip adjusts for time-varying confounding and is thus *unbiased*. (2) Our DeepBlip targets a simpler estimand than many of the above methods, thereby avoiding the need to learn the full counterfactual outcome trajectories and which improves the stability of DeepBlip over long time horizons. Our DeepBlip is built on top of sequential neural networks (e.g., LSTMs, transformers) to capture complex temporal dependencies. For this, it employs a two-stage architecture: Stage 1 models the probability of time-varying treatments and mean outcomes conditioned on a patient's history, while Stage 2 reformulates g-estimation [27, 28] as a risk minimization task to directly learn the blip functions.

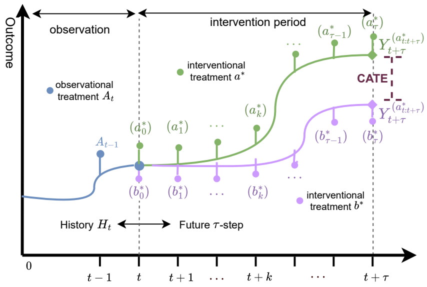

Figure 1: **CATE over time.** Trajectories of potential outcomes under two interventional sequences $a^*_{t:t+\tau}$, $b^*_{t:t+\tau}$ given the shared observed history H_t. The difference between the two curves is the CATE over time.

Our DeepBlip has two further strengths for medical practice: (i) The loss *Neyman-orthogonal*, meaning it is *robust against model misspecification*. (ii) The learned blip effects can be reused to predict treatment outcomes for new treatment sequences *without* re-computation. This enables an efficient approach for *offline evaluation* of different therapeutic strategies. Formally, at inference time, DeepBlip can identify the optimal treatment sequence within just one forward pass. This is unlike other methods, which typically require either re-training [11, 14] or multiple forward passes due to exhaustive search [3, 19, 20, 31]. This property is especially beneficial for clinicians when searching for optimal treatment sequences.

Our **contributions** are three-fold:[1] (**1**) We introduce the first neural framework to predict CATE over time via the SNMM framework. (**2**) Our framework is carefully tailored to medical practice by benefiting from robustness due to Neyman-orthogonality and efficient offline evaluation. (**3**) We conduct extensive experiments across multiple medical datasets to demonstrate that our DeepBlip is effective and also robust across long time horizons.

2 Related Work[2]

Estimating CATE in the static setting: There has been extensive research in estimating ATE/CATE in the static setting (e.g., [1, 7, 17, 33, 37, 41]). Recently, deep learning has been used to improve the non-parametric estimation of ATE/CATE [41]. However, these methods are aimed at static settings and thus struggle with medical datasets such as electronic health records, where patient histories are recorded *over time*.

[1] Code for review is available at https://anonymous.4open.science/r/DeepBlip-A39B Upon acceptance, we will move our code to a public GitHub repository.

[2] We provide an extended related work in Appendix B

Estimating ATE over time: One line of work has developed methods for the ATE over time [10, 34]. However, the ATE captures only population-level effects and thus overlooks differences in treatment effectiveness across patients. In contrast, we focus on the CATE, which provides a more granular, individualized estimate of treatment outcomes, which is highly relevant for personalized medicine [8].

Estimating CATE over time: There are several *neural* methods for this task[3], which can be broadly categorized into two streams but with notable limitations (see Table 1):

Limitation ①: *No proper adjustment for confounding and thus bias:* Some methods for CATE over time fails to properly adjust for time-varying confounding properly, which leads to estimates that are *biased*. Here, prominent examples are counterfactual recurrent network (**CRN**) [3] and the causal transformer (**CT**) [20]. These methods attempt to alleviate time-varying confounding via balanced representations. However, balancing was originally designed for reducing finite-sample estimation variance and *not* for mitigating confounding bias [33]. Hence, such methods act as heuristics without a theoretical justification. The difficulty of enforcing balanced representations may even introduce further confounding bias [21]. Unlike these methods, our DeepBlip allows for proper adjustments and is thus unbiased.

Limitation ②: *Unstable for long time horizons:* Other neural methods build on frameworks from statistics such as marginal structural models (MSMs) [22, 25] and G-computation [26, 29]. Examples are: (i) **G-Net** [31] and G-transformer (**GT**) [14], which are both based on G-computation and thus compute nested conditional expectations over time. Hence, these methods require the entire counterfactual outcome trajectories and thus model the full data-generating process of covariates and outcomes, which becomes exponentially more complex as time horizons grow. (ii) MSMs-based methods like **R-MSNs** [19] and the **DR-learner** (for time-varying settings) [11] use inverse propensity weighting (IPW) to re-weight outcomes as in a randomized control trial. In time-varying settings, the propensity is a multiplication of a sequence of probabilities, which has known to become instable when the horizon is large.[4] In sum, while these methods are unbiased, they require modeling *entire* counterfactual outcome trajectories, which makes learning *unstable* over long time horizons. As a remedy to this, our DeepBlip breaks the CATE into localized effects at each time step via blip functions.

Structural nested mean models (SNMMs): SNMMs [27, 28] offer a principled framework for estimating CATE over time by directly estimating incremental treatment effects via so-called blip functions. In principle, SNMMs could address both of the above limitations; however, SNMMs are only an abstract theoretical foundation, ***not*** an off-the-shelf model that can be directly applied. Early implementations [27, 28] relied on strong parametric assumptions (e.g., linearity) and were limited to short time horizons. Recently, Lewis et al. [18] employed SNMMs with linear models, yet which thus fails to condition on the rich, complex information in patient histories and which makes it unsuitable for personalized medicine in realistic, high-dimensional settings for medicine. So far, to the best of our knowledge, a neural instantion of SNMMs are missing.

Method	**Methodological limitations**		**Benefits for medical practice**	
	① Unbiased	② Stable wrt long horizons	Orthogonal	Offline efficiency
CRN [3]	✗	✓	✗	✓
CT [20]	✗	✓	✗	✓
G-Net [31]	✓	✗	✗	✗†
GT [14]	✓	✗	✗	✗††
R-MSNs [19]	✓	✗	✗	✗†
DR-learner [11]	✓	✗	✓	✗††
DeepBlip (*ours*)	✓	✓	✓	✓

†: Needs re-computation; ††: Needs re-training (to identify best treatment)

Table 1: **Neural methods for learning CATE over time**.

Research gap: To the best of our knowledge, there is <u>no</u> neural implementation of SNMMs. We thus extend upon the work of Lewis et al. [18] and develop the first neural framework for SNMMs. For this, we introduce a new, flexible architecture and a tailored two-stage learning algorithm that allows us to leverage the blip function from SNMMs for CATE estimation over time. As a result, ours is the first neural method to address both limitations ① and ② from above.

[3] There have been some attempts to use non-parametric models [32, 35, 40], yet these approaches impose strong assumptions on the outcomes and their scalability is limited. For these reasons, we foucs on neural methods, which offer better flexibility and scalability for complex, high-dimensional medical data.

[4] This is due to overlap violations, which is especially challenging in time-varying settings, since the number of possible treatment combinations grows exponentially with the horizon [11]. Hence, IPW often leads to extreme weights, and thus instabilities due to division by values close to zero.

3 Problem Formulation

Setup: We follow the standard setup [3, 11, 14, 20] for estimating CATE over time $t \in \{1, 2, \ldots, T\} \subset \mathbb{N}$ given by: (1) the target outcome $Y_t \in \mathbb{R}$; (2) time-varying covariates $X_t \in \mathcal{X} \subset \mathbb{R}^{d_x}$; and (3) treatment $A_t \in \mathcal{A} \subset \mathbb{R}^{d_a}$ that can be either discrete or continuous. We assume w.l.o.g. that the static features (e.g., age, sex) are included in the covariate.

Notation: To simplify notation, we use overlines to denote the full sequence of a variable (e.g., $\overline{X}_t = (X_1, \ldots, X_t)$). We refer to a sequence of variables that starts at t and ends at $t + \tau$ via $A_{t:t+\tau} = (A_t, A_{t+1}, \ldots, A_{t+\tau})$. We use the lowercase letter to denote a realization of a random variable (e.g., $X_t = x_t$). We use an asterisk $*$ to indicate a constant quantity (e.g., a fixed treatment a_t^*). We denote the patient history by $H_t = (\overline{X}_t, \overline{A}_{t-1}, \overline{Y}_{t-1})$.

CATE estimation over time: We build upon the potential outcome framework [30] for the time-varying setting [29]. We aim to estimate the CATE over time between two treatment sequences for a given patient history, i.e.,

$$\mathbb{E}\left[Y_{t+\tau}^{(a_{t:t+\tau}^*)} - Y_{t+\tau}^{(b_{t:t+\tau}^*)} \;\middle|\; H_t = h_t\right], \quad 0 \leq t \leq T - \tau, \tag{1}$$

where $Y_{t+\tau}^{(a_{t:t+\tau}^*)}$ and $Y_{t+\tau}^{(b_{t:t+\tau}^*)}$ represents the τ-step-ahead potential outcomes under interventions $do(A_{t:t+\tau} = a_{t:t+\tau}^*)$ and $do(A_{t:t+\tau} = b_{t:t+\tau}^*)$, respectively (see Appendix A for a formal definition of potential outcomes and interventions).

Identifiability: We make the following identifiability assumptions [24, 25] that are standard in the time-varying setting [3, 19, 20, 31]: (1) *Consistency*: The potential outcome under the intervention by the observed treatment equals the observed outcome, namely, $Y_t^{(A_t)} = Y$. (2) *Overlap*: Given an observed history $H_t = h_t$, if $p(H_t = h_t) > 0$, then any possible treatment has a positive probability of being received: $\forall a \in \mathcal{A}_t, p(A_t = a \mid H_t = h_t) > 0$. (3) *Sequential ignorability*: The potential outcome under an arbitrary intervention is independent of the treatment assignment conditioned on the history, i.e., $Y_t^{(a_t^*)} \perp A_t \mid H_t = h_t$.

However, estimating the CATE over time is non-trivial due to *time-varying confounding* [6, 14]. In the time-varying setting, covariates act as confounders because they are influenced by earlier treatments and affect later treatments. However, these time-varying confounders are *unobserved*, because of which naïve adjustments as in the static setting are impossible (see Appendix A.2). Here, we adjust for time-varying confounding through the use of SNMMs.

Blip function: SNMMs model the incremental effect of treatments (which are called "blips") at time $t + k$ on the mean outcome at $t + \tau$, given observed patient history H_t [36]. These "blips" accumulate over time into the total treatment effect and thus allow to rigorously adjust for time-varying confounding [28] (see Appendix C.1 for details). Formally, the blips are defined via a **blip function** [27, 28]:

$$\begin{aligned} \gamma_{t,k}\big(\bar{x}_{t+1:t+k}, \bar{a}_{t:t+k}\,; h_t\big) \;=\; \mathbb{E}\Big[&Y_{t+\tau}^{(a_{t:t+k},\, d_{t+k+1:t+\tau})} - Y_{t+\tau}^{(a_{t:t+k-1},\, 0,\, d_{t+k+1:t+\tau})} \\ &\Big|\; A_{t:t+k} = a_{t:t+k}, X_{t+1:t+k} = x_{t+1:t+k}, H_t = h_t\Big]. \end{aligned} \tag{2}$$

Intuitively, the blip function $\gamma_{t,k}$ isolates the causal effect of each treatment decision *locally*. This breaks the sequential dependencies over long temporal dependencies and thus is more stable over long horizons (→ thus addressing limitation **②**).

Nevertheless, SNMMs offer *only* a theoretical framework for identifying CATEs – they are *not* ready-to-use algorithms or models. Hence, implementing SNMMs with neural networks in particular is non-trivial: this requires a tailored learning objective that allows for neural parameterization and that supports efficient, end-to-end training and inference, which is the contribution of our DeepBlip.

4 Our DeepBlip Framework

In this section, we present DeepBlip. First, we introduce how we learn the CATE via blip functions using a neural parameterization (Sec. 4.1), then introduce our L^1-moment loss (Sec. 4.2), our model architecture (Sec. 4.3), and the training and inference procedure (Sec. 4.4).

4.1 Learning the CATE via blip functions

Overview: Our DeepBlip leverages Eq. (3) to adjust for time-varying confounding (→ thus addressing limitation ②). Our task thus reduces to estimating the blip functions – in particular, so-called *blip coefficients* that parametrize the blip functions. However, we do **not** attempt to estimate the coefficients directly. Instead, we optimize a L^1-moment loss that directly predicts the blip coefficients and which allows us to estimate Eq. (3) more efficiently.

Parameterization trick: We first explain how we estimate the CATE via the blip function. For this, we adopt a similar parametrization for the blip function as in [18], namely, $\gamma_{t,k}\big(\bar{x}_{t+1:t+k}, \bar{a}_{t:t+k}\,; h_t\big) \;=\; \psi_{t,k}\big(h_t\big)' a_{t+k}$, but where $\psi_{t,k}$ is a **neural network**. Under identifiability assumptions and the parametrization for $\gamma_{t,k}$ defined above, the CATE of a^* against b^* for any two treatment sequences $a^*, b^* \in \mathbb{R}^{(\tau+1)\cdot d_a}$ is (see [18] for a formal derivation):

$$\mathbb{E}\big[Y_{t+\tau}^{(a^*)} - Y_{t+\tau}^{(b^*)} \mid H_t = h_t\big] \;=\; \sum_{k=0}^{\tau} \psi_{t,k}(h_t)'\big(a^*_{t+k} - b^*_{t+k}\big). \tag{3}$$

We refer to $\psi_{t,k}(h_t)$ as the conditional *blip coefficients* of the blip function $\gamma_{t,k}$.

Why do we need a tailored architecture and learning algorithm? A key component of our framework is that the function $\psi_{t,k}(h_t)$ is parameterized by a sequential neural network (e.g., LSTM or transformer). This is a crucial difference from traditional SNMMs, which were developed for estimating the ATE over a fixed number of time steps (i.e. $\mathcal{H}_t = \emptyset \wedge t \equiv 0 \wedge T \equiv \tau$) and where, as a result, blip coefficients are constants. These constants are typically estimated through *iteratively* solving a set of moment equations via g-estimation [27, 28, 36] (see Appendix C.1). However, such an approach is not compatible with neural network-based learning. In contrast, DeepBlip introduces a tailored neural architecture (Sec. 4.3) that we can train via gradient-based optimization (Sec. 4.2).

4.2 L^1-moment loss

We reformulate the moment-based linear equations from [18] as an equivalent iterative minimization problem for $k = \tau, \ldots, k = 0$. At each time step k, we aim to find the minimizer $\psi^*_{t,k}(\cdot)$, which is a function that maps the history h_t to the blip coefficients, via

$$\psi^*_{t,k} = \underset{\widehat{\psi}_{t,k}(\cdot)\in\mathbf{\Phi}_{t,k}}{\arg\min}\; \mathbb{E}\Big[\Big\|\mathbb{E}\big[\big(\widetilde{Y}_{t,k} - \sum_{j=k+1}^{\tau} \psi^*_{t,j}(h_t)'\widetilde{A}_{t,j,k} - \widehat{\psi}_{t,k}(h_t)'\widetilde{A}_{t,k,k}\big)\widetilde{A}_{t,k,k} \mid H_t\big]\Big\|_1\Big], \tag{4}$$

where $\mathbf{\Phi}_{t,k}$ is the function space for the blip coefficient predictors and $\|\cdot\|_1$ is the L^1-norm operator. The expectation outside is taken over all random variables H_t. We name the target as the L^1**-moment loss**.

Here, we employ an L^1 loss for empirical reasons (see our ablation studies in Appendix D.2). The reason is that the moment has a high variance, especially with growing time horizon τ and due to the mini-batch sampling, which introduces another source of variance later. As a result, the L^1 loss is beneficial since it is more robust to such variance.

Theoretical properties: Below, we first show that the loss recovers the ground-truth blip coefficients. Then, we show that our loss is *Neyman-orthogonality* (see [5] for formal definition), which ensures double robustness. This means that the target loss is *robust* against perturbations of the nuisance functions [5, 16].

Remark 1. *If* $\forall 0 \le k \le \tau$, $\psi_{t,k} \in \mathbf{\Phi_{t,k}}$, *then the solution of the risk minimization scheme in Eq.* (4) *given by* $(\psi^*_{t,0}, \ldots, \psi^*_{t,\tau})$, *yields the ground-truth blip coefficients. That is,* $\psi^*_{t,k} = \psi_{t,k}$.

Remark 2. *The moment loss is Neyman-orthogonal.*

The above remarks follow from the theory in [18], which is easy to extend to our setting (see Appendix C.4.1 and Appendix C.4.1, respectively).

Double optimization trick for our L^1**-moment loss:** In order to find $\psi^*_{t,k}$, all the previous blip predictors $\psi^*_{t,j}, j \ge k$ are required. However, the ground-truth predictors are generally not available at the beginning. To avoid solving $\psi^*_{t,k}$ sequentially, we propose a *double optimization trick* that allows *simultaneous* training of all the blip predictors: During each iteration, first, the blip predictor

$\widehat{\boldsymbol{\psi}}_t$ makes two forward passes to generate two sets of the blip coefficients $\widehat{\boldsymbol{\psi}}_t^1(h_t)$ and $\widehat{\boldsymbol{\psi}}_t^2(h_t)$. Then, $\widehat{\psi}_{t,j}^2(h_t)$ is treated as the pseudo blip effects that replaces $\psi_{t,j}^*(h_t)$ in Eq. (4). For $k = 0, \ldots, \tau$, the adapted L^1-moment loss at step k is then given empirically by

$$\mathcal{L}_{\text{blip}}^k = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \Big\| \sum_{i=1}^{n} (\widetilde{Y}_{t+k}^i - \sum_{j=k+1}^{\tau} \widehat{\psi}_j^2(H_t^i)' \, \widetilde{A}_{t,j,k}^i - \widehat{\psi}_k^1(H_t^i)' \, \widetilde{A}_{t,k,k}^i) \cdot \widetilde{A}_{t,k,k}^i \Big\|_1 . \quad (5)$$

4.3 Model architecture

DeepBlip works in two stages (see Fig. 2): • **Stage ①** (*nuisance network*): models the nuisance functions to estimate the residuals in Eq. (6). • **Stage ②** (*blip prediction network*): estimates the blip coefficients given the observed history h_t. The neural networks in both stages have a similar structure: (i) a *sequential encoder* that encodes the observed history H_t, and (ii) multiple *prediction heads* that take the encoded history as input to predict the targets.

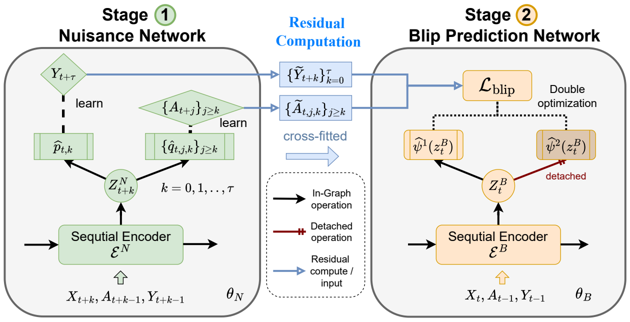

Figure 2: **Neural architecture of the two-stage DeepBlip framework**.

Why we need a two-stage design: To construct the L^1-moment loss defined Eq. (4), we need the variables $\widetilde{Y}_{t,k}$, $\widetilde{A}_{t,j,k}$, defined as (see Appendix C.1 for details):

$$\widetilde{Y}_{t,k} = Y_{t+\tau} - \mathbb{E}\big[Y_{t+\tau} \mid H_{t+k} = h_{t+k}\big], \quad \widetilde{A}_{t,j,k} = A_{t+j} - \mathbb{E}\big[A_{t+j} \mid H_{t+k} = h_{t+k}\big]. \quad (6)$$

As we see here, we must compute the *residuals* between the outcome variable and their regressed means *before* we optimize $\mathcal{L}_{\text{blip}}^k$. We thus follow previous literature [11, 16, 18] and treat the conditional expectations $\mathbb{E}\big[Y_{t+\tau} \mid H_{t+k} = h_{t+k}\big]$ and $\mathbb{E}\big[A_{t+j} \mid H_{t+k} = h_{t+k}\big]$ as *nuisance functions*, so that we first estimate the nuisance function (Stage ①) and the train the blip prediction network to minimize the L^1-moment loss (Stage ②).

Neural backbone: Our DeepBlip is flexible and allows for different neural backbones (e.g., LSTM or transformer). These necessary to capture the patient history: We notice that the networks at both Stage ① and ② take the history variable $H_t = (\overline{X}_t . \overline{A}_{t-1}, \overline{Y}_{t-1}) \in \mathcal{H}_t$ as input. Hence, both stages can be written as a function $f : \cup_{t=1}^{T} \mathcal{H}_t \to \mathbb{R}^c$, where c is the number of outputs. However, $\dim(H_t)$ varies over time, which makes H_t not suitable as a direct input to a neural network. A standard way to handle this is by using a sequential model to iteratively take the inputs (X_t, A_{t-1}, Y_{t-1}) and then maintain a vector $Z_t \in \mathbb{R}^{d_z}$ with fixed dimension that encodes all the necessary information [14, 19, 20, 31]. Here, we thus use LSTMs and transformers (see details of the architectures in Appendix H). Finally, we stress that each stage uses a *separate* encoder: $\mathcal{E}_{\theta_N}^N$ for Stage ①, and $\mathcal{E}_{\theta_B}^B$ for Stage ② (N for **N**uisance and B for **B**lip) with different model weights θ_N and θ_B.

Stage ①: nuisance network. The nuisance network $(\mathcal{E}_{\theta_N}^N, \{\text{gp}_{\theta_N}^k\}_{k=0}^{\tau}, \{\text{gq}_{\theta_N}^{j,k}\}_{0\leq k\leq j\leq\tau})$ consists of a seqential encoder $\mathcal{E}_{\theta_N}^N$ and a collection of prediction heads $\{\text{gp}_{\theta_N}^k\}_{k=0}^{\tau}, \{\text{gq}_{\theta_N}^{j,k}\}_{0\leq k\leq j\leq\tau}$. The nuisance networks are responsible for computing the following nuisance functions:

$$p_{t,k}(h_{t+k}) := \mathbb{E}\big[Y_{t+\tau} \mid H_{t+k} = h_{t+k}\big], \quad 1 \leq t \leq T-\tau, 0 \leq k \leq \tau \quad (7)$$

$$q_{t,j,k}(h_{t+k}) := \mathbb{E}\big[Q_{t,j} \mid H_{t+k} = h_{t+k}\big], 1 \leq t \leq T-\tau, 0 \leq k \leq j \leq \tau \quad (8)$$

For a patient with history H_t and subsequent covariates $X_{t+1:t+k}, A_{t:t+k-1}$, we proceed as follows: First, the encoder $\mathcal{E}_{\theta_N}^N$ learns the representation at time $t+k$ (note that $H_{t+k} = H_t \cup X_{t+1:t+\tau} \cup A_{t:t+\tau-1}$), which is given by $Z_{t+k}^N = \mathcal{E}_{\theta}^N(H_{t+k})$. Second, the prediction heads receive Z_{t+k}^N to compute the regressed outcomes for the nuisance functions via:

$$\text{gp}_{\theta_N}^k(Z_{t+k}^N) = \widehat{p}_{t,k}(H_{t+k}), \quad \text{gq}_{\theta_N}^{j,k}(Z_{t+k}^N) = \widehat{q}_{t,j,k}(H_{t+k}) \quad \text{for } k = 0, \ldots \tau \text{ and } k \leq j \leq \tau \quad (9)$$

where $Z^N_{t+k} = \mathcal{E}^N_{\theta_N}(H_{t+k})$. Third, the residuals are computed via

$$\widetilde{Y}_{t,k} \approx Y_{t+\tau} - \mathrm{gp}^k_{\theta_N}(Z^N_{t+k}), \quad \widetilde{A}_{t,j,k} \approx A_{t+j} - \mathrm{gq}^{j,k}_{\theta_N}(Z^N_{t+k}) \tag{10}$$

<u>**Stage ②: blip prediction network**</u>. The blip prediction network $\left(\mathcal{E}^B_{\theta_B}, \{\mathrm{gb}^k_{\theta_B}\}^{\tau}_{k=0}\right)$ is responsible for predicting the blip coefficients $\boldsymbol{\psi_t}(h_t) = (\psi_{t,0}, \ldots, \psi_{t,\tau}) \in \mathbb{R}^{r(\tau+1)}$ as described in Eq. (4). Here, we proceed as follows. First, the sequential encoder $\mathcal{E}^B_\theta$ (B for **B**lip) processes the patient's history H_t into a representation $Z^B_t = \mathcal{E}^B_\theta(h_t)$. Then, for each horizon $k \in \{0, 1, \ldots, \tau\}$, the prediction head $\mathrm{gb}^k_{\theta_B}$ maps Z^B_t onto the corresponding blip coefficient:

$$\widehat{\psi}_{t,k}(H_t) = \mathrm{gb}^k_{\theta_B}(Z^B_t) \sim \psi_{t,k}(H_t) \in \mathbb{R}^r \quad \text{where } Z^B_t = \mathcal{E}^B_{\theta_B}(H_t). \tag{11}$$

4.4 Training and Inference

Taken together, the training procedure of DeepBlip now follows two steps (see Fig. 2): (1) train the nuisance networks and compute the residuals, and (2) train the blip prediction network. In contrast, inference with DeepBlip is highly efficient as it involves *only* the second-stage blip prediction network. Details are below. We provide the pseudocode in Alg. 1 and Alg. 2 in the appendix.

Step ①: Train nuisance network. The nuisance network is trained to predict nuisance functions $p_{t,k}(h_{t+k})$ and $q_{t,j,k}(h_{t+k})$ simultaneously. Since $p_{t,k}(h_{t+k})$ is the conditional expectation of real outcome $Y_{t+\tau} \in \mathbb{R}$, we use the squared error loss $\mathcal{L}_p = \frac{1}{(T-\tau)(\tau+1)} \sum_{t=1}^{T-\tau} \sum_{k=0}^{\tau} (\mathrm{gp}^k_{\theta_N}(Z^N_{t+k}) - Y_{t+\tau})^2$. For $q_{t,j,k}(h_{t+k})$, which denotes the treatment response, we proceed for the i-th treatment in $A_{t+j} \in \mathbb{R}^{d_a}$ as follows. If $(A_{t+j})_i$ is a continuous variable, then we apply the squared loss: $\mathcal{L}_{q,i} = \frac{2}{(T-\tau)(\tau+1)(\tau+2)} \sum_{t=1}^{T-\tau} \sum_{0 \le k \le j \le \tau} \left(\mathrm{gq}^{k,j}_{\theta_N}(Z^N_{t+k})_i - (A_{t+j})_i)\right)^2$. If $(A_{t+j})_i$ is a binary variable, then we apply the binary cross entropy loss $\mathcal{L}_{q,i} = \frac{2}{(T-\tau)(\tau+1)(\tau+2)} \sum_{t=1}^{T-\tau} \sum_{0 \le k \le j \le \tau} \mathrm{BCE}\big((A_{t+j})_i\ ,\ \mathrm{gq}^{k,j}_{\theta_N}(Z^N_{t+k})_i\big)$. For categorical variables with more than 2 classes, we preprocess the variable into a one-hot vector of binary variables. Since the network predicts these targets simultaneously, we update the parameter θ_N by backpropagating the sum of all the losses discussed above, i.e., $\mathcal{L}_N = \mathcal{L}_p + \frac{1}{d_a} \sum_{i=1}^{d_a} \mathcal{L}_{q,i}$

Step ②: Train blip prediction network. After having trained the nuisance network, we freeze its parameters and then compute the residuals of each sample as in Eq. (6) for the L^1-moment loss. To accelerate the training process, we adopt the double optimization trick from above: For $k = 0, \ldots, \tau.$, we perform two forward passes that create two predictions $\widehat{\boldsymbol{\psi}}^1_{t,k}(H_t)$ and $\widehat{\boldsymbol{\psi}}^2_{t,k}(H_t)$. The latter is then *detached* from the computation graph before feeding into the adapted L^1-moment loss at step k. The final loss target is then given by: $\mathcal{L}_{\text{blip}} = \sum_{k=0}^{\tau} \mathcal{L}^k_{\text{blip}}$. We note that, for $k = \tau$, there is **no** detached blip coefficients term in Eq. (5). Hence, $\widehat{\psi}^1_{t,\tau}(H_t)$ is directly supervised by the true L^1-moment loss and can directly learn the ground-truth. As such, $\mathcal{L}^{\tau-1}_{\text{blip}}$ gradually approximates the true L^1-moment loss, which then supervises $\widehat{\psi}^1_{t,\tau-1}(H_t)$, and so on. As a result, all prediction heads will gradually learn to predict the blip coefficients from $t = \tau$ to $t = 0$.

Remark 3. *Under standard assumptions, the output of our DeepBlip has a mean squared error guarantee:*

$$\max_{t \le T-\tau} \max_{k \in \{0,\ldots,\tau\}} \mathbb{E}\left[\left\|\widehat{\psi}_{t,k} - \psi_{t,k}\right\|^2_{2,2}\right] = O\left(r^2 \delta_n^2\right), \quad \delta_n^2 \propto \frac{\log\log(n)}{n} \tag{12}$$

The adove is adopted from SNMM methods [18] that were originally developed for linear models, yet we offer a neural instantiation. Details are in Appendix C.5.1).

Inference at runtime: Once trained, our DeepBlip predicts the CATE over time (i.e., $\mathbb{E}[Y^{(a^*)}_{t+\tau} - Y^{(b^*)}_{t+\tau} \mid H_t = h_t]$) through only the blip prediction network:

$$\sum_{k=0}^{\tau} \mathrm{gb}^k_{\theta_B}(z^B_t)'(a^*_k - b^*_k), \quad \text{where} \quad z^B_t = \mathcal{E}^B_{\theta_B}(h_t) \tag{13}$$

Efficient offline evaluation: Once we have estimated the blip coefficients, then we can instantly identify the treatment sequence a^* with the best effect compared to the baseline b^* (e.g., a treatment sequence with no interventions). The reason is that the blip coefficients do *not* depend on treatments. Hence, our DeepBlip is much more efficient for evaluating the personalized effects of different treatment sequences compared to existing methods that require re-computation [3, 19, 20, 31] or even re-training [11, 14]. This is highly relevant in personalized medicine where clinicians and patients jointly reason about different treatment strategies [8].

Implementation details. We instantiate our DeepBlip with a transformer architecture (see Appendix H). We also provide a variant based on an LSTM, which, despite the simpler architecture, is sill highly competitive (see Appendix D.1).

5 Experiments

Baselines: We demonstrate the performance of our DeepBlip against key baselines from the literature (see Table 1) for the task of estimating CATE (or conditional average potential outcomes) on medical datasets. Descriptions of the baseline methods are available in Appendix F. We further select the HA-PI-learner from [11] instantiated by transformer (named **HA-TRM**) as a naïve baseline. We provide additional implementation details – including architecture choices, training procedures, and hyperparameter tuning – in Appendix H. To ensure a fair comparison, all methods – including baselines – use the **same** neural backbone architecture, so any performance differences must be attributed solely to that our learning objective is better (i.e., unbiased and stable over longer time horizons). All results are averaged over 5 runs.

Ablations: We include ablation studies in Appendix D.1, where we validate our component-wise blip coefficient estimates in Appendix D.2. We also provide an instantiation of our DeepBlip with an LSTM instead of a transformer. Importantly, even our ablation is highly competitive and outperforms the majority of transformer-based baselines (see Appendix D.1).

5.1 Tumor growth dataset

Setting: We use the pharmacokinetic-pharmacodynamic tumor growth dataset [12], which is commonly used for benchmaing CATE methods over time [3, 14, 19, 20, 31]. The dataset describes the time-varying effects of chemotherapies and radiotherapies, for which treatment assignments depend on previous outcomes, subject to time-varying confounding. The amount of confounding is controlled by the simulation parameter γ_{conf}. Details are in Appendix E.1.

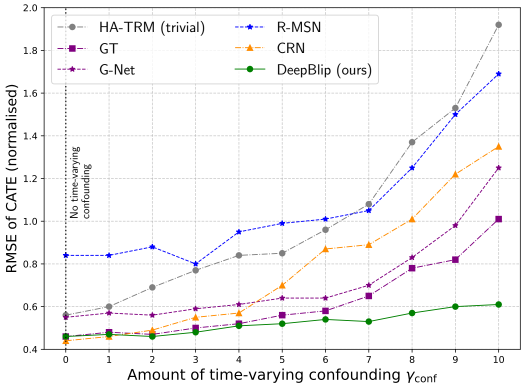

Figure 3: **Results for tumor growth dataset.** Normalized RMSE (averaged over 5 runs) of CATE predictions against ground-truth over growing confounding. Here: $\tau = 2$

Results: Figure 3 shows the average RMSE of CATE against increasing confounding γ_{conf} and under $\tau = 2$. Our **DeepBlip** outperforms all baselines for $\gamma_{\text{conf}} \geq 2$. This matches the purpose of our method to deal with time-varying confounding. More importantly, our DeepBlip achieves large performance gains under strong confounding levels ($\gamma_{\text{conf}} > 6$). This highlights that our DeepBlip is robust against time-varying confounding by providing adequate adjustment for time-varying confounding.

We could further make the following observations: ① The MSM-based **R-MSN** performs poorly across all confounding levels and even has a higher RMSE than HA-LSTM for $\gamma_{\text{conf}} \leq 6$. This aligns with the inverse propensity weighting in MSMs is highly unstable, which was the motivation for our method. ② Baselines like **CT** and **CRN** that use balanced representation (in orange) are ineffective. This is expected as balanced representations were originally developed for reducing finite-sample estimation variance and *not* for proper adjustment (see the original work [33] on balanced representations for a discussion), because of which these baselines are known to be biased. ③ G-computation-based methods like **G-Net** and **GT** show slightly lower RMSE for $\gamma_{\text{conf}} \leq 1$ but still perform significantly worse than our DeepBlip for $\gamma_{\text{conf}} \geq 6$. We attribute this to the fact that the learning is unstable, which we empirically verify in the following by varying the prediction horizon τ.

5.2 MIMIC-III dataset

Setting: Next, we evaluate the performance for longer prediction horizons τ. We build upon MIMIC-III [15], a widely used benchmark for evaluating CATE over time [3, 14, 20]. Following previous literature [14, 20, 32], we extract patient vitals from MIMIC-III and then simulate the patient outcome over time with the mixed dynamics of exogenous dependency, endogenous dependency, and treatment effects combined. Treatments are assigned based on previous outcomes and patient vitals, which again, introduces time-varying confounding (see Appendix E.2).

Results: Table 2 shows the average RMSE (with std. dev.) over five different runs with $\gamma_{\text{conf}} = 1$ and varying prediction horizon τ. First, our **DeepBlip** consistently achieves lower RMSE compared to other baselines for $\tau \geq 2$. Of note, the performance gain from DeepBlip becomes larger as γ_{conf} increases. For $\tau = 10$, DeepBlip achieves $\sim 38\%$ performance gain compared to the second-best model (here: GT). This highlights that our DeepBlip is temporally stable over longer horizons.

We further make the observations that all baselines either struggle with high-dimensional covariates or become unstable as τ increases. ① The MSM-based method (**R-MSNs**) exhibits the highest variance across all τ, indicating that it struggles with high-dimensional propensity modeling and becomes unstable over time with increasing standard deviation. The reason is that inverse propensity weighting produces unstable weights. ② Methods like **CRN** and **CT** perform better than baselines with high variance like R-MSNs due to the way they handle high-dimensional covariates. However, both **CRN** and **CT** are known to be biased and thus inferior to GT and our DeepBlip. ④ G-computation-based methods (i.e., **G-Net** and **GT**) achieve a lower RMSE than the other baselines due to proper adjustments, but still are not as stable as our method. This is because G-computation accumulates error over time due to modeling nested expectations.

	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$	$\tau = 7$	$\tau = 8$	$\tau = 9$	$\tau = 10$
HA-TRM (na"ıve) [11]	0.68 ± 0.02	0.89 ± 0.03	0.97 ± 0.04	1.02 ± 0.10	1.42 ± 0.20	1.92 ± 0.40	2.57 ± 0.44	2.58 ± 0.56	3.11 ± 0.72
R-MSNs [19]	0.73 ± 0.14	0.98 ± 0.17	1.12 ± 0.21	1.25 ± 0.28	1.65 ± 0.57	2.25 ± 1.02	2.85 ± 1.18	3.20 ± 1.42	3.55 ± 1.50
CRN [3]	0.49 ± 0.05	0.66 ± 0.11	0.82 ± 0.12	1.05 ± 0.22	1.22 ± 0.35	1.43 ± 0.33	1.62 ± 0.42	1.83 ± 0.43	2.04 ± 0.54
CT [20]	0.52 ± 0.07	0.64 ± 0.12	0.79 ± 0.11	1.01 ± 0.18	1.18 ± 0.33	1.77 ± 0.52	1.85 ± 0.49	1.99 ± 0.63	1.98 ± 0.60
G-Net [31]	0.42 ± 0.05	0.58 ± 0.08	0.73 ± 0.12	1.05 ± 0.25	1.38 ± 0.40	1.75 ± 0.60	2.15 ± 0.80	2.55 ± 0.90	3.12 ± 1.05
GT [14]	0.40 ± 0.01	0.52 ± 0.02	0.63 ± 0.08	0.75 ± 0.17	0.85 ± 0.13	0.95 ± 0.26	1.10 ± 0.34	1.25 ± 0.37	1.50 ± 0.45
DeepBlip (**ours**)	$\mathbf{0.39} \pm \mathbf{0.11}$	$\mathbf{0.48} \pm \mathbf{0.12}$	$\mathbf{0.56} \pm \mathbf{0.16}$	$\mathbf{0.64} \pm \mathbf{0.19}$	$\mathbf{0.70} \pm \mathbf{0.21}$	$\mathbf{0.79} \pm \mathbf{0.24}$	$\mathbf{0.82} \pm \mathbf{0.27}$	$\mathbf{0.88} \pm \mathbf{0.28}$	$\mathbf{0.93} \pm \mathbf{0.32}$
Improvement	2.5%	7.6%	11.1%	14.7%	17.6%	16.8%	25.5%	29.6%	38.0%

Table 2: **MIMIC-III with longer time horizons τ.** Normalized RMSE (mean ± std. dev. over 5 runs) for τ-step-ahead CATE estimation on the MIMIC-III dataset. We highlight the relative improvement over the best-performing baseline. $\Rightarrow$ Our DeepBlip consistently outperforms the baselines for $\tau \geq 2$.

6 Discussion

Limitations: (1) Our work is subject to the standard assumptions for treatment effect estimation, which are standard in the literature [3, 11, 14, 19, 20, 27, 28, 31]. (2) Our work is further subject to the characteristics of how the blip function is parameterized. (3) The overall training cost is comparable to that of the baseline. Importantly, the runtime ($\sim$ 30 min, see Appendix H) is similar across all baselines, and in practice, we often observe faster convergence due to the more stable learning of our approach. Importantly, the computational cost is typically not a major concern in medical applications, as the models are trained only once, and all baselines scale efficiently to all real-world medical datasets from practice.

Broader impact: We expect our contribution to have a significant impact on *reliable* decision-making in personalized medicine. DeepBlip provides a *stable* learning framework for *efficient* offline evaluation of personalized treatment strategies over long time horizons.

Conclusion: We are the first to build a neural framework using blip functions to estimate CATEs over time.

References

[1] Ahmed M. Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task Gaussian processes.

[2] Ahmed Allam, Stefan Feuerriegel, Michael Rebhan, and Michael Krauthammer. Analyzing patient trajectories with artificial intelligence. *Journal of Medical Internet Research*, page e29812, 2021.

[3] Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. Estimating Counterfactual Treatment Outcomes over Time Through Adversarially Balanced Representations. In *ICLR*, 2020.

[4] Ioana Bica, Ahmed M. Alaa, Christoph Lambert, and Mihaela van der Schaar. From real-world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges. *Clinical Pharmacology & Therapeutics*, 2021.

[5] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/Debiased Machine Learning for Treatment and Causal Parameters. *The Econometrics Journal*, 2018.

[6] Amanda Coston, Edward Kennedy, and Alexandra Chouldechova. Counterfactual predictions under runtime confounding. In *Advances in Neural Information Processing Systems*, 2020.

[7] Alicia Curth and Mihaela van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, 13–15 Apr 2021.

[8] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 04 2024.

[9] Dylan J. Foster and Vasilis Syrgkanis. Orthogonal statistical learning. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, pages 1362–1385, 2019. arXiv preprint arXiv:1901.09036.

[10] Dennis Frauen, Tobias Hatt, Valentyn Melnychuk, and Stefan Feuerriegel. Estimating average causal effects from patient trajectories. 37:7586–7594, Jun. 2023.

[11] Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. Model-agnostic meta-learners for estimating heterogeneous treatment effects over time. In *International Conference on Learning Representations (ICLR)*, 2025.

[12] Changran Geng, Harald Paganetti, and Clemens Grassberger. Prediction of Treatment Response for Combined Chemo- and Radiation Therapy for Non-Small Cell Lung Cancer Patients Using a Bio-Mathematical Model. *Scientific Reports*, 2017. Author correction published in Scientific Reports, 2018, 8(1):12631, doi:10.1038/s41598-018-30761-7.

[13] Miguel A. Hernán, Babette Brumback, and James M. Robins. Marginal structural models to estimate the joint causal effect of nonrandomized treatments. *Journal of the American Statistical Association*, 2001.

[14] Konstantin Hess, Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. G-transformer for conditional average potential outcome estimation over time. In *NeurIPS*, 2025. arXiv preprint arXiv:2405.21012.

[15] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Mahdi Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 2016.

[16] Edward H. Kennedy. Towards Optimal Doubly Robust Estimation of Heterogeneous Causal Effects. *Electronic Journal of Statistics*, 2023.

[17] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Meta-learners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116:4156–4165, 2019.

[18] Greg Lewis and Vasilis Syrgkanis. Double/Debiased Machine Learning for Dynamic Treatment Effects via g-Estimation. In *Advances in Neural Information Processing Systems*, Virtual, Dec 2021.

[19] Bryan Lim. Forecasting treatment responses over time using recurrent marginal structural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[20] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal Transformer for Estimating Counterfactual Outcomes. In *ICML*, 2022.

[21] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Bounds on representation-induced confounding bias for treatment effect estimation. In *ICLR*, 2024. arXiv:2311.11321.

[22] Liliana Orellana, Andrea Rotnitzky, and James M. Robins. Dynamic regime marginal structural mean models for estimation of optimal dynamic treatment regimes, part i: main content. *The International Journal of Biostatistics*, 2010.

[23] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[24] Maya L. Petersen and Mark J. van der Laan. Causal models and learning from data: integrating causal modeling and statistical estimation. *Epidemiology*, 2014.

[25] J. M. Robins, M. A. Hernan, and B. Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 2000.

[26] James Robins. A new approach to causal inference in mortality studies with a sustained exposure period. *Mathematical Modelling*, January 1986.

[27] James M. Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics - Theory and Methods*, 1994.

[28] James M. Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the Second Seattle Symposium in Biostatistics*. 2004.

[29] James M. Robins, Sander Greenland, and Fu-Chang Hu. Estimation of the causal effect of a time-varying exposure on the marginal mean of a repeated binary outcome. *Journal of the American Statistical Association*, 1999.

[30] Donald Rubin. Estimating causal effects of treatments in experimental and observational studies. *ETS Research Bulletin Series*, 1972.

[31] Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M. Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, and Li-wei Lehman. G-Net: a recurrent network approach to G-Computation for counterfactual prediction under a dynamic treatment regime. *Machine Learning for Health*, 2021.

[32] Peter Schulam and Suchi Saria. Reliable Decision Support using Counterfactual Models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6076–6086, Long Beach, CA, USA, 04–09 Dec 2017. Curran Associates, Inc.

[33] Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating Individual Treatment Effect: Generalization Bounds and Algorithms. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3076–3085. PMLR, 06–11 Aug 2017.

[34] Toru Shirakawa, Yi Li, Yulun Wu, Sky Qiu, Yuxuan Li, Mingduo Zhao, Hiroyasu Iso, and Mark van der Laan. Longitudinal targeted minimum loss-based estimation with temporal-difference heterogeneous transformer. *arXiv preprint arXiv:2404.04399*, apr 2024.

[35] Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. In *UAI*, Sydney, Australia, 2017. AUAI Press.

[36] Stijn Vansteelandt and Marshall Joffe. Structural nested models and g-estimation: The partially realized promise. *Statistical Science*, 29, November 2014.

[37] Stefan Wager and Susan Athey and. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 2018.

[38] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, volume 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, UK, 2019.

[39] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. MIMIC-Extract: a data extraction, preprocessing, and representation pipeline for MIMIC-III. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, ACM CHIL '20, April 2020.

[40] Yanbo Xu, Yanxun Xu, and Suchi Saria. A non-parametric bayesian approach for estimating treatment-response curves from sparse time series. In *ML4H*, Proceedings of Machine Learning Research, Northeastern University, Boston, MA, USA, aug 2016.

[41] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *ICLR*, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claims that our proposd DeepBlip framework addesses two key limitations as well as several practical strengths. These claims are supported by SNMM (section 3) the framework (section 4) and experiment (section 5), with both thoretical and empirical content.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations in Sec. 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide full and detailed assumptions in Sec. 3 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all the details needed to reproduce the experiemntal results in Appendix E, H, F, H.4 and H.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use open datasets from MIMIC in our experiment and provide the link to an anonymous github repo: https://anonymous.4open.science/r/DeepBlip-A39B.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We include all the information needed to reproduce the results, including dataset in Appendix E, implementation in Appendix H and hyperparameters in Appendix H.4,H.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes we report standard deviation in the results for MIMIC. For the tumor dataset we also take the average RMSE over five runs although std. dev is not directly visualized in the plot.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computing resources in Appendix H.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impact in the Sec. 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our work does not pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all the code and datasets used in our work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not provide any assets in our paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development for our work does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.