CAUSAL MAMBA FOR ESTIMATING COUNTERFACTUAL OUTCOMES OVER TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately and efficiently estimating counterfactual outcomes over time from observational data is essential in various applications. However, existing methods suffer from computational inefficiency and over-balancing, limiting their practical use. In this paper, we propose Causal Mamba (CM), a novel end-to-end framework for efficient counterfactual estimation over time. Its GCSI-Mamba backbone introduces Gated Cross-Sequence Interaction (GCSI) and follows an interact-then-propagate paradigm, first modeling dependencies among treatment, covariate, and outcome streams and then using stream specific Mamba branches to capture their respective temporal dynamics, thereby preserving stream heterogeneity while retaining linear complexity with respect to sequence length. The Multi-Task Collaboration (MTC) module further coordinates outcome prediction, treatment domain confusion, and covariate reconstruction to reduce treatment specific bias while preserving relevant information for counterfactual prediction, avoiding information loss caused by over-balancing. Experimental results on synthetic and real-world datasets demonstrate that CM not only outperforms baselines but also achieves greater runtime efficiency.

1 INTRODUCTION

Estimating counterfactual outcomes over time from observational data is essential for decision making in fields such as personalized medicine (Feuerriegel et al., 2024), where evaluating the effect of sequential interventions is key to tailoring dynamic strategies (Mahar et al., 2021; Ye et al., 2023). Although randomized controlled trials (RCTs) remain the gold standard for causal inference (Liu et al., 2025), their high cost and ethical constraints restrict widespread adoption. Consequently, methods that leverage observational data, such as electronic health records (EHRs) (Allam et al., 2021), offer a widely accessible and ethical complement to RCTs.

A wide range of methods have been developed for counterfactual estimation in static observational settings (Shalit et al., 2017; Curth & Van der Schaar, 2021; Johansson et al., 2022; Melnychuk et al., 2023; Frauen et al., 2023b), yet the transition to longitudinal data presents acute challenges. The temporal dimension introduces complex dependencies (Loh & Ren, 2025; Rop et al., 2025) such as time-varying confounders, interactions between treatments, and long-term feedback effects on outcomes, which can lead to high generalization error (Alaa & van der Schaar, 2018) or even biased estimates (Frauen et al., 2023a) for traditional machine learning methods when directly predicting counterfactual outcomes. Moreover, the space of potential outcomes grows exponentially with the number of time steps, imposing greater demands on model runtime efficiency.

To address these challenges, state-of-the-art works have incorporated sequential debiasing mechanisms into model designs. Representative methods encompass recurrent marginal structural networks (RMSNs) (Lim et al., 2018), counterfactual recurrent networks (CRN) (Bica et al., 2020b), G-Net (Li et al., 2021), Causal Transformer (CT) (Melnychuk et al., 2022), and IGC-Net (Hess et al., 2026). However, representation balancing methods, such as CRN and CT, while effective in removing the association between patient history and treatment assignments, encounter the issue of over-balancing (Huang et al., 2024). Specifically, during the construction of balanced representations, essential covariate information can be lost, thereby degrading counterfactual estimation performance. Moreover, despite their ability to capture long-range dependencies, existing LSTM- or Transformer-based methods suffer from limited runtime efficiency.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

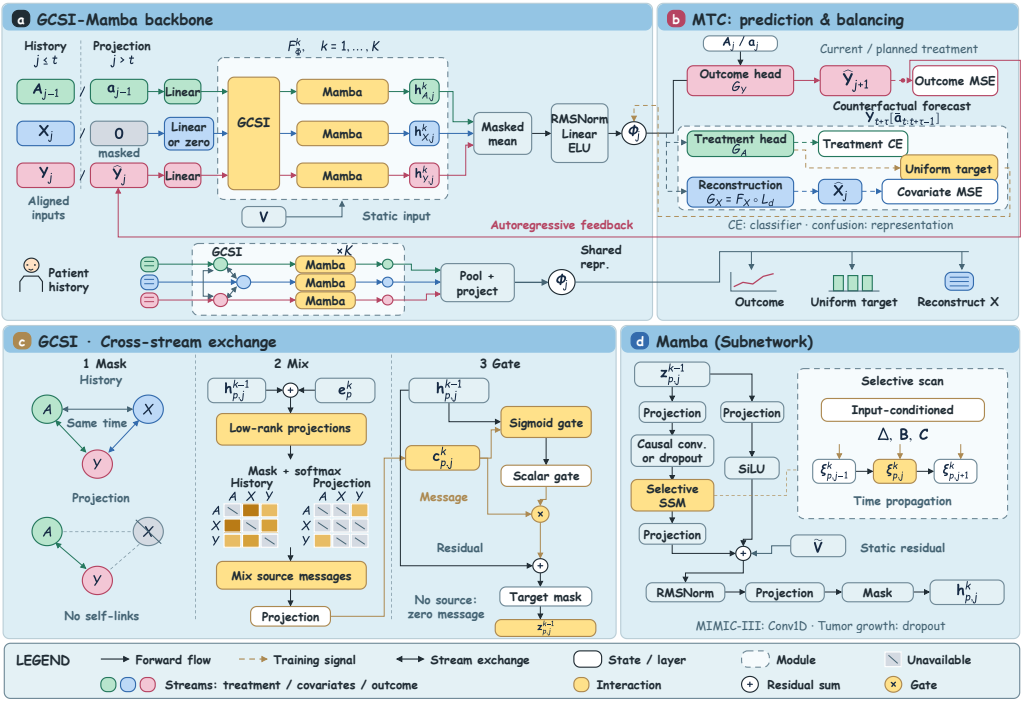


Figure 1: Overview of CM. The framework is built around two main modules: (a) The GCSI-Mamba backbone models aligned treatment, covariate, and outcome streams through cross-stream interaction and temporal propagation. (b) MTC jointly performs outcome prediction, treatment balancing, and covariate reconstruction. Panels (c) and (d) further illustrate the internal mechanisms of GCSI-based cross-stream interaction and the Mamba subnetwork, respectively.

For this, we develop Causal Mamba (CM) as illustrated in Figure 1, a novel end-to-end framework for counterfactual estimation over time. First, longitudinal counterfactual prediction is inherently asymmetric: planned treatments remain available during projection and future outcomes are recursively generated, whereas future factual covariates are unobserved. Motivated by this asymmetry, we introduce the GCSI-Mamba backbone, which employs separate Mamba subnetworks for treatment, covariate, and outcome streams rather than collapsing them into a unified trajectory. To retain interactions among these coupled processes, Gated Cross-Sequence Interaction (GCSI) incorporates information across streams that is aware of availability before Mamba propagation in each stream, allowing context across streams to influence subsequent state evolution without violating information constraints at projection time. The resulting architecture retains linear complexity in sequence length. Second, CM introduces a Multi-Task Collaboration (MTC) module to mitigate over-balancing by coupling balancing with information preservation. Rather than enforcing treatment invariance as an isolated objective, MTC coordinates three complementary tasks on the shared representation: outcome prediction preserves prognostic information, treatment domain confusion removes information that distinguishes the current treatment, and covariate reconstruction encourages retention of the observed covariate information. This collaborative objective discourages information loss caused by excessive balancing and provides explicit control over the tradeoff between confounding adjustment and information preservation.

Overall, our main **contributions** [1] are: (i) We propose the GCSI-Mamba backbone, an efficient multi-stream architecture. This design captures coupled longitudinal dependencies while preserving stream heterogeneity and maintaining linear complexity with respect to sequence length. (ii) We develop a Multi-Task Collaboration (MTC) module that addresses over-balancing by reducing the predictability of the current treatment while preserving information relevant to counterfactual prediction. (iii) Through extensive experiments on synthetic, semi-synthetic, and real-world benchmarks, we demonstrate that CM outperforms existing baselines while achieving lower runtime.

[1] Code is available at `https://anonymous.4open.science/r/causal-mamba/`.

2 Related Work

Estimating counterfactual outcomes in static settings. The focus of static methods is to adjust for observed confounding bias across populations through established strategies including re-weighting, matching, and covariate balancing (Kuang et al., 2020; Imai et al., 2023; Kong et al., 2023). However, these traditional approaches often struggle with high-dimensional data and complex nonlinear confounding structures. These limitations have motivated a shift toward deep learning methods (Wu et al., 2020; Nie et al., 2021; Li & Yao, 2022; Wang et al., 2022; Zhong et al., 2022; Tesei et al., 2023) for more flexible and accurate estimation.

Estimating counterfactual outcomes over time. Early methods for estimating time-varying treatment effects originated in epidemiology within classical statistical frameworks, including G-computation, marginal structural models (MSMs), and structural nested models (Robins, 1986; 1994; Robins et al., 2000; Robins & Hernan, 2008). Subsequent Bayesian non-parametric methods (Soleimani et al., 2017; Luo et al., 2025) were developed to capture temporal dependencies beyond the capacity of linear regression. However, these methods generally rely on strong assumptions about the data generation process, limiting their applicability in complex longitudinal settings.

State-of-the-art methods for time-varying counterfactual estimation leverage neural networks. Representative approaches include RMSNs (Lim et al., 2018), which incorporate inverse probability of treatment weighting within LSTM-based architectures; CRN (Bica et al., 2020b), which utilizes gradient reversal to learn balanced representations; and G-Net (Li et al., 2021), which integrates the G-computation framework with deep learning. Methods relying on LSTMs struggle to model long-range dependencies effectively. Subsequent Transformer-based models such as CT (Melnychuk et al., 2022) and IGC-Net (Hess et al., 2026) improve long-range dependency modeling. CT learns balanced representations, whereas IGC-Net incorporates G-computation to adjust for time-varying confounding. However, the quadratic computational complexity of Transformers introduces efficiency constraints, particularly with long sequences. Moreover, the balancing strategies employed in CRN and CT are prone to over-balancing. This issue stems from an excessive pursuit of treatment-invariant representations, which dilutes predictive information and thereby reduces the accuracy of counterfactual estimation.

Longitudinal causal inference and **state space models** are further discussed in Appendix A.

3 Problem Formulation

Our work builds upon the standard framework for estimating time-varying counterfactual outcomes and uses observational records from N individuals. Each individual i is observed over $T^{(i)}$ discrete time steps t, $t = 1, \ldots, T^{(i)}$. At each time step t, the observed data consist of a d_x-dimensional covariate vector $\mathbf{X}_t^{(i)} \in \mathbb{R}^{d_x}$ (e.g., vital signs), a categorical treatment $\mathbf{A}_t^{(i)} \in \{a_1, \ldots, a_{d_a}\}$ with d_a possible levels (e.g., medication regimens), and a d_y-dimensional outcome vector $\mathbf{Y}_t^{(i)} \in \mathbb{R}^{d_y}$ (e.g., physiological measurements). Static features $\mathbf{V}^{(i)}$ (e.g., demographic information) are also recorded per individual. For notational simplicity and consistency with prior literature (Robins & Hernan, 2008; Lim et al., 2018; Bica et al., 2020b; Li et al., 2021; Melnychuk et al., 2022; Hess et al., 2026), we generally omit the individual superscript (i) unless explicitly required.

Within the potential outcomes framework (Rubin, 1978) and its extension to time-varying treatments and outcomes, we denote the history up to time step t as $\bar{\mathbf{H}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{Y}}_t, \mathbf{V})$, which comprises sequential records of individual covariates $\bar{\mathbf{X}}_t = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$, treatments $\bar{\mathbf{A}}_{t-1} = (\mathbf{A}_1, \ldots, \mathbf{A}_{t-1})$, and outcomes $\bar{\mathbf{Y}}_t = (\mathbf{Y}_1, \ldots, \mathbf{Y}_t)$. For model training, we have access to i.i.d. observational data $\mathcal{D} = \left\{ \{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)}\}_{t=1}^{T^{(i)}}, \mathbf{v}^{(i)} \right\}_{i=1}^{N}$, such as EHRs commonly available in clinical practice. For a projection horizon $\tau \geq 1$, the objective of τ-step-ahead prediction is to estimate the potential outcome $\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}_{t:t+\tau-1}]$ under a non-random treatment intervention $\bar{\mathbf{a}}_{t:t+\tau-1} = (\mathbf{a}_t, \ldots, \mathbf{a}_{t+\tau-1})$, formalized as

$$\mathbb{E}(\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}_{t:t+\tau-1}] \mid \bar{\mathbf{H}}_t). \tag{1}$$

To ensure identifiability of treatment effects from observational data, following prior work (Bica et al., 2020b; Li et al., 2021; Melnychuk et al., 2022; Hess et al., 2026), we adopt the assumptions of consistency, sequential ignorability, and sequential overlap (see Appendix B for details).

4 CAUSAL MAMBA

The proposed framework, CM, is depicted in Figure 1. CM uses separate Mamba subnetworks coupled in every block by GCSI with availability masking. At representation time j, GCSI exchanges information among the aligned states of previous treatment $\mathbf{A}_{j-1}$, current covariates $\mathbf{X}_j$, and current outcome $\mathbf{Y}_j$ before the separate causal scans. Static covariates $\mathbf{V}$ remain shared inputs. During projection, planned treatment $\mathbf{a}_{j-1}$ and autoregressive outcome $\widehat{\mathbf{Y}}_j$ remain available, whereas the stream of future covariates is masked; hence $\mathbf{\Phi}_j$ uses neither $\mathbf{A}_j$ nor unavailable future factual variables. To estimate counterfactual outcomes following an intervention, the model is also provided with the intended future treatment intervention $\bar{\mathbf{a}}_{t:t+\tau-1}$. When performing 1-step-ahead prediction, the model draws primarily upon observational data. For multi-step-ahead prediction, it operates autoregressively, iteratively feeding its own forecasts of outcomes $\hat{\mathbf{Y}}_{t+1:t+\tau-1}$ back into the network from the intervention point onward.

4.1 GCSI-MAMBA BACKBONE

CM produces a sequence of shared representations $\bar{\mathbf{\Phi}}_{t+\tau-1} = (\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{t+\tau-1})$ through a cascade of GCSI-Mamba blocks. At the entrance of each block, GCSI first mixes the available stream states aligned at each time index; separate Mamba subnetworks then propagate the enriched inputs over time. The final block outputs stream-specific states, whose availability-weighted mean is projected to obtain $\bar{\mathbf{\Phi}}_{t+\tau-1}$.

Gated Cross-Sequence Interaction (GCSI). Although maintaining separate treatment, covariate, and outcome streams preserves their heterogeneous temporal dynamics and availability during projection, processing them independently would limit information exchange among these tightly coupled longitudinal processes. GCSI therefore exchanges information among the available streams aligned at the same time index before temporal propagation by their respective Mamba subnetworks. Let $\mathcal{P} \subseteq \{A, X, Y\}$ denote the streams present in a dataset, and let $\mathbf{h}_{p,j}^{k-1} \in \mathbb{R}^{d_h}$ be the incoming state of stream p at time j in block k. The availability indicator $m_{p,j} \in \{0, 1\}$ masks unavailable streams. During encoding of the observed history, GCSI aligns previous treatment $\mathbf{A}_{j-1}$, current covariates $\mathbf{X}_j$, and current outcome $\mathbf{Y}_j$. During projection, $m_{X,j} = 0$, so only planned treatment and autoregressive outcome streams interact.

To score relevance across streams, GCSI adds a learned stream embedding $\mathbf{e}_p^k$ and projects the states into a low-rank interaction space of dimension r:

$$\begin{gathered}\mathbf{q}_{p,j}^k = \mathbf{W}_Q^k(\mathbf{h}_{p,j}^{k-1} + \mathbf{e}_p^k), \quad \boldsymbol{\kappa}_{p,j}^k = \mathbf{W}_K^k(\mathbf{h}_{p,j}^{k-1} + \mathbf{e}_p^k), \quad \mathbf{v}_{p,j}^k = \mathbf{W}_V^k(\mathbf{h}_{p,j}^{k-1} + \mathbf{e}_p^k), \\ s_{p\leftarrow q,j}^k = \frac{(\mathbf{q}_{p,j}^k)^\top \boldsymbol{\kappa}_{q,j}^k}{\sqrt{r}}, \qquad q \neq p,\end{gathered} \tag{2}$$

where $\mathbf{W}_Q^k, \mathbf{W}_K^k, \mathbf{W}_V^k \in \mathbb{R}^{r\times d_h}$. For target stream p, the set of valid sources $\mathcal{S}_{p,j} = \{u \in \mathcal{P} : u \neq p,\ m_{p,j}m_{u,j} = 1\}$ excludes self-edges and unavailable streams. The cross-stream weights are

$$\alpha_{p\leftarrow q,j}^k = \begin{cases} \dfrac{\exp(s_{p\leftarrow q,j}^k)}{\sum_{u\in\mathcal{S}_{p,j}} \exp(s_{p\leftarrow u,j}^k)}, & q \in \mathcal{S}_{p,j}, \\ 0, & q \notin \mathcal{S}_{p,j}. \end{cases} \tag{3}$$

With $\mathbf{W}_O^k \in \mathbb{R}^{d_h\times r}$, a learned sigmoid gate controls how much of the aggregated message enters the target stream through a residual update:

$$\begin{gathered}\mathbf{c}_{p,j}^k = \mathbf{W}_O^k \sum_{q\neq p} \alpha_{p\leftarrow q,j}^k \mathbf{v}_{q,j}^k, \quad g_{p,j}^k = \sigma\big((\mathbf{w}_{g,p}^k)^\top[\mathbf{h}_{p,j}^{k-1}; \mathbf{c}_{p,j}^k] + b_{g,p}^k\big), \\ \mathbf{z}_{p,j}^{k-1} = m_{p,j}\left(\mathbf{h}_{p,j}^{k-1} + g_{p,j}^k \mathbf{c}_{p,j}^k\right).\end{gathered} \tag{4}$$

Here, σ is the sigmoid function and $[\cdot;\cdot]$ denotes concatenation. The enriched input $\mathbf{z}_{p,j}^{k-1}$ is then propagated by the corresponding Mamba subnetwork, allowing the exchanged cross-stream information to influence subsequent state evolution. Further details are provided in Appendix D.

Mamba subnetwork architecture. Each stream uses a Mamba subnetwork (Gu & Dao, 2024), as shown in Figure 1(d). We disable RMSNorm before Mamba while retaining it after each stream and

final aggregation, and combine the main branch, a right branch with SiLU activation, and the static covariate embedding through residual connections. The causal 1D convolution (Oord et al., 2016) is replaced by dropout on the tumor growth dataset (Geng et al., 2017) to mitigate overfitting, and retained on MIMIC-III (Johnson et al., 2016) to capture local dependencies. Background is provided in Appendix C.

4.2 Multi-Task Collaboration Balancing Module

To learn representations with reduced treatment predictability while discouraging loss of current covariate information, we propose a Multi-Task Collaboration (MTC) balancing module inspired by multi-task learning (Zhang & Yang, 2021), which is illustrated in Figure 1(b). MTC jointly optimizes outcome prediction, adversarial treatment domain confusion, and self-supervised covariate reconstruction.

Primary Task. The primary task is carried out by the outcome prediction head G_Y, implemented as an ELU-based feedforward neural network with parameters θ_Y. It takes the balanced representation $\mathbf{\Phi}_t$ and the current treatment $\mathbf{A}_t$ as input to forecast the next outcome $\hat{\mathbf{Y}}_{t+1}$. The prediction head is trained using the mean squared error (MSE) between predicted and observed factual outcomes, i.e.,

$$\mathcal{L}_{G_Y}(\theta_Y, \theta_R, t) = \|\mathbf{Y}_{t+1} - G_Y(\mathbf{\Phi}_t, \mathbf{A}_t; \theta_Y)\|^2. \tag{5}$$

Adversarial Task. The adversarial task drives the learning of balanced representations through a minimax game. One player in this game is the treatment classifier G_A, which aims to accurately predict the true treatment assignment $\mathbf{A}_t$ from the representation $\mathbf{\Phi}_t$. Similar to G_Y, it is a feedforward network with parameters θ_A. The prediction error is measured by a standard cross-entropy (CE) loss, defined as

$$\mathcal{L}_{G_A}(\theta_A, \theta_R, t) = -\sum_{j=1}^{d_a} \mathbb{I}_{[\mathbf{A}_t = a_j]} \log G_A^j(\mathbf{\Phi}_t; \theta_A), \tag{6}$$

where $\mathbb{I}_{[\cdot]}$ is the indicator function. The other player is the representation module, whose objective is to generate representations that confuse the classifier G_A, preventing it from making accurate predictions. This is achieved by optimizing the representation module Φ to minimize the KL divergence between the classifier output and a uniform distribution over treatments, which corresponds to the counterfactual domain confusion (*cf*) loss (Melnychuk et al., 2022), i.e.,

$$\mathcal{L}_{cf}(\theta_A, \theta_R, t) = -\sum_{j=1}^{d_a} \frac{1}{d_a} \log G_A^j(\mathbf{\Phi}_t; \theta_A). \tag{7}$$

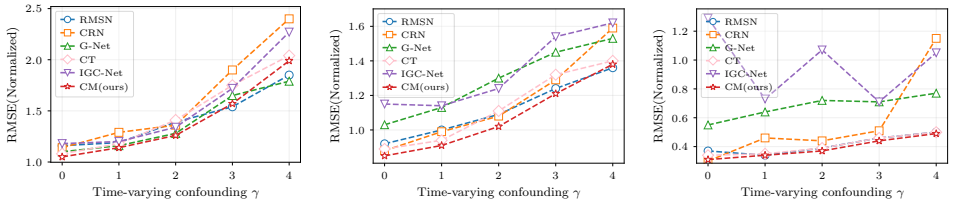
Auxiliary Task. To support the primary prediction task while discouraging the loss of information about current covariates during balancing, we add a self-supervised auxiliary task (Bengio, 2012). The covariate reconstruction head, denoted as G_X, comprises an LSTM decoder L_d and a fully connected layer F_X, i.e., $G_X = F_X \circ L_d$. We refer to the parameters of G_X as θ_X. G_X reconstructs the current time-varying covariates from the representation $\mathbf{\Phi}_t$, providing a stable informational constraint quantified through the MSE loss, given as

$$\mathcal{L}_{G_X}(\theta_X, \theta_R, t) = \|\mathbf{X}_t - G_X(\mathbf{\Phi}_t; \theta_X)\|^2. \tag{8}$$

4.3 Theoretical Analysis of MTC

We analyze the population counterpart $\mathcal{F}$ of Eq. (10) under its pooled training-step law (Appendix E), with $S = \mathbb{E}T \in (0, \infty)$. Let R_P measure squared-error approximation of the conditional outcome mean given the supplied input and treatment, and let R_X denote reconstruction risk. Define $D(\Phi) = \sum_{j=1}^{K} \mathrm{KL}(Q^\Phi \| P_j^\Phi)$, where P_j^Φ is the pooled representation law conditional on $A = a_j$ and Q^Φ its marginal law. At the shared classifier's pooled Bayes response, $\mathcal{F}$ is equivalent, up to scale and a constant, to

$$J^{\mathrm{MTC}} = \mu R_P + D + \lambda R_X, \qquad \mu = K/\alpha_a, \quad \lambda = K\alpha_x/\alpha_a, \quad K = d_a. \tag{9}$$



(a) 1-step-ahead prediction (b) 3-step-ahead prediction (c) 6-step-ahead prediction

Figure 2: Results for 1-step, 3-step, and 6-step-ahead prediction on fully-synthetic dataset from TG simulator. Shown is the mean performance over five runs with different seeds.

Table 1: Results for τ-step-ahead prediction on semi-synthetic MIMIC-III dataset (lower values are better, with the best highlighted in bold). Shown: RMSE as mean $\pm$ standard deviation over five runs.

	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
RMSNs (Lim et al., 2018)	0.15 ± 0.03	0.29 ± 0.04	0.42 ± 0.08	0.53 ± 0.11	0.63 ± 0.15	0.74 ± 0.19
CRN (Bica et al., 2020b)	0.27 ± 0.13	0.35 ± 0.08	0.44 ± 0.05	0.53 ± 0.08	0.61 ± 0.11	0.71 ± 0.15
G-Net (Li et al., 2021)	0.19 ± 0.05	0.35 ± 0.05	0.46 ± 0.07	0.55 ± 0.09	0.62 ± 0.11	0.69 ± 0.13
CT (Melnychuk et al., 2022)	0.16 ± 0.02	0.26 ± 0.04	0.33 ± 0.06	0.41 ± 0.08	0.50 ± 0.10	0.59 ± 0.13
IGC-Net (Hess et al., 2026)	0.16 ± 0.03	0.34 ± 0.15	0.56 ± 0.17	0.64 ± 0.17	0.77 ± 0.10	0.91 ± 0.15
CM (ERM) (ours)	$\mathbf{0.11 \pm 0.02}$	0.23 ± 0.02	0.32 ± 0.03	0.39 ± 0.03	0.46 ± 0.06	0.53 ± 0.09
CM (ours)	$\mathbf{0.11 \pm 0.02}$	$\mathbf{0.21 \pm 0.01}$	$\mathbf{0.28 \pm 0.02}$	$\mathbf{0.37 \pm 0.04}$	$\mathbf{0.44 \pm 0.06}$	$\mathbf{0.51 \pm 0.09}$

Theorem 1 (Joint MTC Risk Bound)**.** *Assume finite pooled second moments of outcomes and covariates, positive treatment priors, and evaluation at the pooled Bayes treatment classifier for every candidate representation. Fix $\alpha_a, \alpha_x > 0$ and $\zeta \geq 0$. Suppose a comparator in the same model classes satisfies $R_P \leq \rho$, $D \leq \delta$, and $R_X \leq \epsilon$, for finite $\rho, \delta, \epsilon \geq 0$. Any ζ-optimal model $\widehat{\theta}$ for $\mathcal{F}$ satisfies, with $B = \mu\rho + \delta + \lambda\epsilon + \mu\zeta/S$,*

$$J^{\mathrm{MTC}}(\widehat{\theta}) \leq B, \qquad R_P(\widehat{\theta}) \leq B/\mu, \qquad R_X(\widehat{\theta}) \leq B/\lambda,$$

and $\mathrm{TV}(P_j^{\widehat{\Phi}}, P_k^{\widehat{\Phi}}) \leq \min\{1, \sqrt{B}\}$ for all j, k.

Proof. See Appendix E.

The comparator quantifies the joint tradeoff among prognostic approximation, pooled treatment balance, and reconstruction. Appendix E.1 connects this guarantee to one-step counterfactual risk under target-policy coverage, accounting for information lost through masked inputs.

For a target treatment policy $q_j(H)$, let H be the complete pre-treatment history and U the supplied input. Write $m_H(H, A) = \mathbb{E}_\star[\mathbf{Y}^+ \mid H, A]$, $m_U(U, A) = \mathbb{E}_\star[\mathbf{Y}^+ \mid U, A]$, and $g_\theta(U, A) = G_Y(\Phi(U), A)$. Define the target risk and input information loss by

$$\mathcal{E}_q(\theta) = \mathbb{E}_{\star,H,U} \sum_{j=1}^{K} q_j(H) \| m_H(H, a_j) - g_\theta(U, a_j) \|_2^2,$$

$$\Delta_{\mathrm{in}} = \mathbb{E}_\star \| m_H(H, A) - m_U(U, A) \|_2^2.$$

Proposition 2 (One-step counterfactual risk)**.** *Under the one-step identification conditions in Appendix E.1, Assumption E.2, and the pooled second-moment condition, let $C_q < \infty$ satisfy $q_j(H) \leq C_q e_j(H)$ almost surely, where $e_j(H) = \mathbb{P}_\star(A = a_j \mid H)$. Every predictor with finite factual risk satisfies*

$$\mathcal{E}_q(\theta) \leq C_q \{ R_P(\theta) + \Delta_{\mathrm{in}} \}.$$

In particular, a model satisfying Theorem 1 obeys

$$\mathcal{E}_q(\widehat{\theta}) \leq C_q \left(\frac{B}{\mu} + \Delta_{\mathrm{in}} \right)$$

$$= C_q \left(\rho + \frac{\alpha_a}{K} \delta + \alpha_x \epsilon + \frac{\zeta}{S} + \Delta_{\mathrm{in}} \right).$$

Proof. See Appendix E.1.

4.4 Training Details

We provide detailed pseudocode in Algorithm 1 to illustrate the training process of CM. CM processes observational data $\mathcal{D}$ as input and returns learned parameters for subsequent inference. The parameters of outcome prediction head G_Y, treatment prediction head G_A, and covariate reconstruction head G_X, i.e., θ_Y, θ_A, and θ_X, are updated by minimizing the loss functions based on Eqs. (5), (6), and (8), respectively. The parameters θ_R in CM for generating the representation $\mathbf{\Phi}_t$ are updated by minimizing the following objective

$$\mathcal{L} = \frac{1}{N} \sum_{i \in \mathcal{D}} \sum_{t=1}^{T^{(i)}} \left(\mathcal{L}_{G_Y}^{(i)}(\theta_Y, \theta_R, t) + \alpha_a \mathcal{L}_{cf}^{(i)}(\theta_A, \theta_R, t) + \alpha_x \mathcal{L}_{G_X}^{(i)}(\theta_X, \theta_R, t) \right), \tag{10}$$

where α_a and α_x denote the weighting coefficients for the adversarial task and auxiliary task, respectively, and control the tradeoff between confounding adjustment and information preservation. The coefficient α_a is initialized to 0.01 following prior work (Tzeng et al., 2015; Melnychuk et al., 2022), while α_x is tuned within a search space (see Table 10 in Appendix F). These coefficients are dynamically scheduled across training epochs, i.e., Adaptive Regularization (Darken & Moody, 1991; Ganin et al., 2016), to balance the learning progress among the primary, adversarial, and auxiliary tasks. The detailed scheduling scheme is provided in Appendix G. Theorem 1 characterizes the time-summed population objective at its pooled Bayes response for a fixed pair of these weights. During training, Exponential Moving Average (EMA) (Tarvainen & Valpola, 2017; Yazıcı et al., 2019) is applied to all trainable parameters θ (i.e., θ_R, θ_Y, θ_A, and θ_X) to smooth parameter updates, thereby enhancing training stability (Athiwaratkun et al., 2018). Formally, we initialize the EMA parameters $\theta_{\text{EMA}}^{(0)}$ and update parameters via

Algorithm 1 Pseudocode for training CM

Input: $\mathcal{D} = \{\{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)}\}_{t=1}^{T^{(i)}}, \mathbf{v}^{(i)}\}_{i=1}^{N}$
Parameters: the index of the optimization step $j = 0$, learning rate η, exponential smoothing parameter β, adversarial task coefficient α_a, auxiliary task coefficient α_x, max epoch n_e
Output: Optimized Parameters $\theta := \{\theta_Y, \theta_A, \theta_X, \theta_R\}$
Initialize EMA parameters $\theta_{\text{EMA}}^{(0)} \leftarrow \theta^{(0)}$
for $e = 1$ **to** n_e **do**
  Update $\alpha_a^{(e)} = \alpha_a \cdot \left(\frac{2}{1+\exp(-10e/n_e)} - 1 \right)$
  Update $\alpha_x^{(e)} = \alpha_x \cdot \left(2 - \frac{2}{1+\exp(-10e/n_e)} \right)$
  for batch $\mathcal{B}$ in epoch **do**
    Compute $\mathcal{L}_{G_Y}^{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T^{(i)}} \mathcal{L}_{G_Y}^{(i)}(\theta_Y, \theta_R, t)$
    Compute $\mathcal{L}_{G_X}^{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T^{(i)}} \mathcal{L}_{G_X}^{(i)}(\theta_X, \theta_R, t)$
    Compute $\mathcal{L}_{cf}^{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T^{(i)}} \mathcal{L}_{cf}^{(i)}(\theta_A, \theta_R, t)$
    $\theta_R \leftarrow \theta_R - \eta(\frac{\partial \mathcal{L}_{G_Y}^{\mathcal{B}}}{\partial \theta_R} + \alpha_a^{(e)} \frac{\partial \mathcal{L}_{cf}^{\mathcal{B}}}{\partial \theta_R} + \alpha_x^{(e)} \frac{\partial \mathcal{L}_{G_X}^{\mathcal{B}}}{\partial \theta_R})$
    $\theta_Y \leftarrow \theta_Y - \eta \frac{\partial \mathcal{L}_{G_Y}^{\mathcal{B}}}{\partial \theta_Y}$
    Compute $\mathcal{L}_{G_A}^{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T^{(i)}} \mathcal{L}_{G_A}^{(i)}(\theta_A, \theta_R, t)$
    $\theta_A \leftarrow \theta_A - \eta \alpha_a^{(e)} \frac{\partial \mathcal{L}_{G_A}^{\mathcal{B}}}{\partial \theta_A}$
    $\theta_X \leftarrow \theta_X - \eta \alpha_x^{(e)} \frac{\partial \mathcal{L}_{G_X}^{\mathcal{B}}}{\partial \theta_X}$
    $j \leftarrow j + 1$
    Update EMA $\theta_{\text{EMA}}^{(j)} \leftarrow \beta \theta_{\text{EMA}}^{(j-1)} + (1-\beta)\theta^{(j)}$
  end for
end for

$$\theta_{\text{EMA}}^{(j)} = \beta \theta_{\text{EMA}}^{(j-1)} + (1-\beta)\theta^{(j)}, \tag{11}$$

where j denotes the iteration number in training and β denotes an exponential smoothing factor, which we set to 0.99 as in the original work (Yazıcı et al., 2019).

In addition to the aforementioned techniques, CM incorporates the mini-batch augmentation with masking (Melnychuk et al., 2022) and Teacher Forcing (Williams & Zipser, 1989) during training. A detailed explanation of these techniques and further implementation details are provided in Appendix G.

CM was implemented using PyTorch Lightning and optimized with Adam (Kingma & Ba, 2015). The trained model was then applied directly for inference.

5 EXPERIMENTS

In this section, we evaluate CM following standard counterfactual inference benchmarking protocols (Bica et al., 2020b; Melnychuk et al., 2022). We benchmark CM against baselines on well-established synthetic and real-world datasets for counterfactual estimation over time, and further compare runtime across methods to assess computational efficiency. Finally, we conduct component-wise ablation studies to quantify the contributions of individual components.

Baselines. We select representative baselines for counterfactual estimation over time, including **RMSNs** (Lim et al., 2018), **CRN** (Bica et al., 2020b), **G-Net** (Li et al., 2021), **CT** (Melnychuk et al., 2022) and **IGC-Net** (Hess et al., 2026). Descriptions of these methods are provided in Appendix H. Hyperparameter tuning is applied to all baselines for a fair comparison (see Appendix F for details).

Benchmarks. Consistent with standard evaluation practices (Bica et al., 2020b; Melnychuk et al., 2022; Hess et al., 2026) for counterfactual estimation over time, we employ synthetic, semi-synthetic, and real-world datasets in our experiments. These include: **(1) The fully-synthetic tumor growth dataset.** The synthetic dataset is generated via a pharmacokinetic-pharmacodynamic (PK-PD) model (Geng et al., 2017) that simulates tumor growth (TG simulator) in lung cancer patients under combined chemotherapy and radiotherapy. The time-varying confounding level γ controls the strength of time-dependent confounding in the treatment assignment process. **(2) The semi-synthetic MIMIC-III dataset.** Following (Melnychuk et al., 2022), the semi-synthetic dataset is built from the Medical Information Mart for Intensive Care III database (MIMIC-III) (Johnson et al., 2016). Its high-dimensional patient trajectories contain controlled confounding, known counterfactual ground truth, and endogenous/exogenous dependencies (Schulam & Saria, 2017). **(3) The real-world MIMIC-III dataset.** This real-world observational dataset consists of patient stays from the intensive care unit (ICU). Following prior studies (Kuzmanovic et al., 2021; Melnychuk et al., 2022; Hatt & Feuerriegel, 2024), we use 25 vital signs and 3 static features, with diastolic blood pressure as the outcome and vasopressors and mechanical ventilation as treatments. This setup reflects a clinical scenario with a high degree of confounding where treatment decisions are closely tied to the evolving physiological state of patients, enabling evaluation of model performance under realistic conditions. As counterfactual outcomes are unobserved, this benchmark evaluates factual outcome forecasting only. Further datasets details are provided in Appendix I.

Table 2: Results for τ-step-ahead factual outcome forecasting on the real-world MIMIC-III dataset. Shown: RMSE as mean $\pm$ standard deviation over five runs.

	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$
RMSNs	5.06 ± 0.18	9.74 ± 0.10	10.94 ± 0.87	12.09 ± 1.78	13.33 ± 2.68
CRN	4.95 ± 0.70	8.98 ± 0.24	9.62 ± 0.25	9.98 ± 0.25	10.26 ± 0.25
G-Net	4.93 ± 0.08	11.53 ± 0.35	12.70 ± 0.63	13.41 ± 0.64	14.20 ± 0.42
CT	4.49 ± 0.12	8.94 ± 0.25	9.54 ± 0.23	$\mathbf{9.85 \pm 0.24}$	$\mathbf{10.10 \pm 0.24}$
IGC-Net	4.52 ± 0.06	8.94 ± 0.18	9.59 ± 0.19	9.91 ± 0.26	10.36 ± 0.42
CM (ours)	$\mathbf{4.47 \pm 0.12}$	$\mathbf{8.89 \pm 0.24}$	$\mathbf{9.53 \pm 0.22}$	9.89 ± 0.22	10.18 ± 0.22

5.1 PERFORMANCE COMPARISON

Fully-synthetic TG simulator dataset. Figure 2 reports the normalized RMSE (nRMSE) under time-varying confounding level γ. CM outperforms baselines across most combinations of projection horizons τ and γ. These results suggest that CM maintains favorable predictive performance as time-varying confounding becomes stronger. Complete results are provided in Appendix J, along with t-SNE visualizations of the learned balanced representations in Appendix K.

Semi-synthetic MIMIC-III dataset. Table 1 summarizes the results. CM achieves consistent and substantial improvements over all baselines across projection horizons τ. To isolate the contribution of the MTC balancing module, we compare CM against its empirical risk minimization variant, CM (ERM), which excludes the MTC module ($\alpha_a = \alpha_x = 0$). The performance gain of CM over CM (ERM) is consistent with the effectiveness of the proposed MTC balancing module design.

Real-world MIMIC-III dataset. Since counterfactual outcomes are unobserved in the real-world dataset, we evaluate τ-step-ahead factual outcome forecasting (see Table 2). CM outperforms all baselines for $\tau = 1, 2, 3$ while consistently surpassing the remaining methods. These results demonstrate competitive factual forecasting performance on real-world MIMIC-III.

Table 3: Runtime of experiments on the fully-synthetic dataset from TG simulator. Experiments are carried out on $1\times$ NVIDIA A10 Tensor Core GPU, where ITPS refers to inference time per sample. Unless otherwise specified, the unit is minutes.

	Main stages of Training & Inference	Training Time	Inference Time	Total Runtime	ITPS (in sec)
RMSNs	2 networks for IPTW & encoder & decoder	9.7 ± 0.1	$\mathbf{0.2 \pm 0.0}$	9.9 ± 0.1	**0.01**
CRN	Encoder & Decoder	5.7 ± 0.0	0.4 ± 0.0	6.1 ± 0.0	0.02
G-Net	Single network & MC sampling for inference	1.4 ± 0.0	15.6 ± 0.3	17.0 ± 0.4	0.94
CT	Single Transformer multi-input block	5.7 ± 0.0	0.9 ± 0.0	6.6 ± 0.1	0.05
IGC-Net	Regression-based Transformer multi-input block	23.9 ± 0.2	0.4 ± 0.0	24.3 ± 0.2	0.02
CM	GCSI-Mamba block	$\mathbf{1.1 \pm 0.0}$	$\mathbf{0.2 \pm 0.0}$	$\mathbf{1.3 \pm 0.0}$	**0.01**

5.2 Efficiency Analysis

Since accuracy and efficiency are both important in practice, we compare the runtime of CM with the baselines. As shown in Table 3, CM achieves the lowest total runtime on the fully-synthetic dataset, with approximately $5.1\times$ speedup over CT, and matches RMSN for the lowest reported inference time per sample (ITPS) at 0.01 s. This efficiency stems from the GCSI-Mamba block and integrated end-to-end training, whereas other methods incur overhead from multi-stage training, Monte Carlo sampling, or the quadratic complexity of Transformers. The inference advantage over CT also holds for longer sequences. Additional runtime results for the semi-synthetic and real-world datasets, trainable parameter counts, and detailed long-horizon comparisons are provided in Appendix L and Appendix M, respectively.

5.3 Ablation Study

We assess the impact of core components in CM via ablation studies on the semi-synthetic dataset, analyzing the contributions of the GCSI-Mamba block, MTC module, and subnetwork architecture. Ablation experiments cover three settings: **(1) GCSI Module**. We remove GCSI to assess the contribution of cross-stream interaction before the selective scans; **(2) Subnetwork Architecture**. We remove the causal convolution, disable dropout, restore RMSNorm before Mamba, and remove the residual connection; **(3) MTC Module**. We evaluate variants without EMA of model weights ($\beta = 0$), variants with EMA but without the MTC module ($\alpha_a = \alpha_x = 0$) and swapping the MTC module for CDC loss from CT while keeping the GCSI-Mamba block, with both using their respective optimal hyperparameters.

Table 4: Ablation study results with average RMSE for CM on the semi-synthetic MIMIC-III dataset.

		$\tau = 1$	$\tau = 3$	$\tau = 5$	$\tau = 6$
CM		0.11	0.28	0.44	0.51
GCSI Module	w/o GCSI	$+0.01$	$+0.02$	$+0.02$	$+0.03$
Subnetwork Architecture	w/o convolution layer	$+0.01$	$+0.09$	$+0.12$	$+0.13$
	w/o dropout	$+0.01$	$+0.07$	$+0.08$	$+0.08$
	w/ RMSNorm	± 0.00	$+0.04$	$+0.04$	$+0.04$
	w/o residual	± 0.00	$+0.10$	$+0.25$	$+0.36$
MTC Module	w/o EMA	$+0.01$	$+0.04$	$+0.02$	$+0.01$
	w/o balancing	± 0.00	$+0.04$	$+0.02$	$+0.02$
	w/ CDC loss	± 0.00	$+0.01$	$+0.02$	$+0.03$

The results of the ablation study are shown in Table 4. Removing GCSI increases RMSE by 0.01, 0.02, 0.02, and 0.03 at prediction horizons $\tau = 1, 3, 5, 6$, respectively. Removing balancing or replacing MTC with CDC also increases RMSE at $\tau = 3, 5, 6$. These results support the contribution of the GCSI-Mamba block and its subnetwork design, especially at longer prediction horizons, and confirm the effectiveness of the MTC module.

6 Conclusion

We introduce Causal Mamba (CM), an end-to-end framework for estimating counterfactual outcomes over time. Its GCSI-Mamba backbone combines stream-specific selective state space dynamics with masked blockwise interaction among treatment, covariate, and outcome states, enabling cross-stream information to propagate efficiently over time, while its MTC balancing module coordinates prediction, adversarial, and reconstruction objectives. Extensive experiments demonstrate that CM not only outperforms existing baselines but also achieves higher runtime efficiency. Extension to continuous treatment regimes is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools for brainstorming, summarizing or analyzing existing literature, and improving the clarity of explanations in some sections of the paper. We have not used generative AI tools for generating novel research contributions, datasets, or experiments. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops and evaluates methods for estimating counterfactual outcomes from longitudinal observational data. Experiments use synthetic data and the de-identified MIMIC-III dataset (Johnson et al., 2016); they do not involve recruiting participants, intervening on patients, or deploying the model in clinical practice. De-identification protects patient privacy.

Because the proposed method may inform healthcare decisions, potential downstream use requires careful assessment of bias, fairness, uncertainty, and safety. Results on observational benchmarks alone do not establish clinical effectiveness or justify treatment decisions. Any clinical application would require further validation and appropriate professional oversight.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental information needed to support reproduction of this work:

- **Theoretical results.** All theorems and lemmas proposed in this paper are presented with complete proofs (see Appendix E). Proofs are written to be self-contained, without reliance on unpublished materials.
- **Empirical evaluation.** Dataset construction and evaluation settings are described in Appendix I, baseline implementations in Appendix H, and hyperparameter search ranges in Appendix F. Section 4.4 and Appendix G document the training procedure and additional implementation details.
- **Code Resources.** We provide the complete source code for reproducing the experiments through an anonymous repository: `https://anonymous.4open.science/r/causal-mamba/`.

REFERENCES

Jong-Hoon Ahn and Akshay Vashist. Gaussian mixture counterfactual generator. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ahmed Alaa and Mihaela van der Schaar. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In *International Conference on Machine Learning*, pp. 129–138. PMLR, 2018.

Ahmed Allam, Stefan Feuerriegel, Michael Rebhan, and Michael Krauthammer. Analyzing patient trajectories with artificial intelligence. *Journal of medical internet research*, 23(12):e29812, 2021.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. *arXiv preprint arXiv:1806.05594*, 2018.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Christoph Bergmeir and José M Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The journal of machine learning research*, 13(1):281–305, 2012.

Ioana Bica, Ahmed Alaa, and Mihaela Van Der Schaar. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *International conference on machine learning*, pp. 884–895. PMLR, 2020a.

Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*, 2020b.

Defu Cao, James Enouen, Yujing Wang, Xiangchen Song, Chuizheng Meng, Hao Niu, and Yan Liu. Estimating treatment effects from irregular time series observations with hidden confounders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6897–6905, 2023.

Alicia Curth and Mihaela Van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1810–1818. PMLR, 2021.

Christian Darken and John Moody. Towards faster stochastic gradient search. *Advances in neural information processing systems*, 4, 1991.

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*, 30(4):958–968, 2024.

Dennis Frauen, Tobias Hatt, Valentyn Melnychuk, and Stefan Feuerriegel. Estimating average causal effects from patient trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7586–7594, 2023a.

Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. Sharp bounds for generalized causal sensitivity analysis. *Advances in Neural Information Processing Systems*, 36:40556–40586, 2023b.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

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 biomathematical model. *Scientific reports*, 7(1):13542, 2017.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

Tobias Hatt and Stefan Feuerriegel. Sequential deconfounding for causal inference with unobserved confounders. In *Causal Learning and Reasoning*, pp. 934–956. PMLR, 2024.

Konstantin Hess, Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Bayesian neural controlled differential equations for treatment effect estimation. In *The Twelfth International Conference on Learning Representations*, 2024.

Konstantin Hess, Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. IGC-Net for conditional average potential outcome estimation over time. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xiaodi Hou, Xiaobo Li, Mingyu Lu, Simiao Wang, and Yijia Zhang. Rrg-mamba: Efficient radiology report generation with state space model. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 7410–7418, 2025.

Qiang Huang, Chuizheng Meng, Defu Cao, Biwei Huang, Yi Chang, and Yan Liu. An empirical examination of balancing strategy for counterfactual estimation on time series. In *International Conference on Machine Learning*, pp. 20043–20062. PMLR, 2024.

Kosuke Imai, In Song Kim, and Erik H Wang. Matching methods for causal inference with time-series cross-sectional data. *American Journal of Political Science*, 67(3):587–605, 2023.

Fredrik D Johansson, Uri Shalit, Nathan Kallus, and David Sontag. Generalization bounds and representation learning for estimation of potential outcomes and causal effects. *Journal of Machine Learning Research*, 23(166):1–50, 2022.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):160035, 2016.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Insung Kong, Yuha Park, Joonhyuk Jung, Kwonsang Lee, and Yongdai Kim. Covariate balancing using the integral probability metric for causal inference. In *International Conference on Machine Learning*, pp. 17430–17461. PMLR, 2023.

Kun Kuang, Peng Cui, Hao Zou, Bo Li, Jianrong Tao, Fei Wu, and Shiqiang Yang. Data-driven variable decomposition for treatment effect estimation. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2120–2134, 2020.

Milan Kuzmanovic, Tobias Hatt, and Stefan Feuerriegel. Deconfounding temporal autoencoder: estimating treatment effects over time using noisy proxies. In *Machine learning for health*, pp. 143–155. PMLR, 2021.

Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, et al. G-net: a recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime. In *Machine Learning for Health*, pp. 282–299. PMLR, 2021.

Xinshu Li and Lina Yao. Contrastive individual treatment effects estimation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 1053–1058. IEEE, 2022.

Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Mingxuan Liu, Xinru Wang, Jin Wee Lee, Bibhas Chakraborty, Nan Liu, and Victor Volovici. Causal inference from observational data in neurosurgical studies: a mini-review and tutorial. *Acta Neurochirurgica*, 167(1):40, 2025.

Wen Wei Loh and Dongning Ren. Estimating time-varying treatment effects in longitudinal studies. *Psychological Methods*, 30(2):240, 2025.

Yu Luo, Kuan Liu, Ramandeep Singh, and Daniel J Graham. A longitudinal bayesian framework for estimating causal dose-response relationships. *arXiv preprint arXiv:2505.20893*, 2025.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Robert K Mahar, Myra B McGuinness, Bibhas Chakraborty, John B Carlin, Maarten J IJzerman, and Julie A Simpson. A scoping review of studies using observational data to optimise dynamic treatment regimens. *BMC medical research methodology*, 21(1):39, 2021.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15293–15329. PMLR, 2022.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Normalizing flows for interventional density estimation. In *International Conference on Machine Learning*, pp. 24361–24397. PMLR, 2023.

Chuizheng Meng, Yihe Dong, Sercan Ö Arık, Yan Liu, and Tomas Pfister. Costar: Improved temporal counterfactual estimation with self-supervised learning. *arXiv preprint arXiv:2311.00886*, 2023.

Lizhen Nie, Mao Ye, Qiang Liu, and Dan Nicolae. Vcnet and functional targeted regularization for learning causal effects of continuous treatments. *arXiv preprint arXiv:2103.07861*, 2021.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Badri Narayana Patro and Vijay Srinivas Agneeswaran. Simba-ts: Simplified channel mixing and mamba for long-term time series forecasting. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

James Robins. 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.

James Robins and Miguel Hernan. Estimation of the causal effects of time-varying exposures. *Chapman & Hall/CRC Handbooks of Modern Statistical Methods*, pp. 553–599, 2008.

James M Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics-Theory and methods*, 23(8):2379–2412, 1994.

James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560, 2000.

Mercy Rop, Innocent Maposa, Taryn Young, and Rhoderick Machekano. Causal estimation of time-varying treatments in observational studies: a scoping review of methods, applications, and missing data practices. *BMC Medical Research Methodology*, 25(1):202, 2025.

Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pp. 34–58, 1978.

Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. *Advances in neural information processing systems*, 30, 2017.

Nabeel Seedat, Fergus Imrie, Alexis Bellot, Zhaozhi Qian, and Mihaela van der Schaar. Continuous-time modeling of counterfactual outcomes using neural controlled differential equations. In *International Conference on Machine Learning*, pp. 19497–19521. PMLR, 2022.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pp. 3076–3085. PMLR, 2017.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. *arXiv preprint arXiv:1704.02038*, 2017.

Souhaib Ben Taieb and Rob Hyndman. Boosting multi-step autoregressive forecasts. In *International conference on machine learning*, pp. 109–117. PMLR, 2014.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Aleksandar Terzic, Michael Hersche, Giacomo Camposampiero, Thomas Hofmann, Abu Sebastian, and Abbas Rahimi. On the expressiveness and length generalization of selective state space models on regular languages. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20876–20884, 2025.

Gino Tesei, Stefanos Giampanis, Jingpu Shi, and Beau Norgeot. Learning end-to-end patient representations through self-supervised covariate balancing for causal treatment effect estimation. *Journal of biomedical informatics*, 140:104339, 2023.

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4068–4076, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shirly Wang, Matthew BA 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*, pp. 222–235, 2020.

Xin Wang, Shengfei Lyu, Xingyu Wu, Tianhao Wu, and Huanhuan Chen. Generalization bounds for estimating causal effects of continuous treatments. *Advances in Neural Information Processing Systems*, 35:8605–8617, 2022.

Xin Wang, Shengfei Lyu, Lishan Yang, Yibing Zhan, and Huanhuan Chen. A dual-module framework for counterfactual estimation over time. In *International Conference on Machine Learning*, pp. 51063–51085. PMLR, 2024.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Anpeng Wu, Kun Kuang, Junkun Yuan, Bo Li, Runze Wu, Qiang Zhu, Yueting Zhuang, and Fei Wu. Learning decomposed representation for counterfactual inference. *arXiv preprint arXiv:2006.07040*, 2020.

Yuhan Wu, Xiyu Meng, Huajin Hu, Junru Zhang, Yabo Dong, and Dongming Lu. Affirm: Interactive mamba with adaptive fourier filters for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21599–21607, 2025.

Timing Yang, Feng Wang, and Guoyizhe Wei. RNN as linear transformer: A closer investigation into representational potentials of visual Mamba models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27398–27408, June 2026.

Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in GAN training. In *International Conference on Learning Representations*, 2019.

Hangting Ye, Zhining Liu, Wei Cao, Amir M Amiri, Jiang Bian, Yi Chang, Jon D Lurie, Jim Weinstein, and Tie-Yan Liu. Web-based long-term spine treatment outcome forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3082–3092, 2023.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34:5586–5609, 2021.

Kailiang Zhong, Fengtong Xiao, Yan Ren, Yaorong Liang, Wenqing Yao, Xiaofeng Yang, and Ling Cen. Descn: Deep entire space cross networks for individual treatment effect estimation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4612–4620, 2022.

Rouyi Zhou, Yangzhi Wu, Jiajun Wen, Can Gao, Feng Liu, Zhihui Lai, and Linlin Shen. Gamba: Mamba-based graph convolutional network with dynamic graph topology learning for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6065–6074, June 2026.

A Extended Related Work

Longitudinal causal models address hidden confounding, continuous-time dynamics, complex treatments, population effects, and distribution shifts. For hidden confounding, the Time Series Deconfounder of Bica et al. (2020a) uses a recurrent factor model of multiple treatment assignments to infer substitutes for multi-cause unobserved confounders, which are then incorporated into treatment effect estimation. LipCDE (Cao et al., 2023) addresses irregular observations and hidden confounding by combining neural controlled differential equations with Lipschitz-constrained networks to bound the effects of hidden confounders. The Sequential Deconfounder (Hatt & Feuerriegel, 2024) instead uses a Gaussian process latent variable model to infer substitute confounders before fitting an outcome model, supporting settings with a single treatment assigned at each time step. For continuous-time estimation, TE-CDE (Seedat et al., 2022) models latent trajectories with neural controlled differential equations, allowing potential outcomes to be evaluated at arbitrary times from irregular observations; adversarial training adjusts for time-dependent confounding. BNCDE (Hess et al., 2024) adds uncertainty quantification through coupled neural controlled and stochastic differential equations, using variational Bayesian inference to obtain posterior predictive distributions of potential outcomes under a treatment sequence. For continuous, multidimensional, and time-dependent treatments, GMCG (Ahn & Vashist, 2025) uses a Gaussian mixture generative model of covariates, treatments, and outcomes with a latent state, whose conditional distribution is used to adjust for confounding when generating counterfactuals. At the population level, DeepACE (Frauen et al., 2023a) combines iterative G-computation with an end-to-end neural model and a sequential targeting procedure to estimate average causal effects from patient trajectories, with double robustness and asymptotic efficiency under the assumptions of the method. Finally, COSTAR (Meng et al., 2023) learns historical representations using a component-wise contrastive objective for temporal treatment and outcome observations. Its self-supervised Transformer formulation is motivated by unsupervised domain adaptation and empirically improves counterfactual prediction and generalization to out-of-distribution data.

State Space Models (SSM) and Mamba. State Space models have achieved significant progress in sequence modeling, describing system evolution through internal states (Gu & Dao, 2024). The S4 model (Gu et al., 2021a) improved modeling efficiency based on HiPPO theory, and subsequent enhancements added parallel scanning (Smith et al., 2022) and gating mechanisms (Mehta et al., 2022). Mamba (Gu & Dao, 2024) further introduced a selective state-space mechanism that employs input-dependent weighting, substantially enhancing runtime efficiency and modeling capability. It achieves substantially lower computational complexity than Transformers while maintaining comparable performance, as demonstrated across natural language processing (Terzic et al., 2025; Hou et al., 2025), computer vision (Yang et al., 2026; Zhou et al., 2026), and time series forecasting (Wu et al., 2025; Patro & Agneeswaran, 2025). Given its efficiency and performance advantages, this work employs the Mamba architecture for estimating counterfactual outcomes over time.

B Problem Assumptions for Causal Identification

Our model is anchored in the potential outcomes framework, originally formalized by Rubin (1978) and further extended to longitudinal settings with time-varying treatments and outcomes as developed by Robins & Hernan (2008). To address the bias introduced by confounding factors in observational data, we make the following identifiability assumptions (Robins et al., 2000; Robins & Hernan, 2008), consistent with prior work (Lim et al., 2018; Bica et al., 2020b; Li et al., 2021; Melnychuk et al., 2022; Hess et al., 2026):

Assumption B.1. (**Consistency**). For a given treatment sequence $\bar{\mathbf{A}}_t = \bar{\mathbf{a}}_t$, an individual potential outcome under that specific sequence $\bar{\mathbf{a}}_t$ is consistent with the observed (factual) outcome, i.e., $\mathbf{Y}_{t+1}[\bar{\mathbf{a}}_t] = \mathbf{Y}_{t+1}$.

Assumption B.2. (**Sequential Overlap**). At each time point t, and for all individuals regardless of their history, the probability of receiving versus not receiving treatment must be nonzero, i.e., $0 < P(\mathbf{A}_t = \mathbf{a}_t | \bar{\mathbf{H}}_t = \bar{\mathbf{h}}_t) < 1, \forall \mathbf{a}_t \in \mathcal{A}$ if $P(\bar{\mathbf{H}}_t = \bar{\mathbf{h}}_t) > 0$, where $\bar{\mathbf{h}}_t$ is a realization of $\bar{\mathbf{H}}_t$.

Assumption B.3. (**Sequential Ignorability**). The treatment received at each time point t is independent of the potential outcomes, conditional on the observed history up to that time, i.e., $\mathbf{A}_t \perp\!\!\!\perp \mathbf{Y}_{t+1}[\mathbf{a}_t] \mid \bar{\mathbf{H}}_t, \forall \mathbf{a}_t \in \mathcal{A}$. This implies that there are no unobserved confounders that affect both treatment and outcome.

The data generating mechanism for $\mathcal{D}$ is shown in Figure 3. The figure captures the core complexities of counterfactual estimation over time: time-varying confounders that jointly influence future treatments and outcomes, treatment effects contingent on dynamic patient history, and the inherent sequential dependencies in longitudinal data.

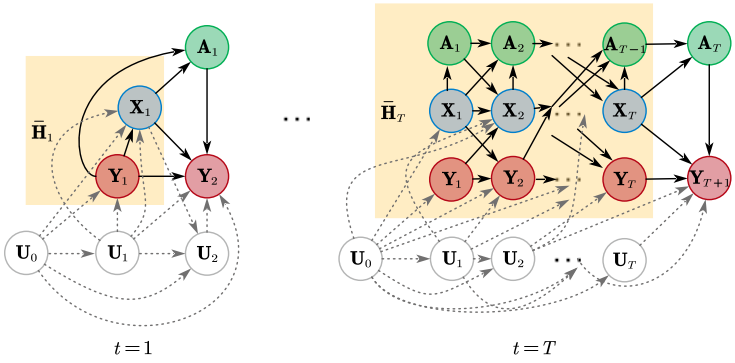


Figure 3: Causal graph depicting the time-varying counterfactual data generation mechanism across time steps t. Here, $\mathbf{U}_t$ denotes unobserved noise that affects only the time-varying covariates and outcomes, but not the treatment assignment. The history $\bar{\mathbf{H}}_t$ conditionally includes all observed time-varying confounders up to time t. Static covariates are omitted for simplicity.

C STATE SPACE MODELS (SSM) AND MAMBA

The State Space Model (SSM) (Gu et al., 2021a;b; Gu & Dao, 2024) is derived from the continuous signal mapping framework. It models sequence dynamics through a continuous-time linear state-space system. The core objective of SSM is to estimate the mapping from an input signal to an output signal through the transition of hidden states. Its continuous-time formulation can be expressed as follows

$$h'(t) = \mathbf{E}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h(t), \tag{12}$$

where $\mathbf{E} \in \mathbb{R}^{N\times N}$ represents the state transition matrix, while $\mathbf{B} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times N}$ denote the input and output matrices, respectively.

The S4 model (Gu et al., 2021a), an improvement upon the SSM framework, discretizes the parameters $\mathbf{E}$ and $\mathbf{B}$ into $\bar{\mathbf{E}}$ and $\bar{\mathbf{B}}$ to yield a discrete-time formulation of the equation given in Equation (12). Its operating principle is summarized as follows

$$h_t = \bar{\mathbf{E}}h_{t-1} + \bar{\mathbf{B}}x_t, \quad y_t = \mathbf{C}h_t, \tag{13}$$

where $\bar{\mathbf{E}} = \exp(\Delta\mathbf{E}), \bar{\mathbf{B}} = (\Delta\mathbf{E})^{-1}(\exp(\Delta\mathbf{E}) - \mathbf{I}) \cdot \Delta\mathbf{B}$, Δ represents the time step. However, S4 uses input-independent state space parameters, so every token is processed by the same linear dynamics. Mamba (Gu & Dao, 2024) introduces selectivity by making the discretization step and the input and output projections functions of the current input. In the notation above, its selective parameters can be written as

$$\mathbf{B}_t = s_B(x_t), \qquad \mathbf{C}_t = s_C(x_t), \qquad \Delta_t = \text{softplus}(\Delta_0 + s_\Delta(x_t)), \tag{14}$$

where s_B, s_C, and s_Δ are learned input projections. The continuous-time transition matrix $\mathbf{E}$ remains a learned, input-independent parameter. Its discretized counterpart is input-dependent through Δ_t rather than being projected directly from x_t:

$$\bar{\mathbf{E}}_t = \exp(\Delta_t\mathbf{E}), \qquad \bar{\mathbf{B}}_t = (\Delta_t\mathbf{E})^{-1}(\exp(\Delta_t\mathbf{E}) - \mathbf{I})\,\Delta_t\mathbf{B}_t, \qquad y_t = \mathbf{C}_t h_t. \tag{15}$$

Thus, the input modulates state retention through Δ_t, information written to the state through $\mathbf{B}_t$, and information read from the state through $\mathbf{C}_t$.

Consequently, Mamba achieves a powerful synergy: it retains the runtime efficiency and long-range dependency modeling capabilities inherent to S4, while gaining the expressiveness and context sensitivity characteristic of attention-based models (Vaswani et al., 2017). This makes it particularly suitable for modeling long sequences where both efficiency and content-aware reasoning are critical, establishing it as a foundational block in modern sequence modeling architectures like the GCSI-Mamba block used in our framework.

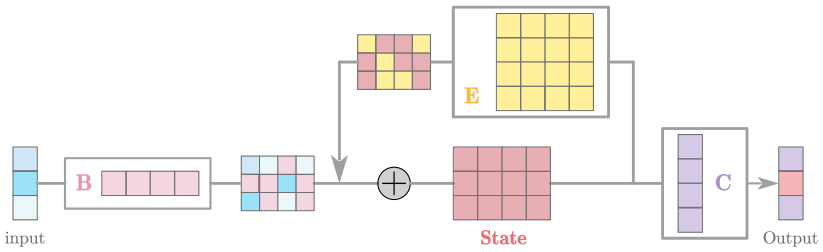


Figure 4: Schematic illustration of the State Space Model.

D FURTHER DETAILS OF THE GCSI-MAMBA BACKBONE

The core interaction equations are presented in Section 4.1. This appendix supplies the surrounding architecture, implementation details, temporal information constraints, and computational complexity analysis.

D.1 INPUT EMBEDDINGS AND AVAILABILITY MASKS

Let $\mathcal{P} \subseteq \{A, X, Y\}$ denote the streams present in a dataset, and let d_h be their common hidden width. We use $\mathcal{P} = \{A, X, Y\}$ when time-varying covariates are available and the two-stream specialization $\mathcal{P} = \{A, Y\}$ for the fully-synthetic tumor data, which have no covariate stream. At representation index $j \le t$, the aligned observations are $(\mathbf{A}_{j-1}, \mathbf{X}_j, \mathbf{Y}_j)$; during projection ($j > t$), only planned treatment $\mathbf{a}_{j-1}$ and autoregressive outcome $\widehat{\mathbf{Y}}_j$ are available. We use a zero start vector for $\mathbf{A}_0$. The X-specific line is omitted when $X \notin \mathcal{P}$; otherwise, the initial embeddings are

$$\begin{aligned}
\mathbf{h}^0_{A,j} &= \begin{cases} \mathrm{Linear}_A(\mathbf{A}_{j-1}), & j \le t, \\ \mathrm{Linear}_A(\mathbf{a}_{j-1}), & j > t, \end{cases} \\
\mathbf{h}^0_{X,j} &= \begin{cases} \mathrm{Linear}_X(\mathbf{X}_j), & j \le t, \\ \mathbf{0}, & j > t, \end{cases} \\
\mathbf{h}^0_{Y,j} &= \begin{cases} \mathrm{Linear}_Y(\mathbf{Y}_j), & j \le t, \\ \mathrm{Linear}_Y(\widehat{\mathbf{Y}}_j), & j > t, \end{cases} \qquad \widetilde{\mathbf{V}} = \mathrm{Linear}_V(\mathbf{V}).
\end{aligned} \tag{16}$$

Let $m_{p,j} \in \{0, 1\}$ indicate that stream p is available at an active sequence position. Thus, with the m_X line applying only when $X \in \mathcal{P}$,

$$\begin{aligned}
m_{A,j} &= m_{Y,j} = m_{\mathrm{active},j}, \\
m_{X,j} &= \begin{cases} m_{\mathrm{active},j}, & j \le t, \\ 0, & j > t. \end{cases}
\end{aligned} \tag{17}$$

For $k = 1, \ldots, K$, block F^k_Φ maps the incoming states $\{\mathbf{h}^{k-1}_{p,1:L}\}_{p\in\mathcal{P}}$ to $\{\mathbf{h}^k_{p,1:L}\}_{p\in\mathcal{P}}$; equivalently, $\mathbf{h}^k_{1:L} = F^k_\Phi(\mathbf{h}^{k-1}_{1:L})$.

D.2 NUMERICAL IMPLEMENTATION AND INITIALIZATION

The query, key, value, and score definitions follow Eq. (2); the gated message update follows Eq. (4). For numerical stability, we evaluate the masked normalization in Eq. (3) using the equivalent max-subtracted form

$$\alpha^k_{p\leftarrow q,j} = \begin{cases} \dfrac{\exp(s^k_{p\leftarrow q,j} - c^k_{p,j})}{\sum_{u\in\mathcal{S}_{p,j}} \exp(s^k_{p\leftarrow u,j} - c^k_{p,j})}, & q \in \mathcal{S}_{p,j}, \\ 0, & q \notin \mathcal{S}_{p,j}, \end{cases} \tag{18}$$

where $c^k_{p,j} = \max_{u\in\mathcal{S}_{p,j}} s^k_{p\leftarrow u,j}$ for a nonempty set of valid sources and is set to zero otherwise. Thus each nonempty row sums to one, whereas an empty row and its message are exactly zero. In the implementation, gcsi_eps only sets a lower bound on the floating point denominator of a nonempty row; it is not added to the normalizing sum. We initialize $b^k_{g,p} = -2$.

Each enriched stream is subsequently processed by its Mamba subnetwork, followed by the existing residual from static covariates, normalization, and output projection:

$$\begin{aligned}
\mathbf{u}^k_{p,1:L} &= \mathrm{Mamba}^k_p(\mathbf{z}^{k-1}_{p,1:L}), \\
\mathbf{h}^k_{p,j} &= m_{p,j}\mathbf{P}^k_p\,\mathrm{RMSNorm}\Big(\mathbf{u}^k_{p,j} + \widetilde{\mathbf{V}}\Big).
\end{aligned} \tag{19}$$

GCSI operates across stream types at the same index rather than directly across temporal indices. Within the selective scan, suppressing unchanged gating terms, the state update has the form

$$\boldsymbol{\xi}^k_{p,j} = \overline{\mathbf{A}}^k_{p,j}\boldsymbol{\xi}^k_{p,j-1} + \overline{\mathbf{B}}^k_{p,j}\mathbf{z}^{k-1}_{p,j}. \tag{20}$$

Consequently, cross-stream information incorporated at representation index j can influence later states through

$$(\overline{\mathbf{A}}^{k}_{p,j+s} \cdots \overline{\mathbf{A}}^{k}_{p,j+1})\overline{\mathbf{B}}^{k}_{p,j}\mathbf{z}^{k-1}_{p,j}. \tag{21}$$

Because GCSI precedes temporal propagation, treatment, covariate, and outcome information can influence the subsequent state evolution of each stream.

After block K, we average only the available stream states and project their mean:

$$\begin{aligned} \mathbf{z}^{K}_{j} &= \frac{\sum_{p\in\mathcal{P}} m_{p,j}\mathbf{h}^{K}_{p,j}}{\max\{1, \sum_{p\in\mathcal{P}} m_{p,j}\}}, \\ \mathbf{\Phi}_{j} &= \mathrm{ELU}\big(\mathbf{W}_{R}\,\mathrm{RMSNorm}(\mathbf{z}^{K}_{j}) + \mathbf{b}_{R}\big)\,. \end{aligned} \tag{22}$$

Thus observed positions average three streams when $X \in \mathcal{P}$ and two streams otherwise, whereas only the planned treatment and autoregressive outcome streams are averaged at projection positions. By induction over time and blocks,

$$\mathbf{h}^{k}_{p,j} \in \sigma\big(\mathbf{V}, \mathbf{A}_{1:j-1}, \mathbf{X}_{1:\min(j,t)}, \mathbf{Y}_{1:j}\big)\,, \tag{23}$$

where the X term is omitted when $X \notin \mathcal{P}$, and planned treatments and autoregressive outcomes replace unavailable future factual variables. Hence neither $\mathbf{A}_j$ nor future factual outcomes or covariates enter $\mathbf{\Phi}_j$.

D.3 Computational complexity

For batch size B, length L, $S \leq 3$ streams, and rank r, one GCSI module costs $O(BL(Sd_h r + S^2 r))$ time and $O(BL(Sd_h + S^2))$ activations. Since S is constant, scaling with respect to sequence length remains linear. Shared query, key, value, and output projections, stream embeddings, and scalar gates add approximately $4d_h r + 3Sd_h + S$ parameters per block. All parameters for embedding, GCSI, stream specific Mamba subnetworks, static residual connections, normalization, fusion, and representation projection belong to θ_R.

E THEORETICAL ANALYSIS OF MULTI-TASK COLLABORATION

We analyze the three-task population objective at fixed positive weights α_a, α_x. The representation, outcome head, and reconstruction decoder share parameters across time as in the model. The result compares the joint prediction, balancing, and reconstruction risks attainable within these model classes.

Population law and temporal weighting. Let T denote the number of valid loss terms for a sampled trajectory and assume $0 < S = \mathbb{E}T < \infty$. For each valid step, U_t contains the pre-treatment input available to the model, the time index, and any availability mask. In the unmasked case, $U_t = (t, \bar{\mathbf{H}}_t)$. The masking distribution is fixed independently of the model parameters; its randomness is included in the expectations. Define a probability law on $(U, A, \mathbf{X}, \mathbf{Y}^+)$ by

$$\mathbb{E}_\star f(U, A, \mathbf{X}, \mathbf{Y}^+) = \frac{1}{S}\mathbb{E}\left[\sum_{t=1}^{T} f(U_t, A_t, \mathbf{X}_t, \mathbf{Y}_{t+1})\right]. \tag{24}$$

The resulting time weights are $\mathbb{P}(T \geq t)/S$, preserving the contribution of each valid step in Eq. (10). Within-trajectory dependence is allowed. Write $Z = \Phi(U)$ for the representation at the selected step, $g_\theta(U, A) = G_Y(Z, A)$ for the prediction, and $r_\theta(U)$ for the reconstruction decoder output at that step, with $\theta = (\Phi, G_Y, G_X)$. This notation includes the causal state of the LSTM decoder. The model inputs and reconstruction targets are those used in the loss.

Set $K = d_a$ and define

$$\pi_j = \mathbb{P}_\star(A = a_j) > 0, \qquad P_j^\Phi = \mathcal{L}_\star(Z \mid A = a_j), \qquad Q^\Phi = \sum_{j=1}^{K} \pi_j P_j^\Phi. \tag{25}$$

The mixture Q^Φ dominates every P_j^Φ, so a common dominating measure always exists. Since Q^Φ dominates every P_j^Φ, we take all densities with respect to a common dominating measure. Write p_j^Φ, q^Φ for densities relative to any such measure ν. Define

$$\begin{aligned} R_Y(\theta) &= \mathbb{E}_\star\|\mathbf{Y}^+ - g_\theta(U, A)\|_2^2, \\ m(U, A) &= \mathbb{E}_\star[\mathbf{Y}^+ \mid U, A], \\ R_P(\theta) &= \mathbb{E}_\star\|m(U, A) - g_\theta(U, A)\|_2^2, \\ R_X(\theta) &= \mathbb{E}_\star\|\mathbf{X} - r_\theta(U)\|_2^2, \qquad D(\Phi) = \sum_{j=1}^{K} \mathrm{KL}(Q^\Phi \| P_j^\Phi). \end{aligned} \tag{26}$$

Assume $\mathbb{E}_\star(\|\mathbf{Y}^+\|_2^2 + \|\mathbf{X}\|_2^2) < \infty$. Risks and KL divergences may otherwise take the value $+\infty$. Throughout, $\mathrm{TV}(P, Q) = \sup_C |P(C) - Q(C)|$, and logarithms are natural.

Assumption E.1 (Multi-task comparator)**.** There is a comparator $\widetilde{\theta} = (\widetilde{\Phi}, \widetilde{G}_Y, \widetilde{G}_X)$ in the same model classes with

$$R_P(\widetilde{\theta}) \leq \rho, \qquad D(\widetilde{\Phi}) \leq \delta, \qquad R_X(\widetilde{\theta}) \leq \epsilon, \quad 0 \leq \rho, \delta, \epsilon < \infty. \tag{27}$$

The theorem holds for any such comparator. If the classes contain constant representations and constant prediction and reconstruction outputs, the moment assumption supplies a finite comparator with $D = 0$. More informative comparators capture the tradeoff between predictive accuracy, treatment balance, and reconstruction of treatment-associated covariates within the model class.

Lemma E.1 (Outcome risk decomposition)**.** *For every model θ,*

$$R_Y(\theta) = \sigma^2 + R_P(\theta), \qquad \sigma^2 = \mathbb{E}_\star\|\mathbf{Y}^+ - m(U, A)\|_2^2. \tag{28}$$

The noise term is finite and independent of the model parameters.

Proof. Conditioning first on (U, A) gives

$$\mathbb{E}_\star[\|\mathbf{Y}^+ - g_\theta(U, A)\|_2^2 \mid U, A] = \mathbb{E}_\star[\|\mathbf{Y}^+ - m(U, A)\|_2^2 \mid U, A] + \|m(U, A) - g_\theta(U, A)\|_2^2,$$

because the conditional residual has mean zero. The identity follows by taking expectations. The derivation remains valid even when the prediction risk is infinite, and it does not rely on an undefined unconditional cross term. □

Lemma E.2 (Bayes treatment classifier; Melnychuk et al., 2022)**.** *For a fixed representation, the minimizer of pooled treatment cross-entropy over all measurable probability-valued classifiers is*

$$G_A^{*,j}(z) = \mathbb{P}_\star(A = a_j \mid Z = z) = \frac{\pi_j p_j^\Phi(z)}{q^\Phi(z)}, \qquad j = 1, \dots, K, \tag{29}$$

for Q^Φ-almost every z.

Proof. Let $\eta(z)$ be the conditional treatment probability vector. For any classifier g, its conditional cross-entropy equals $-\sum_j \eta_j(z) \log \eta_j(z) + \mathrm{KL}(\eta(z) \| g(z))$. Nonnegativity of KL identifies the minimizer. Bayes' rule yields the displayed density ratio. □

This is the argument of Lemma F.1 of Melnychuk et al. (2022), applied to the pooled law. Every classifier-dependent objective below is evaluated at this population response. For a restricted classifier class, we assume that this response is realizable and attained. Pooling follows from the shared classifier's input Z and the time-summed cross-entropy used in training.

Lemma E.3 (Reverse KL identity)**.** *At the pooled Bayes response,*

$$\mathcal{L}_{cf}(\Phi, G_A^*) = -\frac{1}{K} \mathbb{E}_{Z \sim Q^\Phi} \sum_{j=1}^K \log G_A^{*,j}(Z) = \frac{1}{K} \left(D(\Phi) - \sum_{j=1}^K \log \pi_j \right). \tag{30}$$

Proof. Substitute Eq. (29) to obtain

$$-\frac{1}{K} \sum_j \int q^\Phi \log \frac{\pi_j p_j^\Phi}{q^\Phi} \, \mathrm{d}\nu = \frac{1}{K} \sum_j \mathrm{KL}(Q^\Phi \| P_j^\Phi) - \frac{1}{K} \sum_j \log \pi_j.$$

The equality holds in the extended real sense, including when P_j^Φ vanishes on a set of positive Q^Φ mass. □

The full population counterpart of Eq. (10), with the shared classifier at its pooled response, is

$$\mathcal{F}(\theta) = S\{R_Y(\theta) + \alpha_a \mathcal{L}_{cf}(\Phi, G_A^*) + \alpha_x R_X(\theta)\}. \tag{31}$$

With $\mu = K/\alpha_a$, $\lambda = K\alpha_x/\alpha_a$, and $J^{\mathrm{MTC}}(\theta) = \mu R_P(\theta) + D(\Phi) + \lambda R_X(\theta)$, the two lemmas imply

$$\frac{\mu}{S} \mathcal{F}(\theta) = \mu\sigma^2 + J^{\mathrm{MTC}}(\theta) - \sum_{j=1}^K \log \pi_j. \tag{32}$$

The prior and noise terms are independent of θ, so the two objectives have the same minimizers.

Theorem 1 (Joint MTC Risk Bound)**.** *Under the pooled-law and moment conditions above, evaluate classifiers at their pooled Bayes responses. Let $\alpha_a, \alpha_x > 0$, $\zeta \geq 0$, and assume*

$$\mathcal{F}(\widehat{\theta}) \leq \inf_\theta \mathcal{F}(\theta) + \zeta. \tag{33}$$

Under Assumption E.1, put $B = \mu\rho + \delta + \lambda\epsilon + \mu\zeta/S$. Then

$$\begin{aligned} J^{\mathrm{MTC}}(\widehat{\theta}) \leq B, \qquad R_P(\widehat{\theta}) \leq B/\mu, \qquad R_X(\widehat{\theta}) \leq B/\lambda, \\ \mathrm{TV}(P_j^{\widehat{\Phi}}, P_k^{\widehat{\Phi}}) \leq \min\{1, \sqrt{B}\} \quad \textit{for all } j, k, \\ R_Y(\widehat{\theta}) \leq \sigma^2 + B/\mu. \end{aligned} \tag{34}$$

Proof. Equation (32) transfers the optimality tolerance to J^{MTC} as $\mu\zeta/S$. Comparing to $\widetilde{\theta}$ yields

$$J^{\mathrm{MTC}}(\widehat{\theta}) \leq \inf_\theta J^{\mathrm{MTC}}(\theta) + \frac{\mu\zeta}{S} \leq \mu\rho + \delta + \lambda\epsilon + \frac{\mu\zeta}{S} = B. \tag{35}$$

All three summands are nonnegative. Thus $R_P \leq B/\mu$, $R_X \leq B/\lambda$, and $D \leq B$ at $\widehat{\theta}$. For $j \neq k$, write $a = \mathrm{KL}(Q^{\widehat{\Phi}} \| P_j^{\widehat{\Phi}})$ and $b = \mathrm{KL}(Q^{\widehat{\Phi}} \| P_k^{\widehat{\Phi}})$. Pinsker's inequality, symmetry of TV, and its triangle inequality give

$$\mathrm{TV}(P_j^{\widehat{\Phi}}, P_k^{\widehat{\Phi}}) \leq \sqrt{a/2} + \sqrt{b/2} \leq \sqrt{a + b} \leq \sqrt{D(\widehat{\Phi})} \leq \sqrt{B}.$$

Here the two distinct KL terms belong to D; the case $j = k$ is immediate. Also $\mathrm{TV} \leq 1$. The outcome risk bound follows from Lemma E.1. □

Interpretation. The primary loss controls approximation of the conditional outcome mean, the adversarial loss controls pooled treatment-domain discrepancy, and the auxiliary loss controls reconstruction. The comparator makes their interaction explicit: reducing one error can increase the attainable levels of the others. The guarantee concerns the pooled law in Eq. (24); per-time balance is a stronger property. For a single fixed time, the same proof applies with $S = 1$ and that time's observation law. With masked inputs, $m(U, A)$ is the conditional mean given the supplied input; the corresponding σ^2 includes uncertainty due to unavailable history.

The theorem is a fixed-weight population comparison with a Bayes treatment response and population optimality tolerance ζ. Adaptive regularization selects successive weight pairs, while the displayed bound applies to each pair satisfying the theorem's conditions. At $\alpha_x = 0$, the analogous comparison retains only the prediction and balancing terms; the balance analysis requires $\alpha_a > 0$. Causal identification uses the history-based assumptions in Appendix B, separately from this characterization of the training risks. For concurrent binary interventions, the treatment classes are their joint configurations, with positivity imposed on those joint classes.

E.1 Derivation of the Counterfactual Risk Bound

We connect the joint MTC objective to one-step counterfactual prediction under a target treatment policy. All expectations below use the pooled training-step law in Eq. (24). Let H denote the complete pre-treatment history, including the selected time index, and let U be the input supplied to the model. Define the observational propensity and conditional outcome mean by

$$e_j(H) = \mathbb{P}_\star(A = a_j \mid H), \qquad m_H(H, a_j) = \mathbb{E}_\star[\mathbf{Y}^+ \mid H, A = a_j]. \tag{36}$$

Under the consistency, one-step conditional ignorability, and overlap conditions in Appendix B, applied to this pooled history population, $m_H(H, a_j) = \mathbb{E}_\star[\mathbf{Y}^+(a_j) \mid H]$ almost surely. Let $q_j(H) \geq 0$, $\sum_j q_j(H) = 1$, specify a target distribution over the current treatment. We evaluate the conditional-mean prediction risk

$$\mathcal{E}_q(\theta) = \mathbb{E}_{\star,H,U}\left[\sum_{j=1}^{K} q_j(H)\|m_H(H, a_j) - g_\theta(U, a_j)\|_2^2\right]. \tag{37}$$

This target preserves the observed-history distribution and changes the current treatment assignment.

Assumption E.2 (Target-policy coverage and input availability)**.** The supplied input is generated by a fixed kernel $U \sim \mathcal{K}(\cdot \mid H)$, with its randomness independent of treatment and potential outcomes conditional on H. The same kernel is used under the observational and target treatment policies. For some finite C_q, the target policy satisfies

$$q_j(H) \leq C_q e_j(H) \quad \text{almost surely}, \qquad j = 1, \ldots, K. \tag{38}$$

The unmasked case $U = H$ satisfies the input-kernel condition automatically. The coverage condition bounds the target-to-observational treatment ratio; it allows small propensities where the target policy assigns correspondingly small probability.

Write $m_U(U, A) = \mathbb{E}_\star[\mathbf{Y}^+ \mid U, A]$, which is the conditional mean m in Eq. (26), and define

$$\Delta_{\text{in}} = \mathbb{E}_\star\|m_H(H, A) - m_U(U, A)\|_2^2. \tag{39}$$

This term measures the prognostic information lost when the complete history is reduced to the supplied input. It depends on the data and input kernel, independently of the learned parameters, and equals zero when $U = H$.

Proposition 2 (One-step counterfactual risk)**.** *Under the identification conditions above, Assumption E.2, and the pooled second-moment condition, every predictor with finite factual risk satisfies*

$$\mathcal{E}_q(\theta) \leq C_q\{R_P(\theta) + \Delta_{\text{in}}\}. \tag{40}$$

In particular, a model satisfying Theorem 1 obeys

$$\begin{aligned}\mathcal{E}_q(\widehat{\theta}) &\leq C_q\left(\frac{B}{\mu} + \Delta_{\text{in}}\right) \\ &= C_q\left(\rho + \frac{\alpha_a}{K}\delta + \alpha_x\epsilon + \frac{\zeta}{S} + \Delta_{\text{in}}\right).\end{aligned} \tag{41}$$

Proof. The input-kernel condition gives $\mathbb{E}_\star[\mathbf{Y}^+ \mid H, U, A] = m_H(H, A)$. The tower property therefore yields $\mathbb{E}_\star[m_H(H, A) \mid U, A] = m_U(U, A)$. Expanding the squared prediction error around m_U, the cross term has zero conditional expectation given (U, A). Consequently,

$$\begin{aligned} R_H(\theta) &:= \mathbb{E}_\star \|m_H(H, A) - g_\theta(U, A)\|_2^2 \\ &= \mathbb{E}_\star \|m_H(H, A) - m_U(U, A)\|_2^2 + \mathbb{E}_\star \|m_U(U, A) - g_\theta(U, A)\|_2^2 \\ &= \Delta_{\text{in}} + R_P(\theta). \end{aligned} \tag{42}$$

All terms are integrable under the moment and finite-risk conditions. Next, the kernel condition also gives $\mathbb{P}_\star(A = a_j \mid H, U) = e_j(H)$. Conditioning on (H, U) and summing over the factual treatment establishes

$$\mathcal{E}_q(\theta) = \mathbb{E}_\star \left[\frac{q_A(H)}{e_A(H)} \|m_H(H, A) - g_\theta(U, A)\|_2^2 \right], \tag{43}$$

where $q_A = q_j$ and $e_A = e_j$ when $A = a_j$. The coverage condition bounds this expectation by $C_q R_H(\theta)$. Combining with Eq. (42) proves Eq. (40). Finally, substitute $R_P(\widehat{\theta}) \leq B/\mu$ and $B/\mu = \rho + (\alpha_a/K)\delta + \alpha_x \epsilon + \zeta/S$ from Theorem 1. □

For unmasked inputs, the bound reduces to $\mathcal{E}_q(\widehat{\theta}) \leq C_q B/\mu$. For uniform treatment evaluation, $q_j = 1/K$, a propensity bound $e_j \geq \eta > 0$ gives $C_q \leq 1/(K\eta)$. More generally, finite C_q is a quantitative coverage condition beyond pointwise positivity. The two factors in Eq. (40) separate target-policy coverage from prediction error and the information loss associated with the supplied input. For masked histories, the kernel assumption specifies the additional condition under which the MTC approximation guarantee transfers to complete-history counterfactual means.

F HYPERPARAMETER TUNING

We performed hyperparameter optimization for CM and all baselines using a random search strategy (Bergstra & Bengio, 2012), with the selection criterion based on the 1-step-ahead prediction RMSE on the validation set. The ranges for the random searches for RMSNs, CRN, G-Net, CT and IGC-Net are provided in Tables 5, 6, 7, 8, and 9, respectively. The hyperparameter search space for CM is defined in Table 10. For each Mamba subnetwork within GCSI-Mamba, the hyperparameter search space includes the state dimension *d state* and the rank *dt rank*.

During hyperparameter optimization, we tailored the search ranges for each combination of model and dataset to account for variations in input dimensions across different models, while consistently ensuring comparable parameter scales and network depths to maintain fairness in model comparison. Specifically, for the MIMIC-III datasets, we narrowed the search space for hidden unit sizes in representation modules and fully connected layers. The number of training epochs was likewise adjusted per dataset to align with the complexity of the task. Here, FS denotes data from the fully-synthetic TG simulator; SS denotes semi-synthetic data based on real-world MIMIC-III covariates; and RW denotes real-world MIMIC-III data.

For each model and dataset, the configuration achieving the lowest 1-step-ahead RMSE on the validation set was selected for final evaluation. The selected configuration was then used consistently across all projection horizons, avoiding separate tuning for individual horizons and ensuring a uniform evaluation protocol across methods.

Table 5: Hyperparameter search ranges for RMSNs. The symbols N_{Pt}, N_{Ph}, N_E, and N_D denote the propensity treatment network, propensity history network, encoder, and decoder, respectively. Unless otherwise specified, all subnetworks use the same hyperparameter range.

Hyperparameter	**FS Data**	**SS Data**	**RW Data**
LSTM layers	1	1, 2	1, 2
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size (N_{Pt}, N_{Ph}, N_E)	64, 128, 256	64, 128, 256	64, 128, 256
Minibatch size (N_D)	256, 512, 1024	256, 512, 1024	256, 512, 1024
LSTM hidden units (N_{Pt})	2, 4, 8, 12, 16	1, 3, 6	1, 2, 4
LSTM hidden units (N_{Ph}, N_E)	2, 4, 8, 12, 16	37, 74, 148	36, 72, 144
LSTM hidden units (N_D)	4, 8, 16, 32, 64	49, 98, 196	47, 94, 188
LSTM dropout rate	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5
Max gradient norm	0.5, 1.0, 2.0, 4.0	0.5, 1.0, 2.0, 4.0	0.5, 1.0, 2.0, 4.0
Random search iterations (N_{Pt}, N_{Ph}, N_E)	50	50	50
Random search iterations (N_D)	20	20	20
Number of epochs	50	50	200

Table 6: Hyperparameter search ranges for CRN. The symbols N_E and N_D denote the encoder and decoder subnetworks, respectively. Unless otherwise specified, both subnetworks use the same hyperparameter range.

Hyperparameter	**FS Data**	**SS Data**	**RW Data**
LSTM layers	1	1, 2	1, 2
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size (N_E)	64, 128, 256	64, 128, 256	64, 128, 256
Minibatch size (N_D)	256, 512, 1024	256, 512, 1024	256, 512, 1024
LSTM hidden units (N_E)	3, 6, 12, 18, 24	37, 74, 148	36, 72, 144
LSTM hidden units (N_D)	d_r^E	d_r^E	d_r^E
BR size d_r^E (N_E)	3, 6, 12, 18, 24	37, 74, 148	36, 72, 144
BR size d_r^D (N_D)	3, 6, 12, 18, 24	49, 98, 196	47, 94, 188
FC hidden units (N_E)	$0.5d_r^E$, $1d_r^E$, $2d_r^E$, $3d_r^E$, $4d_r^E$	$0.5d_r^E$, $1d_r^E$, $2d_r^E$	$0.5d_r^E$, $1d_r^E$, $2d_r^E$
FC hidden units (N_D)	$0.5d_r^D$, $1d_r^D$, $2d_r^D$, $3d_r^D$, $4d_r^D$	$0.5d_r^D$, $1d_r^D$, $2d_r^D$	$0.5d_r^D$, $1d_r^D$, $2d_r^D$
LSTM dropout rate	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5
Random search iterations (N_E)	50	50	50
Random search iterations (N_D)	30	30	30
Number of epochs	50	100	200

Table 7: Hyperparameter search ranges for G-Net.

Hyperparameter	**FS Data**	**SS Data**	**RW Data**
LSTM layers	1	1, 2	1, 2
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256	64, 128, 256	64, 128, 256
LSTM hidden units	3, 6, 12, 18, 24	37, 74, 148	36, 72, 144
LSTM output size d_o	3, 6, 12, 18, 24	37, 74, 148	36, 72, 144
FC hidden units	$0.5d_o$, $1d_o$, $2d_o$, $3d_o$, $4d_o$	$0.5d_o$, $1d_o$, $2d_o$	$0.5d_o$, $1d_o$, $2d_o$
LSTM dropout rate	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5
Random search iterations	50	50	50
Number of epochs	50	100	200

Table 8: Hyperparameter search ranges for CT.

Hyperparameter	**FS Data**	**SS Data**	**RW Data**
Transformer blocks	1	1, 2	1, 2
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256	32, 64	32, 64
Attention heads	1	2, 3	2, 3
Transformer units	4, 8, 12, 16	24, 48, 64	24, 48, 64
BR size d_r	2, 4, 8, 12, 16	22, 44, 88	22, 44, 88
FC hidden units	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$	$0.5d_r$, $1d_r$, $2d_r$	$0.5d_r$, $1d_r$, $2d_r$
Sequential dropout rate	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5
Max position encoding	15	20	30
Random search iterations	50	50	50
Number of epochs	50	100	300

Table 9: Hyperparameter search ranges for IGC-Net.

Hyperparameter	**FS Data**	**SS Data**	**RW Data**
Transformer blocks	1	1, 2	1, 2
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256	32, 64	32, 64
Attention heads	1	2, 3	2, 3
Transformer units	6, 12, 18, 24	49, 98	24, 48, 64
Hidden Representation size d_r	3, 6, 12, 18, 24	24, 49, 98	22, 44, 88
FC hidden units	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$	$0.5d_r$, $1d_r$, $2d_r$	$0.5d_r$, $1d_r$, $2d_r$
Sequential dropout rate	0.1, 0.2	0.1, 0.2	0.1, 0.2
Max position encoding	15	30	30
Random search iterations	50	50	50
Number of epochs	50	100	100

Table 10: Hyperparameter search ranges used for the CM experiments.

Hyperparameter	**FS Data**	**SS Data**	**RW Data**
Mamba blocks	1	1, 2	1, 2
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size	64, 128	32, 64	32, 64
Mamba units(*d_model*)	4, 8, 12, 16	12, 24, 48	16, 32, 64
BR size d_r	2, 4, 8, 12, 16	22, 44, 88	16, 32, 64
FC hidden units	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$	$0.5d_r$, $1d_r$, $2d_r$	$0.5d_r$, $1d_r$, $2d_r$
Dropout rate	0.1, 0.2	0.1, 0.2	0.1, 0.2
Use Mamba dropout	True	False	False
`dt_rank`	1, 2, 4	1, 2, 4	1, 2, 4
`d_state`	4, 8	4, 8, 16	4, 8, 16
GCSI rank ratio	0.25	0.25	0.25
GCSI gate bias	-2.0	-2.0	-2.0
GCSI eps	10^{-8}	10^{-8}	10^{-8}
Covariate reconstruction head	1	1	1
Adversarial task weight α_a	0.01	0.01	0.01
Auxiliary task weight α_x	0	0.0001, 0.0002, 0.0005	0.0001, 0.0002, 0.0005
Random search iterations	50	50	50
Number of epochs	50	100	100

G EXTRA DETAILS OF TRAINING PROCESS

To improve generalization and stabilize the training process, CM utilizes a combination of regularization and stabilization techniques, including Masked Mini-Batch Augmentation (Melnychuk et al., 2022), Exponential Moving Average (EMA) (Tarvainen & Valpola, 2017; Yazıcı et al., 2019), Adaptive Regularization (Darken & Moody, 1991; Ganin et al., 2016), and Teacher Forcing (Williams & Zipser, 1989).

G.1 MASKED MINI-BATCH AUGMENTATION

Following Melnychuk et al. (2022), we apply mini-batch augmentation with masking to both CM and CT. The procedure creates partially observed trajectories by masking a suffix of the time-varying covariates, thereby simulating the unavailable future covariates encountered in multi-step prediction.

For 1-step-ahead prediction, all required time-varying covariates $\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{T^{(i)}}^{(i)}$ are available in the observational data $\mathcal{D}$. However, for τ-step-ahead prediction ($\tau \geq 2$), the future covariates $\mathbf{x}_{T^{(i)}+1}^{(i)}, \ldots, \mathbf{x}_{T^{(i)}+\tau-1}^{(i)}$ are not available in $\mathcal{D}$. To address this discrepancy, masked mini-batch augmentation is employed during training to simulate the absence of future covariates. Specifically, for each training sample, we uniformly sample a starting time index $1 \leq t_s \leq T^{(i)}$, and then mask (i.e., treat as unobserved) all subsequent time-varying covariates from t_s onward, namely $\mathbf{x}_{t_s}^{(i)}, \ldots, \mathbf{x}_{T^{(i)}}^{(i)}$.

G.2 EXPONENTIAL MOVING AVERAGE (EMA)

Following Melnychuk et al. (2022), CM employs Exponential Moving Average (EMA) during training to stabilize its learning procedure. Applying EMA to the parameters smooths parameter trajectories around an equilibrium and can accelerate convergence (Yazıcı et al., 2019). In our implementation, EMA is applied to all trainable parameters $\theta = (\theta_Y, \theta_A, \theta_X, \theta_R)$ in CM. The parameters are updated throughout the training process using the following formula

$$\theta_{\text{EMA}}^{(j)} = \beta \theta_{\text{EMA}}^{(j-1)} + (1-\beta)\theta^{(j)}, \tag{44}$$

where j denotes the index of the optimization step and β is the exponential smoothing parameter, set to 0.99 for both CT and CM. At the start of training, the EMA parameter is initialized as $\theta_{\text{EMA}}^{(0)} = \theta^{(0)}$.

G.3 ADAPTIVE REGULARIZATION

To achieve coordinated optimization among different tasks during model training and to ensure that the dynamic influences of adversarial and auxiliary tasks align with the training phase of the primary task, we apply adaptive regularization to the regularization coefficients of adversarial and auxiliary tasks. Specifically, the adversarial task coefficient α_a follows an increasing schedule (Ganin et al., 2016), whereas the auxiliary task coefficient α_x follows a schedule of sigmoid decay (Darken & Moody, 1991). Table 10 provides their initial values or tuning ranges. The schedules are given by

$$\alpha_a^{(e)} = \alpha_a \left(\frac{2}{1+\exp(-10e/n_e)} - 1 \right), \qquad \alpha_x^{(e)} = \alpha_x \left(2 - \frac{2}{1+\exp(-10e/n_e)} \right), \tag{45}$$

where $e \in \{1, \ldots, n_e\}$ is the current epoch index. The regularization coefficient λ for both CT and CRN also follows the increasing schedule described above.

G.4 TEACHER FORCING

Teacher forcing is used during training for CM and the autoregressive baselines, including RMSNs, CRN, G-Net, and CT. This technique uses the ground truth observation from the previous time step as input when predicting the next step. During evaluation, the model operates in autoregressive mode, iteratively using its own previous predictions as inputs for subsequent time steps to perform multi-step-ahead forecasting. This approach enables the model to generate coherent multi-step-ahead predictions without access to true future observations, which is a standard practice in sequential prediction tasks (Taieb & Hyndman, 2014).

H BASELINE DETAILS

We selected five representative methods as baselines, which were evaluated under the same dataset configuration as used in our work. These include: Recurrent Marginal Structural Networks (RMSNs) (Lim et al., 2018); Counterfactual Recurrent Networks (CRN) (Bica et al., 2020b); G-Net (Li et al., 2021); Causal Transformer (CT) (Melnychuk et al., 2022); and IGC-Net (Hess et al., 2026). Table 11 provides a comparative summary of these methods for estimating counterfactual outcomes over time. The hyperparameter settings for all baselines can be found in Appendix F. Further details for each baseline are provided in the following section.

Table 11: A comparative summary of methods for estimating counterfactual outcomes over time included in our experiments. The comparison is based on four core components of each model: (1) the model backbone which defines the core sequence modeling architecture; (2) the strategy used for multi-step-ahead counterfactual outcome prediction; (3) the approach for handling time-varying confounding bias; and (4) the overall training and inference structure of the model. The evaluated CM uses the GCSI-Mamba backbone, the MTC balancing module, and an end-to-end structure.

	Model Backbone	Strategy for Outcome Prediction	Handling Time-Varying Confounding	Structure
RMSNs	LSTM	Autoregressive	IPTW	4 subnetworks
CRN	LSTM	Autoregressive	Balanced representation (GR)	encoder, decoder
G-Net	LSTM	Autoregressive	G-Computation	end-to-end
CT	Transformer	Autoregressive	Balanced representation (CDC)	end-to-end
IGC-Net	Multi-input Transformer	Direct CAPO (single pass)	Iterative G-computation	end-to-end
CM	GCSI-Mamba	Autoregressive	Balanced representation (MTC)	end-to-end

H.1 RECURRENT MARGINAL STRUCTURAL NETWORKS (RMSNS)

RMSNs use a sequence-to-sequence architecture for multiple binary treatments (Lim et al., 2018). In the uncensored setting considered here, the architecture comprises four LSTM subnetworks: a propensity treatment network, a propensity history network, an encoder, and a decoder. The encoder summarizes the observed history $\bar{\mathbf{H}}_t$ and performs one-step-ahead prediction. A linear memory adapter maps the encoder state to the decoder state, after which the decoder autoregressively predicts outcomes under the planned treatment sequence. The original RMSN additionally uses networks for censoring propensity when right censoring is modeled.

RMSNs adjust for time-varying confounding through stabilized inverse probability of treatment weighting (IPTW) (Robins et al., 2000). Following Lim et al. (2018), define

$$SW(t,k) = \prod_{n=t}^{t+k} \frac{f(\mathbf{A}_n \mid \bar{\mathbf{A}}_{n-1})}{f(\mathbf{A}_n \mid \bar{\mathbf{H}}_n)}, \qquad k \geq 0. \tag{46}$$

Consequently, the weight associated with the horizon-τ target $\mathbf{Y}_{t+\tau}$ and planned treatments $\bar{\mathbf{A}}_{t:t+\tau-1}$ is $SW(t,\tau-1)$, not $SW(t,\tau)$. For multiple binary treatments, each LSTM produces a vector of sigmoid probabilities and the joint probability is factorized across treatment components, as in the original RMSN implementation:

$$f(\mathbf{A}_n \mid \mathcal{C}_n) = \prod_{\ell=1}^{d_b} f(A_n^\ell \mid \mathcal{C}_n), \qquad \mathcal{C}_n \in \{\bar{\mathbf{A}}_{n-1}, \bar{\mathbf{H}}_n\}. \tag{47}$$

A softmax output is used instead when the treatment is a single categorical variable. The numerator and denominator probabilities in Equation (46) are estimated by the propensity-treatment and propensity-history LSTMs, respectively. The resulting weights are normalized by their mean and truncated at the 1st and 99th percentiles.

Training is sequential. The propensity networks are fitted first. The encoder loss for one step prediction is weighted by $SW(t,0)$, while the decoder loss for prediction step τ is weighted by $SW(t,\tau-1)$. Rolling-origin subsequences are then initialized with representations from the pre-trained encoder and used to train the decoder.

H.2 COUNTERFACTUAL RECURRENT NETWORK (CRN)

CRN addresses time-varying confounding by learning balanced, treatment-invariant representations through adversarial training with gradient reversal (GR) (Ganin & Lempitsky, 2015). CRN adopts a streamlined encoder-decoder framework where both components are single-layer LSTMs. Notably, it omits the memory adapter used in RMSNs, and instead directly sets the hidden unit size d_h of the decoder to equal the dimensionality of the balanced output of encoder.

The architecture processes the hidden state $\mathbf{h}_t$ at each time step t through a shared linear layer, yielding a latent representation $\mathbf{\Phi}_t$. This representation is subsequently fed into two separate fully connected networks: G_Y, which forecasts the subsequent outcome $\mathbf{Y}_{t+1}$; G_A , which estimates the current treatment $\mathbf{A}_t$. The training of both the encoder and the decoder is governed by a common objective function that combines the two prediction tasks, defined as

$$\mathcal{L} = \left\| \mathbf{Y}_{t+1} - G_Y \left(\mathbf{\Phi}_t, \mathbf{A}_t \right) \right\|^2 - \lambda \sum_{j=1}^{d_a} \mathbb{I}_{[\mathbf{A}_t = a_j]} \log G_A^j \left(\mathbf{\Phi}_t \right), \tag{48}$$

where $\mathbb{I}_{[.]}$ denotes the indicator function. The training adversarially updates G_A and $\mathbf{\Phi}_t$ by minimizing and maximizing the same treatment prediction cross-entropy, respectively, via the loss $\mathcal{L}$ with the hyperparameter λ controlling the adversarial balance. We adopted $\lambda = 1$ as used in the CRN, following its sigmoid scheduling strategy during training (see Appendix G for details).

H.3 G-NET

To address time-varying confounding, G-Net is built upon the G-computation formula (Robins, 1986), formulated as

$$\begin{aligned} \mathbb{E}\big(\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}_{t:t+\tau-1}] \mid \bar{\mathbf{H}}_t\big) = \int_{\left(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}\right)^{\tau-1}} & \mathbb{E}\big(\mathbf{Y}_{t+\tau} \mid \bar{\mathbf{H}}_t, \bar{\mathbf{x}}_{t+1:t+\tau-1}, \bar{\mathbf{y}}_{t+1:t+\tau-1}, \bar{\mathbf{a}}_{t:t+\tau-1}\big) \\ & \times \prod_{j=t+1}^{t+\tau-1} \mathbb{P}\big(\mathbf{x}_j, \mathbf{y}_j \mid \bar{\mathbf{H}}_t, \bar{\mathbf{x}}_{t+1:j-1}, \bar{\mathbf{y}}_{t+1:j-1}, \bar{\mathbf{a}}_{t:j-1}\big)\, \mathrm{d}\bar{\mathbf{x}}_{t+1:t+\tau-1} \mathrm{d}\bar{\mathbf{y}}_{t+1:t+\tau-1}, \end{aligned} \tag{49}$$

which formally defines the conditional average counterfactual outcomes $\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}_{t:t+\tau-1}]|\bar{\mathbf{H}}_t$ in terms of the observational data distribution.

G-Net estimates counterfactual outcomes through a sequential procedure. It first learns the conditional distributions $\mathbb{P}(\mathbf{X}_j \mid \bar{\mathbf{H}}_t, \bar{\mathbf{x}}_{t+1:j-1}, \bar{\mathbf{a}}_{t:j-1})$ via a shared LSTM that models the corresponding conditional expectations $\mathbb{E}(\mathbf{X}_j \mid \bar{\mathbf{H}}_t, \bar{\mathbf{x}}_{t+1:j-1}, \bar{\mathbf{a}}_{t:j-1})$ jointly with outcome prediction. Sampling from these distributions is performed using the empirical distribution of residuals calculated on a holdout set (10% of the training data), which is excluded during the estimation step. Based on these sampled trajectories, Monte Carlo simulation is conducted following Equation (49) , and the counterfactual outcome $\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}_{t:t+\tau-1}]$ is obtained as the empirical mean over $M = 50$ Monte Carlo samples.

To ensure comparability with other baselines, we used one or two LSTM layers (consistent with the original works), augmented with an extra fully connected linear representation layer as Melnychuk et al. (2022). A subsequent network containing hidden units is then placed upon this representation layer.

H.4 CAUSAL TRANSFORMER (CT)

CT extends the CRN by introducing a Transformer-based architecture, representing the first adaptation of this model family (Vaswani et al., 2017) to causal inference tasks. Its core innovation is a multi-input Transformer block, comprising three dedicated subnetworks that employ cross-attention to effectively capture interactions among different predictors when extracting information from the patient history $\bar{\mathbf{H}}_t$. Similar to CRN, CT learns balanced representations that are non-predictive of treatment assignment, achieved via adversarial training using a domain confusion loss (Tzeng et al., 2015). Notably, CT is designed as an end-to-end framework, where representation learning and outcome prediction are jointly optimized.

At each time step t, the patient history $\bar{\mathbf{H}}_t$ is linearly projected and processed through this multi-input Transformer block, with the final output yielding the representation $\mathbf{\Phi}_t$. Two fully connected networks, G_A and G_Y, then use $\mathbf{\Phi}_t$ to predict the current treatment $\mathbf{A}_t$ and the next outcome $\mathbf{Y}_{t+1}$, respectively. Following the original CT formulation (Melnychuk et al., 2022), the factual outcome, treatment classification, and counterfactual domain confusion losses are

$$\mathcal{L}_{G_Y} = \|\mathbf{Y}_{t+1} - G_Y(\mathbf{\Phi}_t, \mathbf{A}_t)\|^2, \tag{50}$$

$$\mathcal{L}_{G_A} = -\sum_{j=1}^{d_a} \mathbb{I}_{[\mathbf{A}_t = a_j]} \log G_A^j(\mathbf{\Phi}_t), \qquad \mathcal{L}_{\text{conf}} = -\sum_{j=1}^{d_a} \frac{1}{d_a} \log G_A^j(\mathbf{\Phi}_t). \tag{51}$$

The treatment classifier minimizes $\mathcal{L}_{G_A}$, whereas the representation and outcome networks minimize

$$\mathcal{L}_{\text{CT}} = \mathcal{L}_{G_Y} + \lambda \mathcal{L}_{\text{conf}}, \tag{52}$$

using alternating parameter updates. Here, $G_A^j(\mathbf{\Phi}_t)$ is the predicted probability of treatment class a_j, and $\mathcal{L}_{\text{conf}}$ is the cross-entropy from a uniform distribution over the d_a treatment classes to the classifier output. The coefficient λ (serving the same purpose as α in the original CT paper), initialized to 0.01 following (Melnychuk et al., 2022) and set to increase according to a sigmoid schedule throughout training, regulates the intensity of this adversarial trade-off. To isolate the contribution of the MTC module independently of the CM backbone, we compare the original CT with a hybrid variant in which its CDC balancing objective is replaced by MTC while retaining the CT architecture. Further implementation details are provided in Appendix G.

H.5 IGC-NET

IGC-Net, short for Iterative G-Computation Network, is a neural end-to-end method for estimating conditional average potential outcomes (CAPOs) under time-varying confounding. Unlike G-Net, IGC-Net does not estimate or use Monte Carlo sampling from the high-dimensional distribution of future time-varying confounders. Instead, it rewrites the longitudinal G-computation formula in Eq. (49) as the following recursion of conditional expectations:

$$\begin{aligned} g_{t+\delta}^{\bar{\mathbf{a}}}(\bar{\mathbf{h}}_{t+\delta}^t) &= \mathbb{E}\big[G_{t+\delta+1}^{\bar{\mathbf{a}}} \,\big|\, \bar{\mathbf{H}}_{t+\delta}^t = \bar{\mathbf{h}}_{t+\delta}^t, \mathbf{A}_{t:t+\delta} = \mathbf{a}_{t:t+\delta}\big], \\ G_{t+\delta}^{\bar{\mathbf{a}}} &= g_{t+\delta}^{\bar{\mathbf{a}}}(\bar{\mathbf{H}}_{t+\delta}^t), \qquad \delta = 1, \ldots, \tau - 1, \\ G_{t+\tau}^{\bar{\mathbf{a}}} &= \mathbf{Y}_{t+\tau}. \end{aligned} \tag{53}$$

The architecture comprises a shared neural backbone z_ϕ and G-computation heads indexed by horizon, $\{g_\phi^\delta\}_{\delta=0}^{\tau-1}$. The main experimental implementation uses a connected multi-input Transformer to encode the outcome, covariate, and treatment histories, while the general framework also permits other backbones such as an LSTM.

Training alternates between a *generation step* and a *learning step*. In the generation step, the current network uses the planned treatment prefix to form detached predictions $\widetilde{G}_{t+\delta}^{\bar{\mathbf{a}}} = g_\phi^\delta(z_\phi(\bar{\mathbf{H}}_{t+\delta}^t, \mathbf{a}_{t:t+\delta-1}), \mathbf{a}_{t+\delta})$ of the intermediate pseudo-outcomes for $\delta = 1, \ldots, \tau - 1$. In the learning step, for $\delta = 0, \ldots, \tau - 1$, the model minimizes the squared error between $g_\phi^\delta(z_\phi(\bar{\mathbf{H}}_{t+\delta}), \mathbf{A}_{t+\delta})$ and the next target $\widetilde{G}_{t+\delta+1}^{\bar{\mathbf{a}}}$, where the terminal target is observed and initialized as $\widetilde{G}_{t+\tau}^{\bar{\mathbf{a}}} = G_{t+\tau}^{\bar{\mathbf{a}}} = \mathbf{Y}_{t+\tau}$. After the shared parameters are updated, the improved estimators generate the pseudo-outcomes for the next training iteration. Thus, the two steps implement regression-based iterative G-computation without treating generated pseudo-outcomes as differentiable targets.

During the inference phase, the conditional average potential outcome in IGC-Net is computed via

$$\hat{g}_t^{\bar{\mathbf{a}}_{t:t+\tau-1}}(\bar{\mathbf{h}}_t) = \hat{\mathbb{E}}\Big[G_{t+1}^{\bar{\mathbf{a}}_{t:t+\tau-1}} \mid \bar{\mathbf{H}}_t = \bar{\mathbf{h}}_t, \mathbf{A}_t = \mathbf{a}_t\Big] = g_\phi^0\big(z_\phi(\bar{\mathbf{h}}_t), \mathbf{a}_t\big). \tag{54}$$

In contrast to G-Net, which relies on sequential Monte Carlo simulation for multi-step-ahead prediction, IGC-Net returns the CAPO estimate in a single forward pass of the trained network rather than through an autoregressive rollout.

I BENCHMARK DETAILS

I.1 DETAILS ON EXPERIMENTS WITH FULLY-SYNTHETIC DATASET TUMOR GROWTH SIMULATOR

The fully-synthetic data is derived from the tumor growth model (TG simulator) introduced by Geng et al. (2017), which is capable of predicting tumor volume $\mathbf{Y}_{t+1}$ at $t+1$ days after cancer diagnosis based on covariates and treatment assignments. In addition, the model includes different distributions of tumor sizes based on the cancer stage at diagnosis. The TG simulator incorporates the following components:

- **Two binary treatments**. (1) radiotherapy ($\mathbf{A}_t^r$) exerts an immediate effect d_t on the volume of tumor at the next time step when administered to a patient; (2) chemotherapy ($\mathbf{A}_t^c$) modulates future volume of tumor via an exponential decay mechanism scaled by the scaling factor C_t;
- **1-dimensional Outcome**. The volume of tumor $\mathbf{Y}_{t+1}$ at time $t+1$. More specifically, the volume of tumor $\mathbf{Y}_{t+1}$ is simultaneously influenced by both $\mathbf{A}_t^r$ and $\mathbf{A}_t^c$ via the following process

$$\mathbf{Y}_{t+1} = \left(1 + \rho \log\left(\frac{K}{\mathbf{Y}_t}\right) - \beta_c C_t - \left(\alpha_r d_t + \beta_r d_t^2\right) + \varepsilon_t\right) \mathbf{Y}_t, \tag{55}$$

  where $\rho, K, \beta_c, \alpha_r, \beta_r$ are parameters simulating tumor volume changes and $\varepsilon_t \sim N(0, 0.01^2)$ is independently sampled noise. Here, the parameters β_c, α_r, and β_r capture the underlying heterogeneity in individual patient responses. They are realized by sampling from a truncated normal mixture model with three components.
- **Time-varying confounding and treatment assignment**. Time-varying confounding is introduced by a biased treatment assignment, identical for both treatments, i.e.,

$$\mathbf{A}_t^c, \mathbf{A}_t^r \sim \text{Bernoulli}\left(\sigma\left(\frac{\gamma}{D_{\max}}\left[\bar{D}_{15}(\bar{\mathbf{Y}}_{t-1}) - D_{\max}/2\right]\right)\right), \tag{56}$$

  where $\sigma(\cdot)$ is a sigmoid activation, $D_{\max}$ is the maximum tumor diameter, $\bar{D}_{15}(\bar{\mathbf{Y}}_{t-1})$ is the average tumor diameter over the last 15 days, and γ is the time-varying confounding level.

Overall, in the TG simulator, the key parameter γ governs the magnitude of the confounding effect, thereby controlling the extent of confounding bias. As the value of γ increases, the strength of time-varying confounding bias intensifies. When $\gamma = 0$, the treatment assignment is completely randomized and unaffected by any other factors.

We adopt a distinct strategy for treatment variable specification across methods. Specifically, RMSNs (Lim et al., 2018) are implemented with two binary treatments, while all other approaches use a single categorical treatment in

$$\mathcal{A} = \{(\mathbf{A}_t^c = 0, \mathbf{A}_t^r = 0), (\mathbf{A}_t^c = 1, \mathbf{A}_t^r = 0), (\mathbf{A}_t^c = 0, \mathbf{A}_t^r = 1), (\mathbf{A}_t^c = 1, \mathbf{A}_t^r = 1)\}.$$

For each patient in the test cohort, we generate a set of counterfactual trajectories at each time step, conditioned on the projection horizon τ. In 1-step-ahead prediction, the complete set of all possible counterfactual outcomes $\mathbf{Y}_{t+1}$ is simulated, representing the tumor volume under every feasible treatment combination. For multi-step-ahead prediction, the number of potential outcomes for $\mathbf{Y}_{t+2}, \ldots, \mathbf{Y}_{t+\tau_{\max}}$ grows exponentially with the projection horizon $\tau_{\max}$. Therefore, an alternative approach is adopted: *fixed treatment* (Hess et al., 2026). Specifically, one treatment is randomly selected from the aforementioned set $\mathcal{A}$ and remains fixed throughout the testing phase. This process ultimately yields $2(\tau_{\max} - 1)$ distinct trajectories.

In addition, for each time-varying confounding level γ, we simulated trajectories for 1,000 patients for training, 1,000 for validation, and 1,000 for testing, respectively. The length of each patient trajectory was capped at 30 time steps (some patients have shorter trajectories due to recovery or death). We report a normalized RMSE, where we normalize by the maximum tumor volume $V_{\max} = 1150\ \text{cm}^3$.

I.2 Details on Experiments with MIMIC-III Semi-Synthetic Dataset

The Medical Information Mart for Intensive Care III (MIMIC-III) database (Johnson et al., 2016) is a comprehensive, de-identified repository of electronic health records from intensive care unit (ICU) patients. It consolidates diverse clinical data types, including vital signs, medications, laboratory results, clinical notes, fluid balance, procedure and diagnosis codes, imaging reports, length of stay, and mortality outcomes, all collected from multiple hospital information systems and standardized to support secondary research. Consequently, MIMIC-III serves as a crucial benchmark for evaluating analytical models in complex, real-world clinical environments.

Following the experimental protocol established by Melnychuk et al. (2022), this study constructs a semi-synthetic dataset based on real-world clinical data. The dataset originates from the MIMIC-extract pipeline (Wang et al., 2020), which is applied to the MIMIC-III dataset for subsequent standardized preprocessing. Specifically, we employ the same data processing methods as Melnychuk et al. (2022): missing values are handled using forward and backward filling techniques, and all continuous time-varying features are standardized. From the raw data, following the studies in (Kuzmanovic et al., 2021; Melnychuk et al., 2022; Hatt & Feuerriegel, 2024; Hess et al., 2026; Wang et al., 2024), we extract 25 vital signs as time-varying covariates and retain 3 static covariates (gender, ethnicity, and age). All static variables are represented using one-hot encoding, resulting in a final 44-dimensional feature vector.

We extend the semi-synthetic simulation scheme of Schulam & Saria (2017) to generate counterfactual outcomes under endogenous and exogenous dependencies, and then process the trajectories by applying treatments sequentially. We posit a sparse dependency structure: outcomes are affected by only a limited number of past treatments and time-varying covariates. In turn, the assignment of treatment at any time depends on only a few prior outcomes and covariates. The steps proceed as follows:

- **Selection of samples**. A random sample of 1,000 patients was drawn, with the inclusion criterion of an ICU stay $T^{(i)} \geq 30$ hours. Their length of stay $T^{(i)}$ was then defined as the original stay truncated at 50 hours, thus constraining $T^{(i)}$ to the interval $[30, 50]$.
- d_y **untreated outcomes** $\mathbf{Z}_t^{j,(i)}, j = 1 \ldots, d_y$. The combined model for each patient i, integrating endogenous dynamics (B-spline(t) and random function $g^{j,(i)}(t)$), exogenous dependency $f_Z^j(\mathbf{X}_t^{(i)})$, and noise ε_t, is given by

$$\mathbf{Z}_t^{j,(i)} = \underbrace{\alpha_S^j \mathbf{B}\text{-spline}(t) + \alpha_g^j g^{j,(i)}(t)}_{\text{endogenous}} + \underbrace{\alpha_f^j f_Z^j(\mathbf{X}_t^{(i)})}_{\text{exogenous}} + \underbrace{\varepsilon_t}_{\text{noise}}, \tag{57}$$

  where $\varepsilon_t \sim N(0, 0.005^2)$ and α_S^j, α_g^j and α_f^j are weight parameters. The B-spline(t) basis is composed of three cubic splines, each capturing a distinct trend in ICU length of stay (encompassing a trajectory of rapid decline over the full ICU stay, one of mild decline, and a stable course); $g^{j,(i)}(\cdot)$ is sampled independently for each patient from a Gaussian process with a Matérn kernel; external effect $f_Z^j(\cdot)$ is modeled by employing a Random Fourier Features (RFF) approach to circumvent the need for complex Cholesky decomposition. This configuration is designed to capture multi-scale endogenous dynamics and selectively model exogenous influences on time-varying covariates.
- d_b **binary treatments** $\mathbf{A}_t^l, l = 1, \ldots, d_b$. The term $\mathbf{A}_t^l$ is defined through a two-step procedure: (1) incorporating confounding via the function $f_Y^l(\mathbf{X}_t)$ applied to a subset of current time-varying covariates, and (2) averaging over the corresponding subset of prior T_l treated outcomes $\bar{A}_{T_l}(\bar{\mathbf{Y}}_{t-1})$. We define

$$p_{\mathbf{A}_t^l} = \sigma\left(\gamma_A^l \bar{A}_{T_l}\left(\bar{\mathbf{Y}}_{t-1}\right) + \gamma_X^l f_Y^l(\mathbf{X}_t) + b_l\right), \tag{58}$$

$$\mathbf{A}_t^l \sim \text{Bernoulli}(p_{\mathbf{A}_t^l}), \tag{59}$$

  where $\sigma(\cdot)$ denotes the sigmoid function, γ_A^l and γ_X^l are confounding parameters, and b_l is a constant bias term. The function $f_Y^l(\cdot)$, like $f_Z^j(\cdot)$, is sampled from an RFF approximation of a Gaussian process (Melnychuk et al., 2022). For CM, the concurrent binary treatments at each time step are represented by their joint configuration as a single categorical treatment, yielding $d_a = 2^{d_b}$ treatment levels.

- **Outcome $\mathbf{Y}_t^j$ after treatment**. The administration of treatments begins at the initial time point, with the initial outcome $\mathbf{Y}_1 = \mathbf{Z}_1$. The observed outcome is derived by summing the untreated outcome $\mathbf{Z}_t^j$ and the simulated treatment effect $E^j(t)$, i.e.,

$$\mathbf{Y}_t^j = \mathbf{Z}_t^j + E^j(t), \tag{60}$$

  where $E^j(t)$ quantifies the treatment bias effect over a specified time window $t - w^l, \cdots, t$, i.e.,

$$E^j(t) = \sum_{i=t-w^l}^{t} \frac{\min_{l=1,\ldots,d_b} \mathbb{I}_{[\mathbf{A}_i^l=1]} p_{\mathbf{A}_i^l} \beta_{lj}}{(w^l - i)^2}, \tag{61}$$

  where β_{lj} is the maximum effect size of treatment l. The treatment effect takes one of two forms: a constant common to all outcomes j, or zero (the null scenario where the treatment has no effect).

This process yields the semi-synthetic benchmark based on MIMIC-III. To evaluate different models, we split the cohort of 1,000 patients into training, validation, and test sets using a 60%/20%/20% ratio, respectively. For 1-step-ahead predictions, we simulate all $2^3 = 8$ counterfactual trajectories. For $\tau_{\max}$-step-ahead prediction, we draw $\tau_{\max}$ random trajectory samples. The averaged Root Mean Square Error (RMSE) is then reported, computed specifically over the test data.

I.3 Details on Experiments with MIMIC-III Real-World Dataset

Following Kuzmanovic et al. (2021); Melnychuk et al. (2022); Hatt & Feuerriegel (2024), we extracted 25 indicators based on vital signs and 3 static attributes from MIMIC-III. Categorical features were encoded as indicator variables. These variables, which include dynamic covariates and characteristics that do not change over time, are treated as potential confounders. We consider two binary treatment interventions: vasopressor use and mechanical ventilation. The observed outcome is (diastolic) blood pressure, which may increase or decrease in response to treatment. This is a clinically crucial factor when anticipating patient trajectories under these interventions.

For the experimental setup, a cohort of 5,000 patients was constructed by randomly sampling individuals with intensive care unit (ICU) stays lasting at least 30 hours. The recorded ICU length of stay was truncated at a maximum of 60 hours for each patient. This cohort was subsequently divided into training, validation, and test subsets with a 70%/15%/15% split ratio. The evaluation procedure was adapted according to the projection horizon τ. In the case of 1-step-ahead prediction, all trajectories in the test set were used directly. For multi-step-ahead prediction ($\tau \geq 2$), we set $\tau_{\max} = 5$ and construct evaluation sub-trajectories using a rolling-origin scheme (Bergmeir & Benítez, 2012). For each prediction origin t_0, covariates up to t_0 remain observed, whereas future covariates after t_0 over the projection horizon are masked to prevent look-ahead bias (Kaufman et al., 2012). We report RMSE for all models under this masked evaluation protocol.

J ADDITIONAL RESULTS

In the following, we provide full results for τ-step-ahead prediction on fully-synthetic dataset from TG simulator with different levels of time-varying confounding γ under the *fixed treatment* setting (Table 12). Note that CM (ERM) uses the same hyperparameters as CM. The only difference is that we removed the MTC balancing module from CM, i.e., $\alpha_a = \alpha_x = 0$.

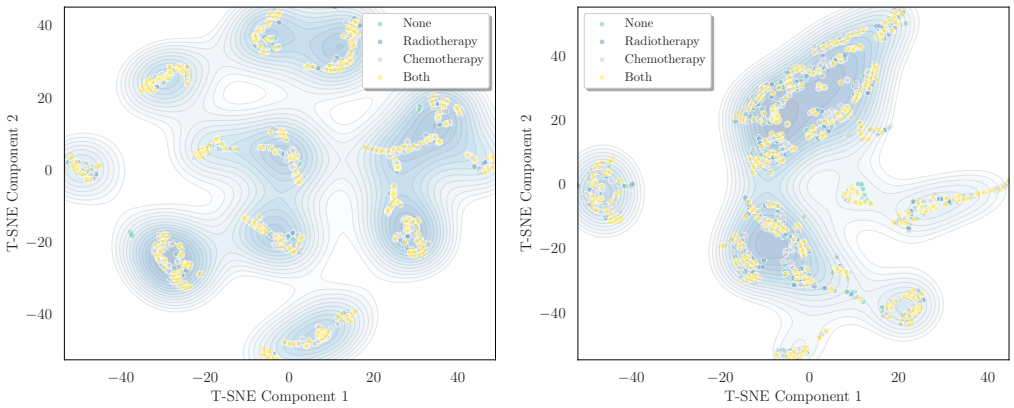
Under the *fixed treatment* setting, RMSE decreases with longer projection horizons, consistent with the anticipated reduction in tumor volume from treatment.

Table 12: Results for τ-step-ahead prediction on fully-synthetic dataset from TG simulator (lower values are better, with the best highlighted in bold). Shown: nRMSE as mean $\pm$ standard deviation over five runs.

		$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
$\gamma = 0$	RMSNs	1.16 ± 0.18	1.18 ± 0.09	0.92 ± 0.09	0.67 ± 0.07	0.49 ± 0.06	0.37 ± 0.09
	CRN	1.14 ± 0.13	1.17 ± 0.08	0.88 ± 0.08	$\mathbf{0.62 \pm 0.07}$	$\mathbf{0.43 \pm 0.06}$	$\mathbf{0.30 \pm 0.05}$
	G-Net	1.10 ± 0.15	1.24 ± 0.13	1.03 ± 0.18	0.81 ± 0.20	0.66 ± 0.21	0.55 ± 0.23
	CT	1.08 ± 0.14	1.15 ± 0.10	0.89 ± 0.09	0.66 ± 0.07	0.48 ± 0.07	0.34 ± 0.06
	IGC-Net	1.18 ± 0.15	1.28 ± 0.09	1.15 ± 0.23	0.65 ± 0.07	0.88 ± 0.71	1.29 ± 1.86
	CM (ERM)	$\mathbf{1.07 \pm 0.16}$	$\mathbf{1.11 \pm 0.08}$	$\mathbf{0.85 \pm 0.08}$	$\mathbf{0.62 \pm 0.07}$	0.44 ± 0.06	$\mathbf{0.30 \pm 0.05}$
	CM	$\mathbf{1.07 \pm 0.16}$	1.12 ± 0.08	0.86 ± 0.08	0.63 ± 0.08	0.45 ± 0.07	0.32 ± 0.06
$\gamma = 1$	RMSNs	1.19 ± 0.16	1.31 ± 0.05	1.00 ± 0.06	0.71 ± 0.06	0.49 ± 0.05	0.34 ± 0.04
	CRN	1.29 ± 0.21	1.25 ± 0.14	0.99 ± 0.10	0.75 ± 0.07	0.57 ± 0.08	0.46 ± 0.10
	G-Net	1.16 ± 0.11	1.35 ± 0.14	1.13 ± 0.18	0.91 ± 0.07	0.74 ± 0.25	0.64 ± 0.28
	CT	1.18 ± 0.12	1.24 ± 0.07	0.94 ± 0.08	0.69 ± 0.08	0.50 ± 0.07	0.35 ± 0.05
	IGC-Net	1.20 ± 0.17	1.39 ± 0.15	1.14 ± 0.13	0.73 ± 0.08	0.94 ± 0.74	0.73 ± 0.13
	CM (ERM)	$\mathbf{1.14 \pm 0.15}$	$\mathbf{1.20 \pm 0.07}$	$\mathbf{0.91 \pm 0.07}$	$\mathbf{0.66 \pm 0.06}$	$\mathbf{0.47 \pm 0.05}$	$\mathbf{0.33 \pm 0.04}$
	CM	$\mathbf{1.14 \pm 0.14}$	$\mathbf{1.20 \pm 0.07}$	$\mathbf{0.91 \pm 0.07}$	$\mathbf{0.66 \pm 0.06}$	0.48 ± 0.06	0.34 ± 0.05
$\gamma = 2$	RMSNs	1.39 ± 0.20	1.43 ± 0.11	1.09 ± 0.08	0.79 ± 0.07	0.55 ± 0.06	0.39 ± 0.05
	CRN	1.36 ± 0.18	1.38 ± 0.05	1.08 ± 0.04	0.81 ± 0.03	0.60 ± 0.05	0.44 ± 0.07
	G-Net	1.28 ± 0.11	1.56 ± 0.25	1.30 ± 0.25	1.04 ± 0.29	0.83 ± 0.28	0.72 ± 0.31
	CT	1.41 ± 0.15	1.47 ± 0.10	1.11 ± 0.10	0.79 ± 0.08	0.56 ± 0.06	0.39 ± 0.04
	IGC-Net	1.34 ± 0.22	1.55 ± 0.13	1.24 ± 0.09	0.81 ± 0.11	0.80 ± 0.18	1.07 ± 0.68
	CM (ERM)	$\mathbf{1.27 \pm 0.18}$	$\mathbf{1.35 \pm 0.06}$	$\mathbf{1.01 \pm 0.06}$	$\mathbf{0.73 \pm 0.05}$	$\mathbf{0.52 \pm 0.04}$	$\mathbf{0.37 \pm 0.03}$
	CM	$\mathbf{1.27 \pm 0.18}$	$\mathbf{1.35 \pm 0.06}$	$\mathbf{1.01 \pm 0.06}$	$\mathbf{0.73 \pm 0.06}$	$\mathbf{0.52 \pm 0.05}$	$\mathbf{0.37 \pm 0.04}$
$\gamma = 3$	RMSNs	$\mathbf{1.54 \pm 0.21}$	1.65 ± 0.15	1.24 ± 0.14	0.90 ± 0.13	0.65 ± 0.12	0.46 ± 0.10
	CRN	1.90 ± 0.17	1.66 ± 0.12	1.29 ± 0.11	0.96 ± 0.10	0.70 ± 0.10	0.51 ± 0.09
	G-Net	1.65 ± 0.20	1.79 ± 0.08	1.45 ± 0.17	1.11 ± 0.24	0.88 ± 0.29	0.71 ± 0.33
	CT	1.75 ± 0.21	1.75 ± 0.11	1.32 ± 0.09	0.93 ± 0.08	0.64 ± 0.08	0.46 ± 0.08
	IGC-Net	1.72 ± 0.25	1.85 ± 0.25	1.54 ± 0.26	1.34 ± 0.49	1.10 ± 0.33	0.71 ± 0.32
	CM (ERM)	1.57 ± 0.18	1.63 ± 0.05	1.22 ± 0.08	0.88 ± 0.09	$\mathbf{0.62 \pm 0.08}$	$\mathbf{0.45 \pm 0.07}$
	CM	1.57 ± 0.17	$\mathbf{1.60 \pm 0.05}$	$\mathbf{1.21 \pm 0.07}$	$\mathbf{0.87 \pm 0.08}$	$\mathbf{0.62 \pm 0.08}$	$\mathbf{0.45 \pm 0.07}$
$\gamma = 4$	RMSNs	1.85 ± 0.30	$\mathbf{1.79 \pm 0.05}$	$\mathbf{1.36 \pm 0.11}$	$\mathbf{1.00 \pm 0.15}$	$\mathbf{0.71 \pm 0.14}$	$\mathbf{0.50 \pm 0.11}$
	CRN	2.40 ± 0.24	1.96 ± 0.04	1.59 ± 0.12	1.36 ± 0.26	1.23 ± 0.41	1.15 ± 0.53
	G-Net	$\mathbf{1.79 \pm 0.28}$	1.89 ± 0.07	1.53 ± 0.13	1.18 ± 0.20	0.93 ± 0.24	0.77 ± 0.26
	CT	2.04 ± 0.18	1.86 ± 0.07	1.40 ± 0.04	$\mathbf{1.00 \pm 0.08}$	$\mathbf{0.71 \pm 0.08}$	$\mathbf{0.50 \pm 0.07}$
	IGC-Net	2.27 ± 0.32	1.91 ± 0.14	1.62 ± 0.15	1.20 ± 0.07	1.10 ± 0.37	1.05 ± 0.56
	CM (ERM)	2.00 ± 0.24	1.91 ± 0.06	1.44 ± 0.08	1.03 ± 0.09	0.73 ± 0.08	0.53 ± 0.08
	CM	2.00 ± 0.23	1.87 ± 0.06	1.41 ± 0.07	$\mathbf{1.00 \pm 0.08}$	$\mathbf{0.71 \pm 0.08}$	0.52 ± 0.07

K VISUALIZATION OF BALANCED REPRESENTATIONS BY T-SNE

The resulting visualizations are presented in Figure 5(a,b,c,d), obtained by applying t-SNE (Maaten & Hinton, 2008) to the representations constructed by CM and its ERM variant, CM (ERM). As described, the MTC module coordinates the prediction, adversarial, and reconstruction tasks by imposing distributional constraints on the representations. This is clearly demonstrated by comparing the t-SNE visualizations: the model with the balance module exhibits a more uniform and intertwined distribution of representations across treatment groups within a shared latent region. The MTC balancing module is central to achieving this distribution.



(a) CM, $\gamma = 0$

(b) CM (ERM), $\gamma = 0$

(c) CM, $\gamma = 4$

(d) CM (ERM), $\gamma = 4$

Figure 5: t-SNE embeddings of learned representations of CM and CM (ERM) with indicated current treatments. Each subfigure illustrates the complete trajectories of 100 samples, which are taken from the validation set of the fully-synthetic dataset TG simulator and span multiple levels of confounding ($\gamma = 0$, $\gamma = 4$). In the legend, None, Radiotherapy, Chemotherapy, Both correspond to treatments $(\mathbf{A}_t^c = 0, \mathbf{A}_t^r = 0), (\mathbf{A}_t^c = 0, \mathbf{A}_t^r = 1), (\mathbf{A}_t^c = 1, \mathbf{A}_t^r = 0)$, and $(\mathbf{A}_t^c = 1, \mathbf{A}_t^r = 1)$, respectively.

L RUNTIME AND MODEL SIZE COMPARISON

Tables 13 and 14 report runtime comparisons under identical hardware and software configurations. CM achieves the lowest mean total runtime across all three datasets, requiring 1.3, 4.0, and 10.3 minutes on the fully-synthetic, semi-synthetic, and real-world datasets, respectively. Notably, on real-world MIMIC-III, its lower inference cost offsets the higher training time relative to G-Net, yielding a slightly lower total runtime (10.3 vs. 10.6 minutes). CM also achieves the lowest inference times and ITPS on both the real-world and semi-synthetic MIMIC-III datasets.

Furthermore, Table 15 reports model sizes after hyperparameter tuning. CM has the fewest trainable parameters on semi-synthetic and real-world MIMIC-III, with 27.1K and 36.4K parameters, respectively. On the fully-synthetic dataset, CM uses 3.0K parameters at every confounding level and is smaller than all baselines except IGC-Net at $\gamma = 0$, which uses 2.4K. These results show a compact parameter footprint, with runtime advantages depending on the dataset and metric.

Table 13: Runtime of experiments on the semi-synthetic MIMIC-III dataset. Experiments are carried out on $1\times$ NVIDIA A10 Tensor Core GPU, where ITPS refers to inference time per sample. Unless otherwise specified, the unit is minutes.

	Main stages of Training & Inference	Training Time	Inference Time	Total Runtime	ITPS (in sec)
RMSNs	2 networks for IPTW & encoder & decoder	16.0 ± 5.8	$\mathbf{0.1 \pm 0.0}$	16.1 ± 5.8	**0.02**
CRN	encoder & decoder	15.3 ± 5.2	0.2 ± 0.0	15.5 ± 5.3	0.07
G-Net	single network & MC sampling for inference	4.5 ± 1.3	6.1 ± 0.5	10.6 ± 1.7	1.84
CT	single Transformer multi-input block	8.6 ± 0.1	0.3 ± 0.0	8.9 ± 0.1	0.08
IGC-Net	regression-based Transformer multi-input block	254.7 ± 3.8	1.6 ± 0.1	256.3 ± 3.8	0.49
CM	GCSI-Mamba block	$\mathbf{3.9 \pm 0.1}$	$\mathbf{0.1 \pm 0.0}$	$\mathbf{4.0 \pm 0.1}$	**0.02**

Table 14: Runtime of experiments on the real-world MIMIC-III dataset.

	Main stages of Training & Inference	Training Time	Inference Time	Total Runtime	ITPS (in sec)
RMSNs	2 networks for IPTW & encoder & decoder	73.9 ± 4.1	$\mathbf{0.2 \pm 0.0}$	74.1 ± 4.1	**0.06**
CRN	encoder & decoder	78.9 ± 1.3	0.3 ± 0.0	79.2 ± 1.3	0.08
G-Net	single network & MC sampling for inference	$\mathbf{6.6 \pm 0.0}$	4.0 ± 0.2	10.6 ± 0.2	1.19
CT	single Transformer multi-input block	167.4 ± 1.8	2.5 ± 0.2	169.9 ± 1.9	0.74
IGC-Net	regression-based Transformer multi-input block	4764.4 ± 92.6	26.7 ± 1.1	4791.1 ± 92.1	8.02
CM	GCSI-Mamba block	10.1 ± 1.2	$\mathbf{0.2 \pm 0.0}$	$\mathbf{10.3 \pm 1.2}$	0.07

Table 15: Total number of trainable parameters of models after hyperparameter tuning. Here, we distinguish (1) data from fully-synthetic (FS) dataset based on benchmark TG simulator; (2) data from semi-synthetic (SS) dataset based on real-world benchmark MIMIC-III dataset; (3) data from real-world (RW) based on benchmark MIMIC-III dataset. It should be noted that the reported number is the sum of trainable parameters among all the subnetworks for RMSNs and CRN.

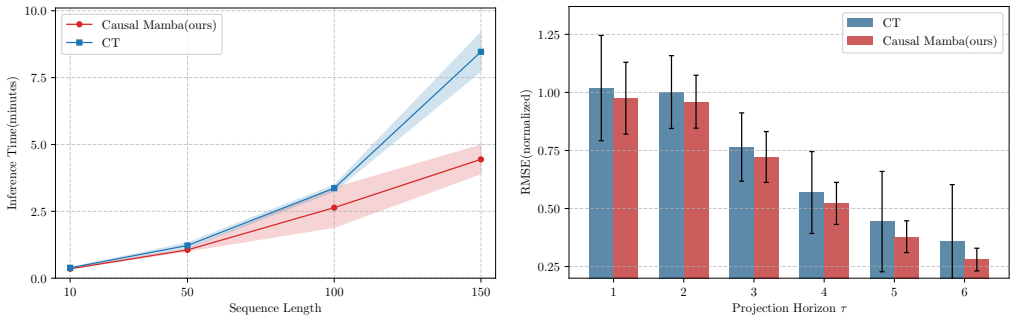
	FS data					SS data	RW data
	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$		
RMSNs	20.8K	27.5K	7.3K	69.0K	30.5K	890.1K	1532.6K
CRN	8.5K	19.9K	22.5K	13.7K	19.4K	204.7K	361.3K
G-Net	3.8K	3.8K	3.8K	6.2K	3.6K	206.8K	310.2K
CT	17.6K	5.6K	5.7K	5.9K	5.6K	58.8K	67.5K
IGC-Net	2.4K	17.1K	17.1K	5.2K	5.7K	132.0K	65.8K
CM	3.0K	3.0K	3.0K	3.0K	3.0K	27.1K	36.4K

M PERFORMANCE COMPARISON ON LONG SEQUENCES

To evaluate CM for counterfactual prediction over long sequences, we compare it with CT on the TG simulator dataset. Specifically, we vary the historical trajectory lengths (i.e., 10, 50, 100, 150) and adopt two key metrics: inference runtime and prediction accuracy.

Figure 6(a) shows that the inference time of CT exhibits a quadratic growth trend with increasing sequence length, whereas the runtime of CM increases approximately linearly. When the sequence length reaches 150, CM requires approximately 4.1 minutes, which is substantially shorter than the 8.4 minutes of CT, representing a reduction of 51%. Regarding prediction accuracy, Figure 6(b) demonstrates that for sequences of length 150, CM outperforms CT across all projection horizons, achieving both lower prediction error and smaller variance.

Collectively, these findings show favorable efficiency and accuracy for CM in the evaluated long-sequence counterfactual outcome setting. Its observed inference scaling and stable error performance make CM suitable for the evaluated setting, where both computational tractability and reliable long-range predictions are essential. Moreover, the integrated architecture avoids multi-stage training.



(a) Inference time with varying sequence length

(b) nRMSE of τ-step-ahead prediction with sequence length 150

Figure 6: Efficiency and performance comparison between CM and CT on the fully-synthetic TG simulator dataset under different sequence lengths. Here, the time-varying confounding level is $\gamma = 4$.