A Recurrent Network Approach to G-Computation for Sepsis Outcome Prediction Under Dynamic Treatment Regimes

by

Stephanie Hu

B.S. Computer Science and Engineering, Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author .....
Department of Electrical Engineering and Computer Science
August 13, 2021

Certified by .....
Roger G. Mark
Distinguished Professor
Thesis Supervisor

Certified by .....
Li-wei H. Lehman
Research Scientist
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

A Recurrent Network Approach to G-Computation for Sepsis Outcome Prediction Under Dynamic Treatment Regimes

by

Stephanie Hu

Submitted to the Department of Electrical Engineering and Computer Science
on August 13, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Sepsis is a life-threatening condition that occurs when the body's normal response to an infection is out of balance. A key part of managing sepsis involves the administration of intravenous fluids and vasopressors, but prescribing the correct balance of interventions is challenging since both under- and over-resuscitation can lead to adverse outcomes. While many retrospective studies have attempted to understand the relationship between sepsis treatment, fluid overload, mortality, and other outcomes, most are correlation-based and cannot actually estimate the causal effects of intervention. Prospective randomized clinical trials allow researchers to test the effects of alternative therapies more directly, but these types of studies tend to span multiple years and recent results regarding optimal regimes have been conflicting.

In this thesis, we use methods from causal inference to predict outcomes in sepsis patients under different fluid and vasopressor strategies. Specifically, we explore a recurrent neural network approach to g-computation, a technique that allows us to estimate effects under treatments that are dynamic and time-varying. Our work builds on a previous sequential deep learning implementation known as G-Net. We evaluate G-Net using synthetic physiological data and show that it outperforms traditional linear regression models in predicting patient trajectories under alternative interventions. We then adapt and apply the improved architecture for analyzing outcomes under counterfactual treatment strategies in a real-world cohort of sepsis patients, using observational data collected from the intensive care unit. Our results demonstrate that G-Net is able to generate reasonable counterfactual estimates under alternative regimes.

Thesis Supervisor: Roger G. Mark
Title: Distinguished Professor

Thesis Supervisor: Li-wei H. Lehman
Title: Research Scientist

Acknowledgments

First and foremost, I would like to thank my direct thesis supervisor, Li-wei Lehman, for an enriching MEng experience. She was incredibly helpful in guiding the direction of the project, and I am both impressed and deeply appreciative of the time and passion she dedicated as an advisor. Working under her for the past two years has taught me how to better think about difficult problems and ask interesting, meaningful questions, and I feel that I have grown significantly as a researcher as a result.

I would also like to thank my supervisor Professor Roger Mark for allowing me the opportunity participate in the Laboratory for Computational Physiology and for making me feel welcome as a student. I am grateful for his clinical expertise and feedback, and for really pushing me to consider the "Why?" aspects of research. He has further inspired me to pursue challenges at the intersection of medicine and technology.

To my mentor, Zach Shahn, I would like to say thank you for helping me better understand the causal inference side of the project and for his patience during the countless project discussions we had during the year. I did not have much experience prior to starting my MEng but have learned a tremendous amount since then. I would also like to express appreciation to the IBM team for providing expertise on machine learning and modeling.

Next, I would like to extend gratitude to Dr. Elias Baedorf-Kassis for taking time to address various clinically-related questions about the project, as well as Alistair Johnson for answering my numerous queries about MIMIC. Also thanks to Yuria Utsumi for orienting me to the existing G-Net repository and Mingyu Lu for assisting with questions on CVSim. I feel fortunate to have had the opportunity to work with and learn from so many different people throughout my MEng journey.

Last but not least, I am grateful to my friends for making the last few years as amazing and memorable as they were, and to my parents and sister for their unconditional love and support. I would not be where I am today without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sepsis occurs when the body undergoes an extreme response to infection [23]. The mechanisms normally used to fight off the infection trigger a chain of adverse events in the body, leading to severe multiple organ damage and possibly even death. The number of sepsis cases in the United States is at least 1.7 million per year, with a mortality rate of 15% to 30% [5, 25]. Sepsis accounts for one-third of all patients who die in a hospital setting [5].

A key part to managing septic patients is restoring tissue perfusion through the administration of intravenous fluids and/or vasopressors [7]. Prescribing the optimal balance of fluids and vasopressors remains challenging, since both under- and over-resuscitation can lead to adverse outcomes. Under-resuscitation from fluids may insufficiently treat the septic shock condition, ultimately resulting in multi-organ failure, while over-resuscitation from fluids may harm the cardio-respiratory systems and cause pulmonary and peripheral edema. Vasopressors are typically administered when patients' blood pressure fails to respond to fluids, but they are known to have harmful effects on patients and can induce arrhythmias if overdosed [16].

Analyzing the effects of fluid treatment strategies in sepsis patients has been a recent area of interest for researchers and clinicians aiming to improve outcomes for the condition. Although there have been many retrospective studies looking at the relationship between fluid interventions, fluid overload, and mortality, most of these use correlational approaches like regression analyses which do not adjust for

time-varying confounders and cannot estimate the causal effects of fluid therapy on sepsis prognosis [8, 17]. While prospective studies can better tease apart causal relationships, they often take many years to execute and are very costly in terms of time and resources [1, 27, 30]. Consequently, conflicting results from various clinical trials across the last two decades has hindered the adoption of standardized protocol-based intervention guidelines for sepsis, resulting in substantial variations in treatment decisions that depend on the treating clinician.

In our study, we developed a deep sequential modeling approach to g-computation for estimating the effects of fluid and vasopressor strategies among sepsis patients admitted to the intensive care unit (ICU). G-computation is a causal inference method for estimating expected counterfactual outcomes under dynamic, time-varying treatment strategies. While previous research using g-computation employed linear regression models in their implementations, we propose using recurrent neural networks to better capture the complex temporal dependencies between covariates in the patient history.

Our work builds on prior work by Li et al. [18] to further develop G-Net, an RNN-based approach to g-computation, by experimenting with different architectural designs for the model, investigating modeling issues encountered in real-world (as opposed to synthetic) data, and applying the model to a practical clinical setting. To evaluate G-Net's performance using different architectural designs, we conducted experiments on simulated data from CVSim, a program that simulates the human cardiovascular system [10]. Finally, we applied G-Net to predicting outcomes of sepsis patients in the ICU under alternative fluid resuscitation treatment regimes using real-world observational data from the MIMIC database [15]. The alternative treatment regimes were defined based on guidelines from CLOVERS and ProCESS, two well-known randomized clinical trials studying fluid resuscitation strategies in sepsis patients [1, 30].

The rest of this chapter focuses on setting up the problem of counterfactual prediction and describing how g-computation can be used to carry out this task. We conclude by outlining the contributions of our work and the organization of this thesis.

1.1 G-Computation for Counterfactual Prediction

In the real world, we can only observe one set of outcomes (i.e. those that actually occurred under the observational regime), but often we might wonder what might have happened had a different course of action been followed. This is particularly important for clinicians who may have to choose between multiple treatment options for their patients but do not have the luxury of testing all strategies before making a decision. Sepsis patients, who often display heterogeneous responses to the same therapies, may particularly benefit from clinicians being able to predict the effects of these alternative intervention strategies. The task described can be formally classified as *counterfactual prediction*, in which the goal is to estimate the trajectories of potential outcomes under different interventions given previous observed covariate history [11, 28].

Prediction of outcomes as a function of therapy is difficult due to the fact that the outcomes depend on complex interactions among multiple time-varying, dynamic treatments and evolving patient covariate history. *Time-varying* describes treatments that comprise decisions at multiple time points while *dynamic* indicates that the intervention at each time point is dependent on the history up to that time point. While administering large volumes of fluids to septic patients may be required to increase their blood pressure and promote blood perfusion through their organs, such strategies can also lead to fluid overload which result in their own set of adverse effects [7, 16]. The regime that clinicians adopt for a given septic patient will likely involve decisions at multiple timesteps on how much fluid to administer (*time-varying*), and the volume ultimately administered will strongly depend on the patient's observed history up until the present (*dynamic*). As such, any method used to address problems involving counterfactual prediction must be able to account for underlying relationships between intervention regimes and other variables of interest.

Our study focuses on using g-computation to predict the effects of alternative treatment strategies on outcomes in sepsis patients. Given the data under the observational regime, g-computation first learns the distribution of covariates at each timestep conditioned on the history of covariates and treatments up until that timestep, and

then estimates the counterfactual outcomes by simulating the data forward in time under different treatment strategies [11].

Typically, the conditional distribution of covariates on observed history is estimated via regression models. While any model could theoretically be used as input to g-computation, most prior work has focused on generalized linear regression models (GLMs), which are unable to capture temporal dependencies present in the data. A recent study by Li et al. proposed a recurrent neural network (RNN) implementation of G-computation (G-Net) and evaluated its performance using synthetic patient data generated by CVSim, a well-established mechanistic model of the cardiovascular system [18]. They also applied G-Net to predict diuretics onset as a fluid overload indicator on real-world ICU data, but their work was limited due to the fact that diuretics can be prescribed for a number of reasons (e.g. a patient on mechanical ventilation) that are not directly related to fluid overload; this resulted in the model outputs being difficult to interpret. In addition, their implementation did not account for the fact that real-world ICU data can be sparse and irregularly sampled, a challenge frequently encountered in the development of deep sequential models for clinical applications [20].

1.2 Contribution

In our study, we investigated various architectural designs and modeling techniques for G-Net, a deep sequential modeling approach based on g-computation for predicting outcomes under dynamic, time-varying treatment regimes. We tested and validated G-Net's performance under different architectural designs using CVSim and applied G-Net to estimating effects of fluids and vasopressor strategies among sepsis patients in the ICU. Evaluation on synthetic data (CVSim) was important to assess the performance of G-Net in predicting outcomes of alternative regimes, as real-world data only contains information about outcomes under the observational regime. In other words, there is no "ground-truth" counterfactual dataset that can be used for testing, and we must rely on clinical expertise to evaluate the predictions made by the model

under the different interventions of interest. The use of synthetic data can provide more rigorous evidence on the performance of G-Net in the task of counterfactual prediction. The contributions of this thesis are outlined as follows:

1. **Improvements to G-Net for counterfactual prediction on synthetic data.** We extended the framework introduced by Li et al. [18], which is capable of capturing complex temporal dependencies among multiple time-varying variables. To test the model, we used synthetic data generated by CVSim. A component of this work involved refining the process of data generation to better align with real-world physiological observations, which subsequently aided in improving the training and prediction performance of G-Net.

2. **Adaptation of G-Net for clinical applications.** We explored various modifications to adapt G-Net for counterfactual estimation on real-world patient data, such as testing different architectures and experimenting with methods for modeling missingness. The clinical dataset we focused on was a retrospective cohort of sepsis patients from the ICU, which we defined and compiled as part of our work here.

3. **Analysis of counterfactual treatment strategies for sepsis patients.** We applied our RNN-based g-computation approach to predict adverse outcomes (e.g. fluid overload and mortality) as a function of fluid management strategies in sepsis patients. Unlike previous studies on sepsis interventions that focused on correlational relationships between fluid intervention strategies and sepsis prognosis, this work used a causal inference approach that adjusts for time-varying confounders in order to provide insight into the causal role of different treatment regimes on outcomes in sepsis patients.

1.3 Thesis Organization

The next chapter provides further background on sepsis and its treatment, in addition to describing previous studies using g-computation and deep neural networks for

counterfactual prediction. Chapter 3 formalizes the counterfactual prediction problem and outlines the g-computation framework in greater depth. This helps set up Chapter 4, which discusses the implementation, training, and simulation of G-Net.

We first tested and validated G-Net on a synthetic dataset generated by CVSim; the data generation process and subsequent G-Net experiments using CVSim data are addressed in Chapters 5 and 6, respectively. In Chapter 7, we provide details on the cohort design of sepsis patients used in our study, while in Chapter 8, we report the results of performing predictive check analyses with G-Net on this cohort. In Chapter 9, we present the outcome predictions made by G-Net under various counterfactual regimes. Finally, we conclude with a summary of our results and future research directions in Chapter 10.

Chapter 2

Related Work

In this chapter, we provide additional background on sepsis treatment research that helps better motivate our clinical problem of interest. We also discuss previous studies employing g-computation for clinical applications and consider recent progress in developing deep learning models for counterfactual prediction. Finally, we introduce previous work on the G-Net framework in order to set up the foundation for our contributions.

2.1 Improving Sepsis Treatment Strategies

There have been multiple clinical trials that have attempted to study optimal treatment strategies for sepsis, with the goal of developing a set of improved intervention guidelines. For example, in 2001, Rivers et al. reported groundbreaking results on the effectiveness of early-goal directed therapy (EGDT) from a randomized trial on patients with severe sepsis or septic shock, which subsequently led to clinicians pursuing more liberal fluid administration strategies for treating septic patients [27].

A follow-up randomized study on protocol-based care for early septic shock (ProCESS), conducted more than a decade later, aimed to investigate whether the EGDT findings were generalizable and how the EGDT protocol compared against (1) protocol-based standard therapy that did not require the placement of a central venous catheter, administration of inotropes, or blood transfusions and (2) the usual care [1]. The re-

sults suggested that protocol-based treatments did not necessarily improve outcomes in sepsis patients diagnosed in emergency departments. This is consistent with other randomized studies following Rivers et al. that have shown no benefits of administering large volumes of fluid over standard treatment practices and is supported by a growing body of observational literature [31].

More recently, clinical trials have increasingly focused on investigating strategies that limit fluid administration in the early intervention of sepsis. The CLOVERS study, which began in 2018 and was targeted for recent completion in June 2021, tested two treatment regimes, early use of vasopressors (and thus restricted use of fluids) or early use of liberal fluids, in affecting downstream outcomes in patients presenting in the ICU with sepsis-induced hypotension [30]. The findings of this study have yet to become available, as it frequently takes several years for trials examining fluid resuscitation strategies to be executed and validated.

Because of the lengthy timeline required to carry out prospective clinical trials, Shahn et al. proposed using causal inference techniques to gain some insight into sepsis treatment decisions in the meantime [31]. They performed a retrospective cohort study of ICU patients with sepsis to estimate 30-day mortality outcomes resulting from administering different volumes of fluid during the first 24 hours of ICU care. They used a dynamic marginal structural model (MSM), adjusting for confounding between treatment strategy and patient characteristics, and found that there was a beneficial effect of fluid resuscitation caps on 30-day mortality.

While their study demonstrated the potential for causal inference in analyzing treatment outcomes and informing the findings of past and future clinical trials, the utility of their results is limited due to the restricted ability of MSMs in making predictions for dynamic, time-varying intervention strategies. The model they used assumes that the optimal dynamic treatment regime would be chosen among a moderate set of enforceable regimes, which were selected based on only a subset of past covariate history [31]. A method known as g-computation, on the other hand, can make predictions using the entire covariate and treatment history, which provides a richer set of information to inform those predictions.

2.2 Applications of the Parametric G-Formula

G-computation is an approach first proposed by Robins et al. for estimating time-varying treatment effects [28], though for many years its applications were limited by the lack of sufficient computational software and of rich clinical time-series data. In one of the first major studies applying the parametric g-formula to an epidemiological problem, Taubman et al. aimed to predict the population risk of coronary heart disease under a number of hypothetical lifestyle interventions, including no smoking, exercising at least 30 minutes a day, maintaining a healthy diet, consuming at least 5 grams of alcohol a day, maintaining a lower body mass index, a combination of the previous 5 interventions, and no intervention [33]. This was a retrospective cohort study using data from the Nurses' Health Study collected between 1982 and 2002, and the results illustrated the potential for the g-formula to be used in estimating the effects of counterfactual treatment regimes that are time-varying and dynamic.

The g-computation algorithm, described further in Chapter 3, requires arbitrary regression models to learn observational covariate distributions conditioned on past history. While previous studies, including Taubman et al., have mostly relied on GLMs for this task, there is no conceptual barrier to substituting these models with more complex ones.

2.3 Deep Learning for Counterfactual Prediction

Due to their ability to handle complex time dependencies between variables, there has been increased interest in applying RNNs to the problem of estimating time-varying treatment effects. Bica et al. introduced the Counterfactual Recurrent Network (CRN), which applies domain adversarial training to build treatment-invariant representations of patient history for estimating effects of counterfactual intervention strategies over time [4]. Prior to that, Lim et al. explored the use of RNNs in implementing marginal structural models (so-called recurrent MSMs) for counterfactual predictions and demonstrated that their model outperformed linear baselines and tra-

ditional MSM approaches [19]. Recent research by Atan et al., Alaa et al., and Yoon et al. have also looked at predicting outcomes under counterfactual strategies given data under the observational regime [2, 3, 37].

Although these techniques show promise in estimating treatment effects, they are restricted to either learning point exposures and/or making outcome predictions for time-varying treatment strategies that are static, whereas we are interested in predictions for *dynamic* regimes as well. An example of a static treatment strategy might be "give 1 liter fluid each hour for the next 3 hours" as it does not depend on recent covariate history. Contrast this with a dynamic strategy, which might say "for each hour in the next 3 hours, if blood pressure is less than 65, give 1 liter fluids, otherwise give 0 liters." Methods like history-adjusted MSMs would only be able to estimate outcomes of interest under the first strategy, but not the second, while g-computation could handle both. G-computation relies on different modeling assumptions than MSMs, and can handle high dimensional health history in particular. It is also able to estimate the distribution of counterfactual outcomes under a time-varying treatment strategy, which is not as straightforward to do with MSMs.

In a previous study by Schulam et al., researchers used Gaussian processes (GPs) to implement a continuous time version of g-computation; however, they only considered static time-varying strategies when developing their model [29]. Xu et al. also explored GPs to predict individual patient-level treatment response curves, but their evaluation was limited to predictive checks on a held-out test set derived from real-world ICU data; they did not examine the performance of their model on counterfactual predictions [36]. Another limitation of both studies is the use of GPs, which are intractable for large datasets and have high time complexity as the number of variables increases. RNNs, and especially long short-term memory networks (LSTMs), on the other hand, are more scalable and better able to handle higher-dimensional data; indeed, they have been shown to achieve start-of-the-art performance on a variety of time series regression tasks [6, 34, 35]. But despite these successes, to the best of our knowledge, there have not been any "deep" implementations of g-computation attempted yet.

2.4 Background on G-Net

Previous research by Li et al. developed a flexible sequential deep learning framework for g-computation to estimate the conditional distribution of covariates given patient history at each time point and to simulate the covariates under various treatment strategies [18]. Simulating the joint distribution is difficult when covariates possess different distribution types (e.g. categorical versus continuous), so the covariates were separated into groups or types and sequentially simulated, group by group.

In their study, Li et al. used two covariate groups, one for categorical variables and one for continuous variables, and explored four different neural network architectures involving different combinations of linear and LSTM layers for computing representations of patient history and modeling the distributions of the covariate groups [18]. The authors found that the sequential deep learning models, i.e. employing LSTMs to model conditional covariate distributions, demonstrated improved performance over the linear regression baseline on synthetic patient data. Our work builds on this initial implementation of G-Net by refining the model architecture and adapting it for real-world use cases.

Chapter 3

G-Computation Overview and Problem Setup

Counterfactual prediction is useful in scenarios that necessitate individuals to make decisions under uncertainty by allowing them to estimate the effects of the various possible courses of action. Depending on the scope of the problem, there are many methods that could potentially be employed for the task of counterfactual prediction. When the treatment strategies being studied are time-varying, we can employ a class of generalized methods known as "g-methods" to estimate their effects [24]. G-methods allow us to obtain consistent estimates of the effects of different treatment plans and the ratio of their outcomes, and include marginal structural models (MSMs), structural nested models, and g-computation [24].

In our study, we aimed to estimate outcomes under dynamic, time-varying counterfactual treatment strategies given observed patient histories with high-dimensional histories. Of the g-methods discussed, g-computation is particularly well-suited to this task. The framework is carried out by (1) learning the conditional distributions of covariates based on past covariate values and treatment actions, and (2) estimating counterfactual outcomes by simulating from these distributions forward in time via Monte Carlo methods. In this chapter, we review the g-computation procedure in more depth and discuss how it can be applied to our problem of interest.

3.1 Problem Setup

We are interested in measuring the effect of different treatment strategies g on a set of outcomes, where both the treatment and outcomes may be influenced by a set of covariates. The strategies of interest are dynamic and time-varying, which means the treatment administered at timestep t depends on the patient covariate history and treatment actions taken up until this point. An example of such a strategy is "at each timestep t, give 500mL of fluid if mean arterial blood pressure is less than 65mmHg and if the patient has not developed pulmonary edema; otherwise administer vasopressors." G-computation can be used to estimate effects of such time-varying and dynamic treatment strategies. In contrast, other methods such as MSMs, can only make predictions for static, time-varying regimes. (See Section 2.3 for examples.)

Using g-computation, we can test a number of different treatment strategies of interest by simulating the effects of those strategies on covariates and outcomes. From the simulations, we can derive insights on population-level treatment effects and compare results across counterfactual regimes. Let us define:

- $t \in \{0, \ldots, K\}$ to be discrete-valued time, with K being the end of followup
- A_t to be the observed treatment action at time t
- Y_t to be the observed value of the outcome(s) at time t
- L_t to be a vector of d covariates at time t that may influence treatment decisions or be associated with the outcome
- $\bar{X}_t$ to be the history $X_0, \ldots, X_t$ and $\underline{X}_t$ to be the future $X_t, \ldots, X_K$ for arbitrary time varying variable X (for example, $\bar{L}_t$ would denote patient covariate history up to and including timestep t)

A dynamic, time-varying treatment strategy g can be written as a collection of functions $g = \{g_0, \ldots, g_K\}$, such that g_t maps patient history onto a treatment action $g_t(\bar{L}_t, \bar{A}_{t-1})$ at time t, where $(\bar{L}_t, \bar{A}_{t-1})$ is the patient history preceding intervention at time t. That is to say, treatment at timestep t depends on the covariates up to

and including timestep t and the sequence of treatment actions preceding timestep t (i.e. up to and including timestep $t-1$).

Given a strategy g, we denote $Y_t(g)$ to be the patient outcomes observed at time t as a result of following g from baseline [28]. For a patient who has had $m-1$ timesteps of observed treatment $\bar{A}_{m-1}$, for whom we would like to predict the effects of a different strategy g administered from timestep m onward, we can denote the counterfactual outcome at time t to be $Y_t(\bar{A}_{m-1}, \underline{g}_m)$, where $t \geq m$.

The goal of counterfactual prediction is to estimate the expected counterfactual patient outcomes

$$\{E[Y_t(\bar{A}_{m-1}, \underline{g}_m)|\bar{L}_m, \bar{A}_{m-1}], t \geq m\} \tag{3.1}$$

given observed patient history through time m, observed treatment history through time $m-1$, and some specified treatment strategy g. In addition, it is also possible to estimate the counterfactual outcome distributions at future time points

$$\{p(Y_t(\bar{A}_{m-1}, \underline{g}_m)|\bar{L}_m, \bar{A}_{m-1}), t \geq m\} \tag{3.2}$$

for $t \geq m$. It is helpful to note that if we do not condition on $(\bar{L}_m, \bar{A}_{m-1})$ in either Equation 3.1 or 3.2, we obtain the expectation and distribution, respectively, over the full population.

3.2 The G-Computation Framework

Provided the setup described above, we can now introduce a formal explanation of the g-formula and the assumptions underlying g-computation.

3.2.1 The G-Formula

To estimate Equations 3.1 and 3.2 through g-computation, the following assumptions must hold [28]:

- **Consistency:** the counterfactual outcome is the same as the observed outcome when the counterfactual regime is the observed regime; that is, $Y_t(\bar{A}_t) = Y_t$ for $t \in \{0, \ldots, K\}$

- **Sequential Exchangeability:** there is no unobserved confounding of treatment at any time, so that all drivers of treatment are observed at every hour

- **Positivity:** the counterfactual strategy of interest has some non-zero probability of being followed; this is important to avoid conditioning on events with zero probability

Under these assumptions, we can rewrite Equation 3.2 as

$$p(Y_m(\bar{A}_{m-1}, g_m)|\bar{L}_m, \bar{A}_{m-1}) = p(Y_m|\bar{L}_m, \bar{A}_{m-1}, A_m = g_m(\bar{L}_m, \bar{A}_{m-1})), \tag{3.3}$$

for $t = m$. In other words, the conditional distribution of the counterfactual outcome is simply the conditional distribution of the observed outcome given patient history and the treatment strategy of interest.

The equation above becomes more complicated for $t > m$, as it is necessary to adjust for time-varying confounding. If we use $X_{i:j}$ to represent $X_i, \ldots, X_j$ for any arbitrary variable X, under Assumptions 1-3, the g-formula yields

$$\begin{aligned} &p(Y_t(\bar{A}_{m-1}, \underline{g}_m) = y|\bar{L}_m, \bar{A}_{m-1}) \qquad (3.4)\\ &= \int_{l_{m+1:t}} p(Y_t = y|\bar{L}_m, \bar{A}_{m-1}, L_{m+1:t} = l_{m+1:t}, A_{m:t} = g(\bar{L}_m, \bar{A}_{m-1}, l_{m+1:t})) \\ &\times \prod_{j=m+1}^{t} p(L_j = l_j|\bar{L}_m, \bar{A}_{m-1}, L_{m+1:j-1} = l_{m+1:j-1}, A_{m:j-1} = g(\bar{L}_m, \bar{A}_{m-1}, l_{m+1:j-1})) \end{aligned}$$

This captures the fact that outcomes at time $t > m$ depend on (1) observed patient covariate history up to and including timestep m, (2) treatment actions under the observational regime up to and including timestep $m-1$, (3) treatment actions under the counterfactual regime starting from timestep m, and (4) the effects of the new regime on covariates, which would manifest starting from timestep $m+1$.

3.2.2 Steps of G-Computation

In practice, it is difficult to solve for the integral in Equation 3.4 in its closed form, but Monte Carlo simulations can be used to approximate it. That is, we can simulate a population of patients under the counterfactual treatment regime and use the empirical outcome distribution as an estimation of the actual outcome distribution.

To construct a simulated trajectory for a single patient under some intervention strategy of interest, we sample from the joint distribution $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$ at each timestep $t \in \{m, \ldots, K\}$. We then repeat this process M times to form a simulated population, where the empirical distribution of the outcomes at each time t allows us to approximate Equation 3.2. The sample averages of the draws at each time t are an estimate of the conditional expectations in Equation 3.1 that can serve as point predictions for $Y_t(\bar{A}_{m-1}, \underline{g}_m)$.

The procedure described above is provided as pseudocode in Algorithm 1, starting at timestep m. Without loss of generality, the outcome Y_t is assigned to be a variable in the covariate vector L_t for simplicity. Note that when $t = m + 1$, line 4 of the algorithm simply simulates l^*_{m+1} from the distribution $p(L_t|\bar{L}_m, \bar{A}_{m-1}, A_m = a^*_m)$.

Algorithm 1: G-Computation

1 **for** $n \leftarrow 1$ to M
2 Set $a^*_m = g_m(\bar{L}_m, \bar{A}_{m-1})$
3 **for** $t \leftarrow m + 1$ to K
4 Simulate l^*_t from $p(L_t|\bar{L}_m, \bar{A}_{m-1}, L_{m+1:t-1} = l^*_{m+1:t-1}, A_{m:t-1} = a^*_{m:t-1})$
5 Set $a^*_t = g_m(\bar{L}_m, \bar{A}_{m-1}, l^*_{m+1:t}, a^*_{m:t-1})$

Chapter 4

G-Net: A Recurrent Network Approach to G-Computation

The g-computation algorithm requires us to first learn the conditional distributions $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$ for each covariate given patient history at timestep t. Once these distributions have been learned, we can apply the procedure described in Algorithm 1 of Section 3.2.2 to simulate the covariates under different intervention strategies. Traditional methods employ GLMs to carry out g-computation, but here we propose an RNN approach for this task. RNNs have been shown to perform well in situations involving high-dimensional, multivariate, time-varying data and thus were our model of choice for implementing the G-Net framework. To perform g-computation, G-Net operates in two modes: (1) in training mode, we fit the network in a manner that enables us to simulate from $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$, then (2) in simulation mode, we use the trained model to sample covariates at each timestep following Algorithm 1.

4.1 Estimating the Conditional Distributions of Covariates

For a vector L_t, we can separate the individual covariates into p (potentially multivariate) disjoint groups, where the number of covariates in each group need not be

the same. We denote the p components of L_t by $L_t^0, \ldots, L_t^{p-1}$, where p ranges from 1 through the number of covariates d. By setting $p = 1$, we can model all covariates simultaneously and directly approximate $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$. On the other hand, by setting $p > 1$, we can model groups of covariates separately. This may be desirable if the distribution of the covariates are of different types, rendering it difficult to sample from their joint distribution such as in the case of mixing continuous and categorical variables. Learning a separate model for each type of distribution, then, could theoretically help improve performance. In fact, we could even set p to be d and train a custom model for every variable. Notationally, we will refer to these covariate-specific models as *boxes* (Figure 4-1).

When $p > 1$, we can impose an arbitrary ordering on the covariate groups $L_t^0, \ldots, L_t^{p-1}$ and estimate the conditional distributions of each covariate group L_t^j given all variables preceding it in this ordering. In other words, we can learn p conditional distributions of the form $p(L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1})$. Once we obtain these distributions, we can simulate from $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$ by exploiting the basic probability identity:

$$\begin{aligned} p(L_t|\bar{L}_{t-1},\bar{A}_{t-1}) = p(L_t^0|\bar{L}_{t-1}, \bar{A}_{t-1}) \times p(L_t^1|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0) \\ \times \cdots \times p(L_t^{p-1}|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{p-2}) \end{aligned} \tag{4.1}$$

In other words, we can simulate from $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$ by simulating each L_t^j from $p(L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1})$ in the order that was imposed during training.

4.2 Network Architecture

When designing G-Net, we aimed to create a flexible architecture for carrying out g-computation that can be conveniently configured depending on the problem at hand. At each timestep, G-Net is trained to predict L_t, the covariates at timestep t, given $(\bar{L}_{t-1}, \bar{A}_{t-1})$, the covariates and treatment action at timestep $t-1$.

The simplest implementation involves a single box $f^0(\cdot; \Lambda_0)$ with learnable parameters Λ_0 that takes as input all covariates and treatment actions up to timestep

$t-1$ and uses them to predict the covariates at timestep t. That is to say, the input at each timestep is $(\bar{L}_{t-1}, \bar{A}_{t-1})$ and the output is L_t. For values of p greater than 1, we use p boxes to model each of the p covariate groups. The input to the first box $f^0(\cdot;\Lambda_0)$ is simply $(\bar{L}_{t-1}, \bar{A}_{t-1})$, while the output is $L_t^0 = f^0(\bar{L}_{t-1}, \bar{A}_{t-1};\Lambda_0)$, the covariates in the first group at timestep t. This output is then concatenated with L_{t-1} to form the vector $(\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0)$, which serves as the input to the next box $f^1(\cdot;\Lambda_1)$. In general, the input to the jth box $f^j(\cdot;\Lambda_j)$ is $(\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1})$ while the output is $L_t^j = f^j(\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1};\Lambda_j)$. In the CVSim experiments (Chapter 6), we used $p = 2$ while in the the MIMIC experiments (Chapter 8), we set p to the number of covariates. We refer to this latter design as *one variable per box*, which captures the fact that each covariate distribution is being learned by a separate model. Specific implementation details are provided in Sections 6.1.2 and 8.1.1.

With this general schematic in mind, there are many possible implementations of G-Net we can consider. Of particular interest in our study was varying the type of model used for the p boxes: specifically, we focused on comparing the performance of linear versus LSTM layers. Another extension we explored was learning a representation of the covariates $R_t = r(\bar{L}_{t-1}, \bar{A}_{t-1};\Theta)$ with learnable parameters Θ that could be fed into the subsequent covariate boxes; the input to $f^j(\cdot;\Lambda_j)$ would then be $(R_t, L_t^0, \ldots, L_{t-1})$. Alternatively, it was also possible to learn separate representations $R_t^j = r^j(\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1};\Theta_j)$ for each box so that the box simply computes $f^j(R_t^j;\Lambda_j)$. A basic diagram of the G-Net framework is provided in Figure 4-1. The purpose of the representational layer was to create abstractions of patient histories and allow for more flexibility in how information was shared across variables and time.

4.3 Training

G-Net is fit to a one-step-ahead prediction task that provides us with estimates of the conditional expectations $E[L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1}]$ for all t and j. Given these conditional expectations, we can simulate from $p(L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1})$

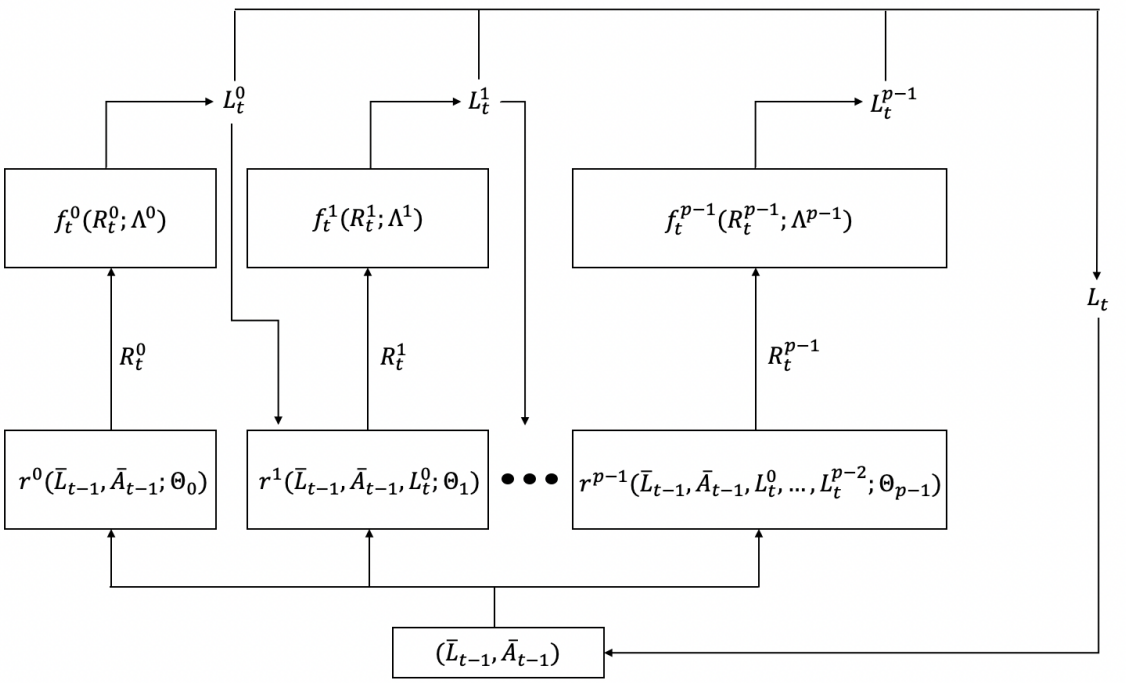

Figure 4-1: Flexible architecture for the G-Net framework using separate representations for each timestep. The treatment A_t at timestep t is calculated deterministically based on the covariates L_t.

as described in Section 4.4. Here we discuss some key design choices made during the training portion of our experiments.

4.3.1 Teacher-Forcing

For each covariate group j, G-Net is trained to predict the values for that group given patient history (i.e. covariates and treatments up to timestep $t-1$) and the values of covariate groups 0 through $j-1$ at timestep t. In this section, we will denote L_t as the ground-truth data and $\hat{L}_t$ as the predicted covariates.

The sequential nature of the G-Net framework gives rise to two alternate methods for training (Figure 4-2). The first method is *student-forcing*, in which the input to each box after $f^0(\cdot;\Lambda_0)$ comprises the ground-truth covariates at time $t-1$ and the predicted covariates at time t from the previous boxes. In other words, box j computes $\hat{L}_t^j = f^j(\bar{L}_{t-1}, \bar{A}_{t-1}, \hat{L}_t^0, \ldots, \hat{L}_t^{j-1}; \Lambda_j)$. Contrasting with *student-forcing* is *teacher-forcing*, where the inputs to every box come from the ground-truth data.

This means that for each box $f^j(\cdot;\Lambda_0)$, we concatenate the ground-truth values for $L_t^0,\ldots,L_t^{j-1}$ at timestep t to the patient history up to timestep $t-1$ and compute $\hat{L}_t^j = f^j(\bar{L}_{t-1},\bar{A}_{t-1},L_t^0,\ldots,L_t^{j-1};\Lambda_j)$. In our experiments, we used teacher-forcing during training, which is essential for learning conditional expectations that are reflective of the observational dataset, from which we can derive the conditional probabilities as required by g-computation.

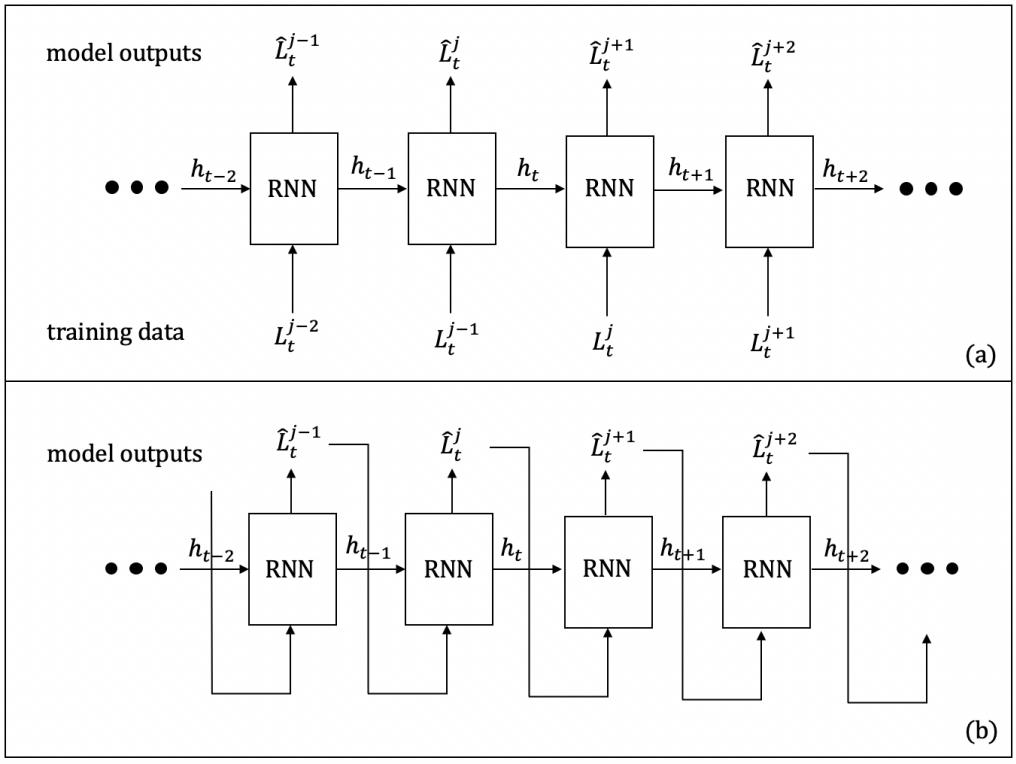

Figure 4-2: Training by (a) teacher-forcing or (b) student-forcing. The input to the first box f^0 at timestep t under both paradigms is $(\bar{L}_{t-1},\bar{A}_{t-1})$.

4.3.2 Loss Optimization

Through optimizing appropriately defined loss functions during training, we can teach G-Net to accurately estimate the covariates at each time point t using standard gradient descent techniques. During training, we can optimize either the joint loss over all boxes or individual losses over each box separately; the former requires training all the boxes together while the latter allows flexibility in training one box at a

time. In both cases, we averaged the loss over all patients and timesteps in a given batch to reduce possible biases related to the number of timesteps. We employed cross-entropy (CE) loss for categorical variables, binary cross-entropy (BCE) loss for binary variables, and mean squared error (MSE) loss for continuous variables. When optimizing the joint loss, an additional weight parameter may be introduced to vary the relative contribution of each covariate group to the overall loss value.

4.4 Simulation

While the training procedure described above provides us with the conditional expectations $E[L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1}]$, our ultimate goal is to simulate from the conditional distributions $p(L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1})$. For boxes that model categorical covariates, the last layer applies a softmax function that produces a vector of probabilities describing the likelihood of the sample being classified as each category. This softmax output can be used to define a categorical distribution from which we sample the value of the categorical variable at simulation time. Similarly, for binary variables, G-Net outputs a single number describing the probability of that variable being assigned the positive class. To obtain the actual value of the variable, we sample from a Bernoulli distribution parametrized by this number.

Variables with continuous densities are more complicated to simulate, but there are various approaches we might take. One possible method takes advantage of the conditional expectations and empirical losses as follows

$$L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1} \sim E[L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1}] + \epsilon_t^j \tag{4.2}$$

where $L_t^j|\bar{L}_{t-1}, \bar{A}_{t-1}, L_t^0, \ldots, L_t^{j-1}$ is the value of the covariate L_t^j conditioned on patient history and ϵ_t^j is randomly sampled from the empirical distribution of the residuals $L_t^j - \hat{L}_t^j$. These residuals are timestep-specific and computed from a holdout validation dataset not used to fit the model parameters (Θ, Λ) that generated the conditional expectations. This approach does not make any parametric assumptions,

which is advantageous because we do not need to limit the application of G-Net to covariates with specific underlying distributions. However, it does make the simplifying assumption that covariate error distribution does not depend on patient history.

Chapter 5

Synthetic Data Generation Using CVsim

Using real-world datasets to evaluate counterfactual predictions presents a challenge because we are only able to observe outcomes from treatments that were actually administered to patients. To evaluate the ability of our models in estimating effects of counterfactual strategies, then, it is necessary to use simulated data in which the "ground truth" effects for various alternative strategies can be obtained through simulation. To this end, we used a physiological simulator, CVSim, to generate synthetic observational and counterfactual datasets for the evaluation of G-Net performance.

5.1 Background on CVSim

CVSim is an open-source program that simulates the dynamics of the human cardiovascular system [10]. Multiple versions of varying complexity are available, though for the purposes of our study, we focused on CVSim-6. This version consists of 6 compartments, two of which function as the left and right ventricles and four of which represent the systemic and pulmonary arteries and veins separated by microcirculation. Two important physiological reflex systems for maintaining blood pressure homeostasis, the arterial baroreflex (ABR) and cardiopulmonary reflex (CPR), are also implemented as part of the model and were turned on in our experiments.

Given a set of input variables defining the hemodynamic system, CVSim-6 is capable of simulating forward various outputs, such as blood pressure, blood flow, and heart rate, under the lumped-parameter hemodynamic model. To generate the synthetic datasets for our experiments, we modified and extended CVSim by adding (a) treatment functions to allow for intervention actions to be taken at each timestep, and (b) stochastic components in order to generate patients with different baseline physiological measurements.

5.1.1 Inputs

To simulate the cardiovascular system under different conditions, we can vary one or more of CVSim's hemodynamic parameters, such as total peripheral resistance, arterial compliance, nominal heart rate, and zero-pressure filling volume. At the start of simulation for each new patient, we randomly generated values for the parameters listed in Table A.1 to set their baseline physiological state. The ranges referenced in Table A.1 were determined with clinical guidance based on plausible physiological values and were only used to set the patient's initial parameters (i.e. they do not bound the covariates for later timesteps $t > 0$). Note that altering these parameters causes the CVSim program to deviate from its original steady state and leads to activation of the ABR and CPR to achieve a new steady state. We ensured this new steady state was reached prior to beginning data collection of outputs, described in Section 5.1.2. The first timestep in which data is collected is taken to be $t = 1$, and the preceding timestep is $t = 0$.

5.1.2 Outputs

Given the initial settings of the input variables, CVSim deterministically simulates forward 25 hemodynamic outputs, including vascular resistance and variables characterizing blood flow, pressure, and volume. In addition to these outputs, we derived 4 additional outputs for our experiments: systolic blood pressure, diastolic blood pressure, mean arterial pressure, and a pulmonary edema indicator, yielding a total of

29 variables collected at each timestep. Below are the definitions of the additional parameters:

- **Systolic Blood Pressure (SBP):** the highest measured arterial blood pressure while the heart is contracting.
- **Diastolic Blood Pressure (DBP):** the lowest measured arterial blood pressure while the heart is contracting.
- **Mean Arterial Pressure (MAP):** the average arterial pressure throughout one cardiac cycle, computed as MAP $= \frac{2}{3}$DBP $+ \frac{1}{3}$SBP. It is sometimes also referred to as mean blood pressure (MBP).
- **Pulmonary Edema (PE):** a binary variable indicating pulmonary venous pressure above a certain threshold, determined as PE = [PVP $>$ 25mmHg] in our study.

The complete list of output covariates from CVSim is provided in Table A.2, with the subset of covariates used in subsequent G-Net experiments bolded. By modeling only a subset of variables in our dataset, we ensured that there were long range temporal dependencies present that were mediated by the excluded variables. We will hereon refer to these covariates by their abbreviations.

5.2 Data Generation

We generated three datasets with CVsim: a simulated observational dataset D_o under treatment regime g_o and two different counterfactual datasets D_{c1} and D_{c2} under alternative regimes g_{c1} and g_{c2}. Our goal was to learn the conditional distributions of the covariates using D_o during training and evaluate the resulting model on a counterfactual prediction task using D_{c1} and D_{c2} during testing.

An important concern for physicians presented with hypotensive patients in a clinical setting is to restore those patients' blood pressure (specifically, MAP and CVP) to some reasonable level. This can be accomplished by administering fluids

or vasopressors, which increase blood volume and vascular resistance, respectively. Provided with this information, we developed an observational treatment strategy g_o that was stochastic and employed fluids and vasopressors in combination.

Under g_o, the probability of receiving a non-zero dose of either fluids or vasopressors at a given timestep increased as the MAP and CVP decreased according to a logistic function. Conditioned on the treatment being non-zero, the dose was sampled from a normal distribution with mean inversely proportional to MAP and CVP. The counterfactual regimes g_{c1} and g_{c2} were similar to g_o, except that they were deterministic and used different coefficients relating treatment to MAP and CVP. Regime-specific details are provided in Sections 5.2.3 and 5.2.4.

The generation process for each of the three datasets differed only in the treatment assignment rules; all other aspects were held constant. In D_{c1} and D_{c2}, patients were assigned treatment according to g_o for the first $m-1$ simulation timesteps; then for timesteps m to K, the strategy switched to one of g_{c1} or g_{c2} depending on the dataset. This is illustrated in Figure 5-1 for a single sample patient simulated by CVSim. Since the first instance of the counterfactual strategy is administered at timestep m, any changes in covariates as a result of the new regime will be observed starting at timestep $m+1$.

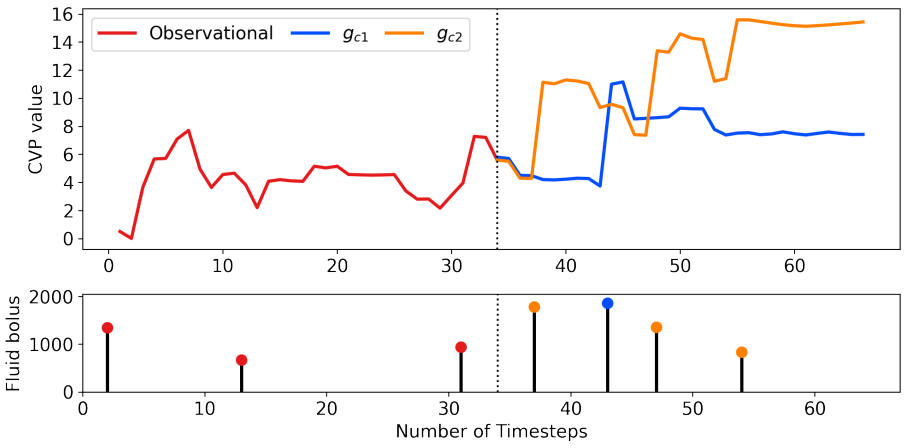

Figure 5-1: Covariate trajectories for the same patient under two different treatment (fluid bolus) strategies: g_{c1} (blue) and g_{c2} (orange) starting at $t = 34$ (black dashed line).

5.2.1 Disease Simulation

In real-world scenarios, patients are only given treatment when afflicted with some pathological condition. To model the disease process in CVSim, we introduced a function for injecting either sepsis or blood loss events into our simulated patients. At each timestep, patients developed disease with probability $P(\text{disease}) = 0.05$; conditioned on developing disease at timestep t, patients could experience either sepsis or blood loss with probability $P(\text{sepsis}|\text{disease}) = P(\text{blood loss}|\text{disease}) = 0.5$. These two events were mutually exclusive within a single timestep, but it did not preclude the patient from experiencing both diseases at different time points across the course of the simulation. Sepsis and blood loss were modeled deterministically as follows:

- To simulate sepsis, we decreased TPR at time $t+1$ by setting $L_{t+1}^{tpr} = \alpha_{sep} \times L_t^{tpr}$, where $\alpha_{sep} = 0.7$.
- To simulate blood loss, we decreased TBV at time $t+1$ by setting $L_{t+1}^{tbv} = \alpha_{loss} \times L_t^{tbv}$, where $\alpha_{loss} = 0.85$.

5.2.2 Treatment Simulation

At every timestep of simulation, patients could be given fluids, vasopressors, or neither if no treatment was needed; the two treatments were mutually exclusive in the CVSim data generation experiements and patients could not receive both at the same time. To represent the intervention action taken at time t, we define the vector $A_t = (A_t^1, A_t^2)$, where A_t^1 denotes the amount of fluid administered and A_t^2 denotes the amount of vasopressor administered. In CVSim, administration of fluids was modeled by an increase in TBV while administration of vasopressors was accounted for by an increase in TPR.

Treatment at time t was a function of patient covariate history and given by $A_t = g(\bar{L}_t)$ for arbitrary intervention strategy g. The probability of choosing fluids versus vasopressors was dependent on whether the patient had pulmonary edema L_t^{pe}: since pulmonary edema indicates fluid overload, it was less likely for additional

fluids to be administered to these patients and more likely for vasopressors to be used instead. The dosage of the treatment depended on a subset of covariates, specifically MAP and CVP, which were chosen based on the fact that maintaining adequate blood pressure is an important goal in clinical practice [26]. The target MAP was 70 mmHg while the target CVP was 10mmHg, so the dosages varied as a function of $\Delta_{map,t} \equiv 70 - L_t^{map}$ and $\Delta_{cvp,t} \equiv 10 - L_t^{cvp}$. The more positive the difference between current and target pressure, the greater the dose of treatment administered. If the values of $\Delta_{map,t}$ and $\Delta_{cvp,t}$ produced a treatment dose that was negative, then no treatment action was taken at timestep t (i.e. we set the dose amount to zero).

5.2.3 Observational Regime

The observational treatment strategy g_o was stochastic and determined as a function of patient MAP and CVP according to the steps below:

1. Compute the probability of treatment $P(A_t|L_t) = \frac{1}{1+e^{-x}}$, where $x = \gamma_1 \times \Delta_{map} + \gamma_2 \times \Delta_{cvp} + \gamma_0$ to determine whether treatment is administered.

2. If treatment is administered, determine whether the patient should receive fluids or vasopressors, where the probability of fluids $P(A_t^1|L_t) = \rho_1 - \rho_2 \times L_t^{pe}$.

3. If fluids are administered, generate the dose (in mL) as $A_t^1 \sim \max(0, \beta_1^1 \times \Delta_{map,t} + \beta_2^1 \times \Delta_{cvp,t} + \mathcal{N}(0, 500))$ and update TBV as $L_{t+1}^{tbv} = L_t^{tbv} + A_t^1$.

4. If vasopressors are administered, generate the dose as $A_t^2 \sim \max(0, \beta_1^2 \times \Delta_{map} + \beta_2^2 \times \Delta_{cvp} + \mathcal{N}(0, 1))$ and update TPR as $L_{t+1}^{tpr} = L_t^{tpr} + 1 - \frac{1}{A_t^2+1}$.

After experimenting with the various parameters introduced above, we opted to use the following settings: $\gamma_0 = 0.65, \gamma_1 = 0.3, \gamma_2 = 0.24, \rho_1 = 0.5, \rho_2 = 0.2, \beta_1^1 = 20, \beta_2^1 = 120, \beta_1^2 = 0.2, \beta_2^2 = 0.3$.

5.2.4 Counterfactual Regimes

Unlike g_o, counterfactual regimes g_{c1} and g_{c2} were deterministic; however, the treatment dose administered under either strategy was still dependent on patient MAP

and CVP. Counterfactual strategy g_{c1} is outlined as follows:

1. If $L_t^{map} < 65$mmHg, the probability of treatment is $P(A_t|L_t) = 1$; otherwise, no treatment is administered.

2. If $L_t^{pe} = 0$, the probability of administering fluids is $P(A_t^1|L_t) = 1$; otherwise, $P(A_t^1|L_t) = 0$ and vasopressors are administered instead.

3. Given that fluids are administered, generate the dose (in mL) as $A_t^1 \sim max(0, \beta_1^1 \times \Delta_{map,t} + \beta_2^1 \times \Delta_{cvp,t} + \beta_0^1)$ and update TBV as $L_{t+1}^{tbv} = L_t^{tbv} + A_t^1$.

4. If vasopressors are administered, generate the dose as $A_t^2 \sim max(0, \beta_1^2 \times \Delta_{map} + \beta_2^2 \times \Delta_{cvp})$ and update TPR as $L_{t+1}^{tpr} = L_t^{tpr} + 1 - \frac{1}{A_t^2+1}$.

Counterfactual strategy g_{c2} was similar to g_{c1}, except that $P(A_t|L_t) = 1$ if and only if $L_t^{map} < 75$mmHg. This was a more aggressive intervention strategy where the threshold to qualify for treatment was lower. The settings for the parameters under g_{c1} and g_{c2} were the same as g_o, and we also introduced the additional constant β_0^1, which was set to 1000 in our experiments.

5.2.5 Patient Generation Process

To generate the simulation trajectory for a single patient under treatment strategy g in CVSim, we first initialized the input variables V in Table A.1 by drawing from independent uniform distributions bounded by the the defined ranges. At each timestep from 0 to K, we did the the following:

1. Generate L_t from the outputs of CVSim, which are a function of $(V, \bar{L}'_{t-1}, A_{t-1}, S_{t-1})$. Note that $A_{t-1} = 0$ and that $\bar{L}'_{t-1}$ is equal to $\bar{L}_{t-1}$ except with TBV and TPR altered post-hoc (i.e. after $\bar{L}_{t-1}$ is recorded) according to A_{t-1}.

2. Generate A_t as $g(\bar{L}_t)$.

3. Generate $\bar{L}_t{}'$ from $\bar{L}_t$ according to A_t.

Using CVSim, we generated 15,000 trajectories under g_o and 1,000 trajectories each under g_{c1} and g_{c2}. All patients were simulated for 66 timesteps in length. After the generation process was complete, we removed patients whose non-treatment covariates at the first timestep exceeded one or more of the following thresholds:

- MAP $>$ 250 mmHg
- HR $>$ 250 beats per minute
- All other non-treatment covariates $> 99^{th}$ percentile of the dataset

Once patients with physiologically improbable values were filtered out, we were left with 12,774 trajectories under the observational regime and 851 trajectories under each counterfactual regime. We removed an additional 774 trajectories under the observational regime to obtain a dataset of exactly 12,000 samples, of which 10,000 (83%) were used for training and 2,000 (17%) were used for validation. There was no overlap of patients between the training, validation, and counterfactual (i.e. testing) datasets.

To allow for accurate comparisons between the effects of g_{c1} and g_{c2}, we used the same $n = 851$ patients in D_{c1} and D_{c2}; that is, the set of trajectories for the first $m-1$ timesteps in the counterfactual datasets were identical, and only diverged starting from $t = m$ once the alternative strategies were applied (Figure 5-1). We can conceptualize this setup as having n patients who have received $m-1$ timesteps of the observational regime and are now being tested under multiple alternative treatment paths starting at timestep m. While only one path is possible in real-world settings, multiple paths can be explored in this simulated environment.

5.2.6 Data Transformations

To adjust continuous-valued covariates to a similar scale, we normalized values by subtracting the mean and dividing by the standard deviation. That is, we computed $x' = \frac{x-\bar{x}}{\sigma(x)}$ and used x' as the input to our models, where the mean $\bar{x}$ and standard deviation $\sigma(x)$ were both derived from the training dataset; this is known as Z-score

standardization. There were no categorical variables in the set of outputs we collected from CVSim, but binary variables were treated as categorical and one-hot encoded.

Chapter 6

Evaluating G-Net Performance on CVSim Data

Using the datasets generated by CVSim, we assessed the ability of G-Net to estimate patient outcomes under alternative treatment strategies. These experiments were critical in validating our model as we had access to ground-truth counterfactual datasets during testing time, which was not the case for our later studies using real-world clinical data. In this chapter, we describe the experimental setup and results of applying G-Net to the CVSim data, where we trained the model on a one-step-ahead prediction task using trajectories from the observational dataset D_o and evaluated its performance on counterfactual simulation on the test datasets D_{c1} and D_{c2}. Starting with the basic model architecture described in Chapter 4, we experimented with 4 different implementations of the G-Net framework, described below.

6.1 Experimental Setup

A typical clinical scenario may involve a physician who has observed a patient for $m-1$ timesteps under the observational regime and would like to predict how they will respond to a different treatment strategy. Given data D_o on prior patients who received the observational regime for K timesteps, and assuming that the current patient is from the same population as those in D_o, we can estimate the effects of the

counterfactual regime on this particular patient from timestep m onward, conditioned on their history from timesteps 0 through $m-1$. This situation is, in fact, a standard use case for counterfactual prediction and is elaborated on in a more detailed example in Section 6.1.1. While real-world limitations preclude the ability to test multiple regimes on a single patient starting at the same physiological state, we can simulate such a scenario using the CVSim datasets, which provide ground-truth trajectories for the same patient under different intervention strategies.

Our approach was to train G-Net on D_o, comprising patients who were observed under the observational regime for K timesteps, and use it to predict the trajectories of patients in D_{c1} and D_{c2} for timesteps m to K under different counterfactual regimes. During prediction, G-Net was fit to the first $m-1$ timesteps of each patient trajectory; then for $t \geq m$, the model computes the appropriate treatment to apply at each timestep under the counterfactual strategy of interest and predicts the resulting covariates. In our experiments, we generated $M = 100$ Monte Carlo simulations for each patient in D_{c1} and D_{c2} according to Algorithm 1. This yielded a total of $M \times N$ simulations under each counterfactual strategy g_{c1} and g_{c2}, where N was the number of ground-truth trajectories in each of the counterfactual datasets.

6.1.1 Sample Use Case

Figure 6-1 illustrates how G-Net could be used for decision-making with individual patients using a hypothetical scenario. Both patients (a) and (b) have relatively similar MAP trajectories in the first half of their trajectories, during which they received the observational regime. After some time (e.g. at $t = 34$ in the example), we may be interested in switching to a different strategy and want to be able to predict how each patient will respond. Based on the simulations, we see that while patient (a)'s MAP is projected to increase significantly under a more aggressive fluid strategy g_{c2} compared to a less aggressive strategy g_{c1}, patient (b)'s MAP does not show much difference between the two treatment options, potentially because their blood volume is already very high. High blood volume and constant MAP may suggest that using fluids to treat patient (b) is not an effective strategy. If these were real patients, a

clinician interpreting these results might choose to administer fluids to patient (a) but not to patient (b) because the small predicted gain in MAP in patient (b) would not be worth the risk of fluid overload.

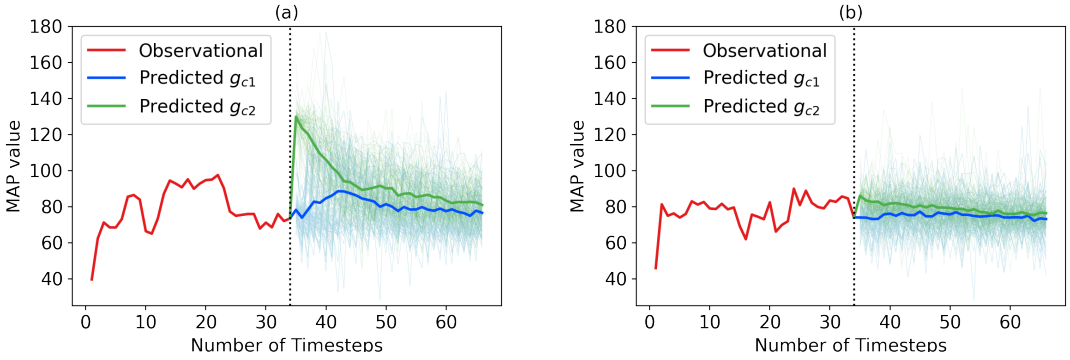

Figure 6-1: Estimated effects of two treatment strategies g_{c1} (blue) vs g_{c2} (green) in two patients with similar observed MAP (red). The 100 Monte Carlo simulated trajectories are shown in light blue and green respectively. Under g_{c2}, both patients would receive similar volume of fluid, but patient (a)'s predicted treatment effect is larger compared to patient (b)'s, possibly due to differences in their underlying physiology. Predicted treatment effect under g_{c1} is similar for both patients.

6.1.2 G-Net Implementation

For the CVSim experiments, we focused on a 2-box model of G-Net, splitting covariates into $p = 2$ groups so that all the categorical variables L_t^0 were modeled by one box and all the continuous variables L_t^1 were modeled by a second box. By experimenting with the shared representation layer r and the covariate boxes, we developed 4 implementations of G-Net, described below.

- (*Linear*): a GLM baseline employing 2 non-temporal linear layers to model the covariate groups.
- (*LSTM1*): similar to (*Linear*) with an LSTM representational layer (r = LSTM). This model also uses linear layers to model the categorical and continuous covariates separately.
- (*LSTM2*): bypasses representation learning (r = Identity) and uses LSTMs for the 2 covariate boxes.

- (*LSTM3*): combines the features of (*LSTM1*) and (*LSTM2*) by employing separate LSTMs to implement representational learning and model the 2 covariate groups.

As an additional baseline, we also explored multi-layer perceptrons (MLP) as a non-linear estimator. This was important to demonstrate how flexible non-linear modeling and automatic construction of relevant summaries of patient history provided by LSTMs can help improve counterfactual prediction performance. All five models were fit to a one-step-ahead prediction task during training time and used to simulate covariates forward during testing time.

For (*LSTM1*) and (*LSTM3*), we used a representation layer that was shared across the two boxes, so that at timestep t, box f^0 computed $f^0(R_t; \Lambda_0)$ and box f^1 computed $f^1(R_t, L_t^0; \Lambda_0)$, where $R_t = r(\bar{L}_{t-1}, \bar{A}_{t-1})$. In order to train this representation layer appropriately, we elected to optimize the joint loss of boxes $f^0(\cdot; \Lambda_0)$ and $f^1(\cdot; \Lambda_1)$, rather than separate box-specific losses.

6.1.3 Model & Training Parameters

To optimize our model, we tuned over the hyperparameter space displayed in Table 6.1. During training, we used the Adam optimizer with early stopping for a maximum of 50 epochs. The early stopping window was 10 steps and the stop tolerance was 0.01. The experiments were performed on NVIDIA Tesla V100 SXM2 GPUs. Note that due to limited compute resources, we did not tune over the number of layers when training *LSTM3*, so the number of layers (representational, categorical, and continuous) was always set to 1.

6.2 Evaluation

There are a number of different evaluation methods that can be used to assess the performance of the various G-Net models on the counterfactual prediction task. In our study, we focused on the root mean squared error (RMSE) and calibration over

Table 6.1: Hyperparameter search space for G-Net in CVSim experiments. *Denotes parameter setting in the optimal parameter set for this model.

	Hyperparameters	Search Range
Linear	Learning Rate	0.001*, 0.01
MLP	Number of Layers	2*, 4
	Hidden Dimension	16, 32*
	Learning Rate	0.001*, 0.01
LSTM 1	Number of Layers (Representation)	1, 2, 4*
	Hidden Dimension (Representation)	16, 32*
	Learning Rate	0.001, 0.01*
LSTM 2	Number of Layers (Categ & Contin)	1, 2, 4*
	Hidden Dimension (Categorical)	8*, 16
	Hidden Dimension (Continuous)	32, 64*
	Learning Rate	0.001*, 0.01
LSTM 3	Hidden Dimension (Representation)	16, 32*
	Hidden Dimension (Categorical)	8*, 16
	Hidden Dimension (Continuous)	32, 64*
	Learning Rate	0.001, 0.01*

time. As a qualitative assessment for each covariate, we analyzed the general shape of the simulated population-level and individual patient-level trajectories in comparison to the ground-truth trajectories.

6.2.1 Calculating RMSE

We evaluated the accuracy of the simulated counterfactual trajectories against the ground-truth counterfactual trajectories by computing the RMSE as follows. Consider a counterfactual dataset D_c with N trajectories. For each patient i in D_c, we use G-Net trained on D_o to generate M simulations predicting the covariate trajectories for that patient under g_c. These simulations are illustrated by the light blue lines in Figure 6-2, which displays the MAP of a sample patient in D_{c1}. In our experiments, M was set to 100.

For patient i, let us use L_{ti} to denote the coviarates at time t of the ground-truth counterfactual trajectory and $\tilde{L}_{tik}$ to denote the covariates at time t of the kth simulated counterfactual trajectory, where $k \in \{1, \ldots, M\}$. Furthermore, let us

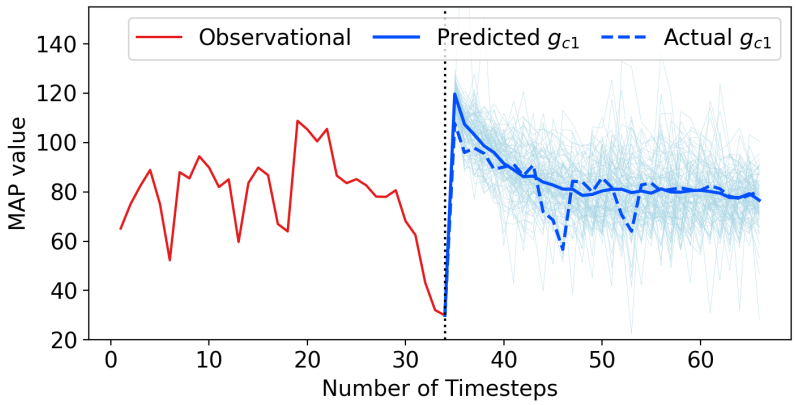

Figure 6-2: G-Net simulated MAP trajectories (100 Monte Carlo simulations in light blue, average in solid dark blue) and ground truth (dashed dark blue) for one patient under g_{c1} (starting $t = 34$).

use the superscript c to denote individual covariate values, such that L_{ti}^c represents the value of the cth covariate in the vector L_{ti}. For a given covariate c, the point prediction $\hat{L}_{ti}^c$ that G-Net makes for L_{ti}^c can be obtained by averaging the results of the M simulations, that is, $\hat{L}_{ti}^c = \frac{1}{M}\sum_{k=1}^{M} \tilde{L}_{tik}^c$. This is the dark blue line in Figure 6-2. Individually taking the average across simulations for all covariates produces $\hat{L}_{ti}$, the estimate for the full covariate vector L_{ti}. To compute the MSE, we can average the difference between L_{ti} and $\hat{L}_{ti}$ across all covariates, timesteps, and patients as shown in Equation 6.1.

$$\text{MSE} = \frac{1}{N(K-m)d}\sum_{i=1}^{N}\sum_{t=m}^{K}\sum_{c=1}^{d}(L_{ti}^c - \hat{L}_{ti}^c)^2 \tag{6.1}$$

The RMSE is simply the square root of the MSE given above. Treatment variables are not included in the calculation of RMSE as they are experimentally adjusted based on the counterfactual strategy of interest.

6.2.2 Determining Model Calibration

In addition to having the actual trajectory align closely with the average of the simulated trajectories, we ideally also want it to fall within the range of all the simulated trajectories. A model that is able to produce such simulations is considered to be *well-calibrated*.

To formally define calibration, let us start by noting the lower and upper quantiles α_{low} and α_{high}. Calibration measures the frequency with which the actual counterfactual covariate L^c_{ti} is between the α_{low} and α_{high} quantiles of the M simulated values $\tilde{L}^c_{tik}$, for $k \in \{1, \ldots, M\}$. A higher frequency of actual trajectories that are within the lower and upper quantiles of their corresponding simulated trajectories suggests a better calibrated model; ideally, the frequency should be close to $\alpha_{high} - \alpha_{low}$. In our experiments, we set $\alpha_{low} = 0.05$ and $\alpha_{high} = 0.95$ and targeted a frequency of 0.9.

6.2.3 Analyzing Population Covariate Trajectories

For each covariate, we can conduct a visual inspection of model performance by comparing population-level simulated trajectories for that covariate to the population-level actual trajectories. This makes sense since our goal is to have G-Net be able to predict counterfactual outcomes for a certain population of interest. A well-performing model will have the population-level simulated trajectories closely match those of the actual trajectories for most, if not all, covariates. For continuous variables, we take the average across patients at each timestep. For binary variables (or binary variables modeled as categorical), we plot the proportion of patients assigned to the positive class at each timestep. There were otherwise no categorical variables in the CVSim dataset.

6.3 Results

We first fit G-Net to the 10,000 samples in the training portion of D_o and used the remaining trajectories for validation. All trajectories comprised 66 timesteps. Given observed covariate history through timestep $t = 34$ and treatment history through timestep $t = 33$ for patients in D_{c1} and D_{c2}, we then simulated covariates forward under each counterfactual strategy for 32 timesteps, from $t = 35$ to $t = 66$. Since the first instance of the counterfactual strategy was administered at $t = 34$, the effect on covariates was not observed (predicted) until $t = 35$.

6.3.1 RMSE Over Time

The RMSE of the simulated counterfactual trajectories in comparison to the actual counterfactual trajectories was computed for timesteps 35 to 66 and the results for both g_{c1} and g_{c2} are displayed in Figure 6-3. The best performing model was (*LSTM3*), though the performance of all 3 LSTM implementations of G-Net showed significant improvements over the GLM baseline. In particular, the advantage increased over time under both counterfactual regimes, as illustrated by the widening gap between the linear RMSE curve and the LSTM RMSE curves. This is reasonable as LSTMs are expected to handle long-range dependencies better than simpler GLM models.

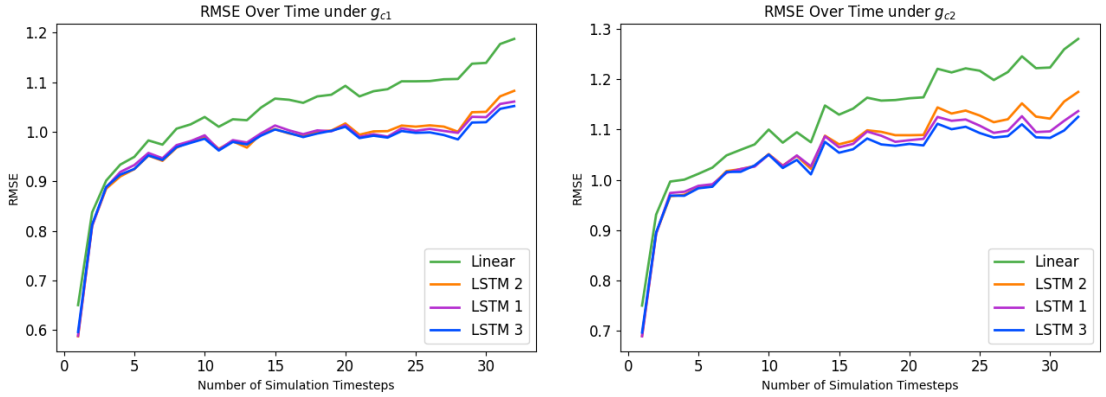

Figure 6-3: Normalized RMSE averaged across all output covariates over time under counterfactual regimes g_{c1} and g_{c2}.

As seen in Figure 6-3, we did not include the results for MLP because the RMSE over time (1.03 at $t = 35$ to 1.80 at $t = 66$) was much higher than both the LSTM and GLM models. On the other hand, the MLP model showed better validation loss than the GLM (0.48 for MLP versus 0.51 for GLM) on the one-step-ahead prediction task, which is reasonable given the flexibility provided by its non-linear functions. But ultimately, this flexibility without suitable incorporation of patient history led to unstable counterfactual prediction and poor performance in the multi-step-ahead simulation task.

6.3.2 Calibration Over Time

Calibration plots under g_{c1} and g_{c2} are provided in Figure 6-4. While the calibration of the linear model is comparable to the LSTM models at the beginning of simulation, its performance decreases more quickly at later timesteps. This is consistent with the RMSE results, which also demonstrate the advantage afforded by LSTMs as the number of simulation timesteps grows. Under both counterfactual regimes, all three LSTM models outperform the linear implementation in the latter portion of the simulation.

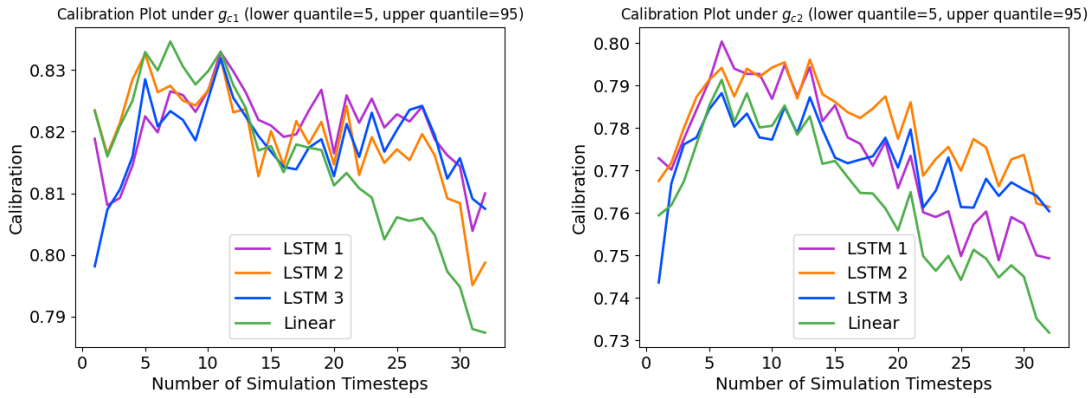

Figure 6-4: Model calibration across all output covariates over time under counterfactual regimes g_{c1} and g_{c2}.

Even so, we observe that the calibration coverage rates for all four models are somewhat below the target level of 0.9, with the frequency decreasing over time. This behavior is expected as errors accumulate and propagate with increasing number of simulation timesteps, causing the predicted trajectories to diverge further from ground-truth; in addition, the result is in line with the increased RMSE over time seen in Figure 6-3. One hypothesis for the less-than-nominal calibration rates may be that the counterfactual predictive density estimates in our experiments do not take into account uncertainty about model parameter estimates, a challenge that could potentially be addressed with variational dropout in future studies. Note that we again exclude MLP from the calibration plots due to significantly poorer performance compared to either the LSTM or linear models.

6.3.3 Population-Level Trajectories

To further analyze G-Net's performance in estimating population-level counterfactual outcomes, it is helpful to visualize the average simulated trajectories for individual covariates in comparison to the ground-truth data. This can provide insight into which variables are simulating well or poorly and how different covariates might be linked. The average estimated and actual trajectories under g_{c1} and g_{c2} are plotted in Figures 6-5 and 6-6, respectively, for a majority of the CVSim outputs used in our experiments. We excluded pulmonary edema in these figures as the incidence of the condition was extremely low (less than 0.5%) in the population; thus, all covariates displayed are continuous.

From these trajectory plots, we again see that G-Net outperforms GLMs in estimating outcomes under alternative treatment strategies for a majority of variables, with (*LSTM3*) demonstrating the most accurate predictions. For some covariates (e.g. CVP, AQ, TBV) under the linear model, the predicted trajectories are similar to ground-truth at earlier timesteps before diverging over time, a result that is similar to the RMSEs computed in Figure 6-3. For other covariates (e.g. MAP, VT, LVC), the linear model seems to deviate from ground-truth immediately after the switch from observational to counterfactual regime, suggesting that only data from the previous timestep is being used to predict values at the next timestep. This is not the case with LSTMs, which are able to incorporate the entire patient history under the observational regime when simulating covariates forward under the new treatment strategy, leading to better estimations.

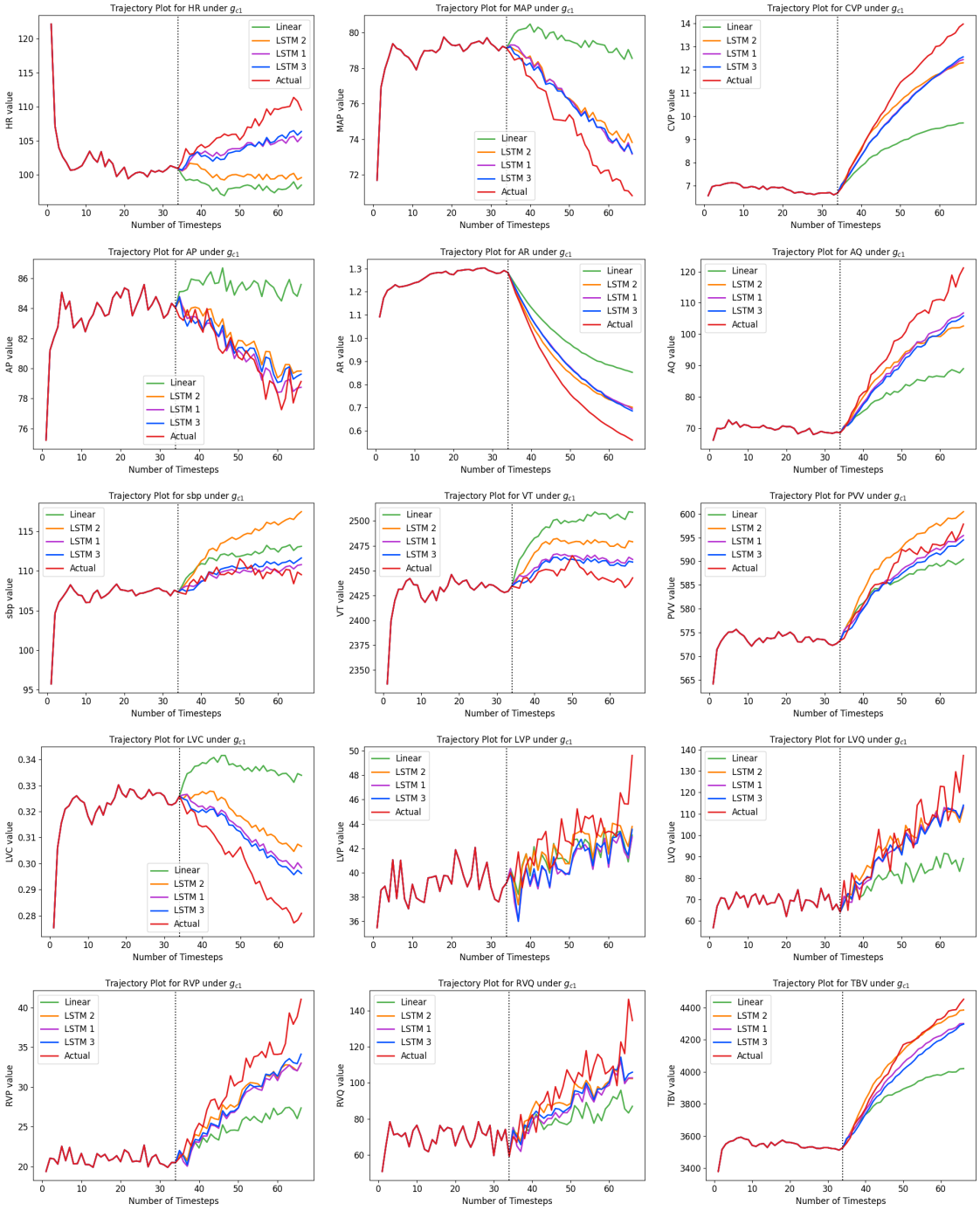

Figure 6-5: Estimated and actual population average trajectories under g_{c1} for selected covariates. All LSTM implementations are shown in comparison to the GLM baseline.

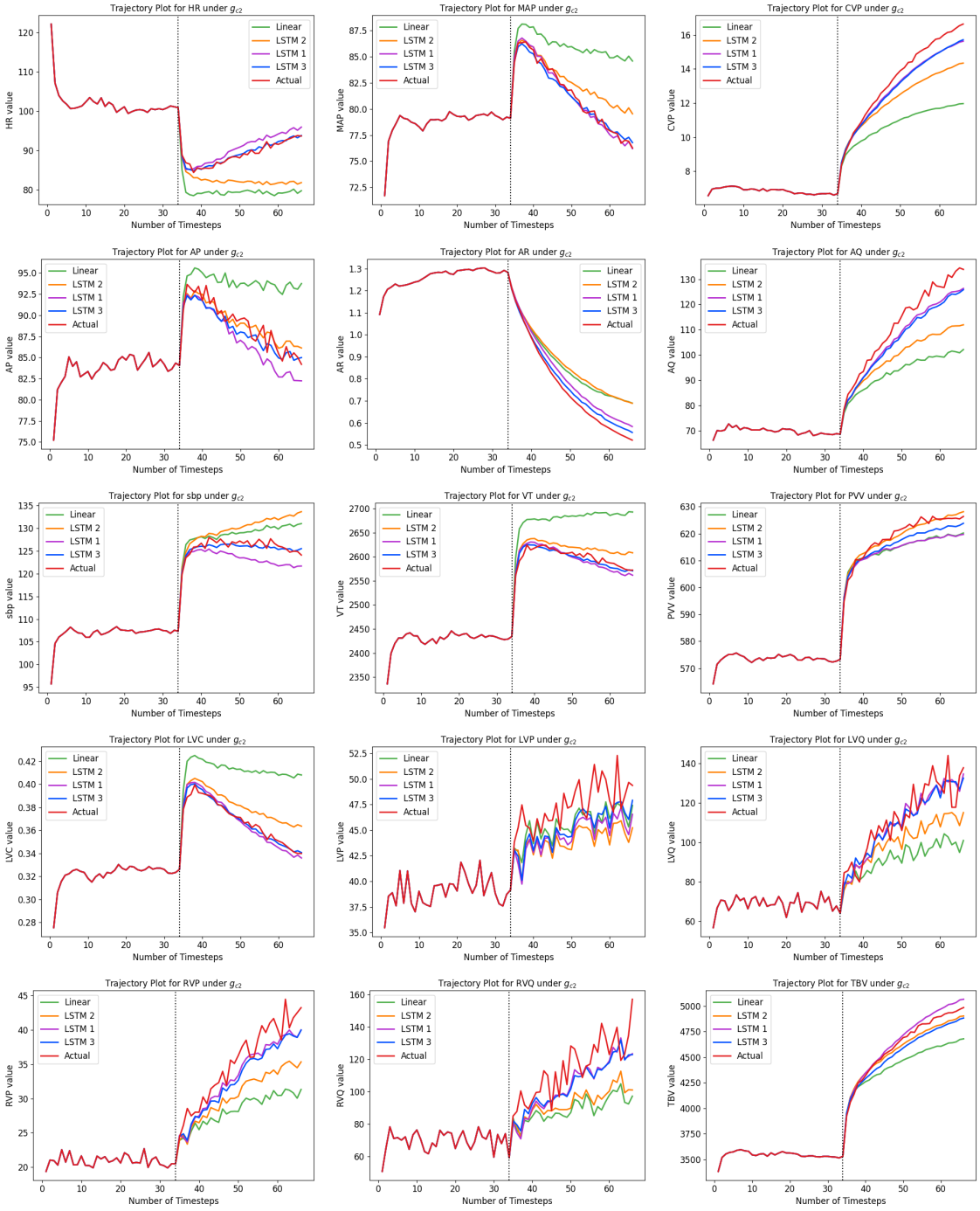

Figure 6-6: Estimated and actual population average trajectories under g_{c2} for selected covariates. All LSTM implementations are shown in comparison to the GLM baseline.

Chapter 7

Sepsis Dataset and Cohort Selection

Having validated the ability of G-Net to make predictions under counterfactual treatment strategies on a simulated dataset, we now turn to the problem of analyzing effects of alternative interventions on a real-world cohort of sepsis patients. Sepsis patients are typically treated with a combination of intravenous fluids and vasopressors, but the exact balance may vary from patient to patient according to the status of their condition, their underlying physiology, and their treatment history. Predicting how each individual may respond to a given intervention is difficult, but having the ability to do so would be useful for clinicians in the ICU attempting to select the optimal course of action for their patients. In this situation, a tool like G-Net could help estimate the effects of alternative treatment strategies and assist physicians in their decision-making processes.

This chapter presents background regarding the cohort of sepsis patients used in our experiments, including details on the study design, criteria for inclusion and exclusion, and procedures for extraction and processing of raw clinical data. All data employed in this work was compiled using the Medical Information Mart for Intensive Care IV (MIMIC-IV) database v1.0, containing medical records from more than 523,500 hospital admissions and 76,500 ICU stays at the Beth Israel Deaconess Medical Center (BIDMC) between 2008 and 2019 [15].

7.1 Cohort & Study Design

Our cohort was limited to ICU stays in which the patient was identified as septic under the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3) and did not meet the criteria for exclusion described in Section 7.1.3. The final dataset consisted of 8,532 ICU stays, each associated with a distinct patient. Statistics on cohort characteristics are provided in Table A.3.

7.1.1 Sepsis-3 Definition

The Sepsis-3 criteria provides the most up-to-date definitions for identifying sepsis and septic shock [32]. Under Sepsis-3, patients are classified as septic if they meet both of the following requirements: (a) an episode of suspected infection and (b) a Sequential Organ Failure Assessment (SOFA) score of 2 points or more [14, 32]. An episode of suspected infection is defined as one of the following: (a) an antibiotic was administered and a culture was sampled within 24 hours of the antibiotic or (b) a culture was sampled and an antibiotic was administered within 72 hours of the culture. The earlier of the two events (antibiotic administration and culture sampling) is used as the time of onset of suspected infection.

7.1.2 Study Period

Previous studies have shown that sepsis prognosis is strongly determined by treatments given in the early hours of a patient's ICU stay [1, 30, 31]. Because of this, our experiments focused on analyzing how intervention strategies implemented within 24 hours of ICU admission affect various clinical outcomes in the first 72 hours of the stay, along with overall in-hospital mortality. The start of our study coincided with ICU admission, while the end was taken to be the earlier of 72 hours and the end of the stay (i.e. due to death or release). During simulation, we only considered timesteps up to 24 hours (or earlier if the patient left the ICU), as this was the period in which treatments of interest were administered. For patients still in the ICU after 24 hours, we assumed that they returned to the observational regime.

To determine ICU admission (i.e. start time) for our study, we used the first heart rate measurement taken in the interval between 6 hours preceding and following the ICU admission time documented in MIMIC. If the first heart rate was taken outside the target interval, we used the documented ICU admission time as the actual ICU admission time. This modification was made because the documented ICU admission time is not necessarily reliable [14]. On the other hand, it is typical for patients to have their vital signs measured immediately upon admission to the ICU, which suggests that the first heart rate recording is a reasonable indication of when their stay began.

7.1.3 Study Population

ICU patients were required to meet the Sepsis-3 criteria outlined in Section 7.1.1 to be included in our study. All patients were adults (age 16 or above) at the time of sepsis onset and none had missing data, yielding a total of 27,139 patients with 35,010 sepsis-related ICU stays among them (as some patients were admitted on more than one occasion).

Since we used the ICU admission time as the starting point of our study, patients whose time of suspected infection was greater than 24 hours after ICU admission were excluded from the analysis. In addition, patients admitted to the ICU following cardiac, vascular, or trauma surgery were also removed because they possess a mortality risk that is inherently different from other ICU patients [14]. For patients with more than one ICU stay in MIMIC-IV, only the first stay was included in our study to avoid repeated measures. Of the 35,010 sepsis-related ICU stays, there were 11,109 stays that were secondary (or greater), 4,973 stays admitted from the cardiothoracic surgical service, and 1,110 stays with a late suspected infection time. There was some degree of overlap between these exclusion groups, and in total we excluded 18,135 stays meeting one or more of these criteria.

From the remaining 16,875 stays (each associated with a unique patient), we further excluded individuals who did not have any documentation in MIMIC regarding fluids administered prior to their ICU stay. Most, if not all, sepsis patients are

expected to receive some form of fluid therapy before being admitted to the ICU, so a lack of pre-ICU documentation likely indicates a failure to record the information rather than an actual case of zero pre-ICU fluids being given. Including these patients without appropriately accounting for their pre-ICU intake would cause issues with confounding, so we excluded them from our study. After removing these patients, we were left with a cohort size of 8,963.

The last step in generating our study population was filtering out patients who had outlier values of certain measurements determined by clinical expertise. These limits are displayed in Table A.4, with additional details including units given in Table A.6. Notably, we capped the amount of pre-ICU intake at 10L (10,000mL), since patients who receive a higher volume are unlikely to be given more fluids during treatment. As we were interested in using resuscitation fluids as part of our intervention strategies, it would not be meaningful to include these patients in our experiments. The final size of our study population was 8,532 patients with one sepsis-related ICU stay each. During our experiments, we used 6,825 (80%) patients in the training set, 853 (10%) in the validation set, and 854 (10%) patients in the testing set. An outline of the cohort construction process is provided in Figure 7-1.

7.1.4 Covariates

As predictors to our model, we selected covariates that are typically monitored in the ICU and important for determining sepsis intervention strategies, as well as potential confounders. The covariates we used were similar to that of Li et al. [18], encompassing, but not limited to, basic demographic information, an Elixhauser comorbidity score, a SOFA score, laboratory values and vital signs, and urine output [18]. A comprehensive list is provided in Tables A.5 and A.6. The demographics, comorbidities, and pre-ICU fluids were unmodeled and regarded as static while the remaining variables were modeled and regarded as dynamic (time-varying). For convenience, the intervention and outcome variables described in Sections 7.1.5 and 7.1.6 were also treated as covariates during modeling.

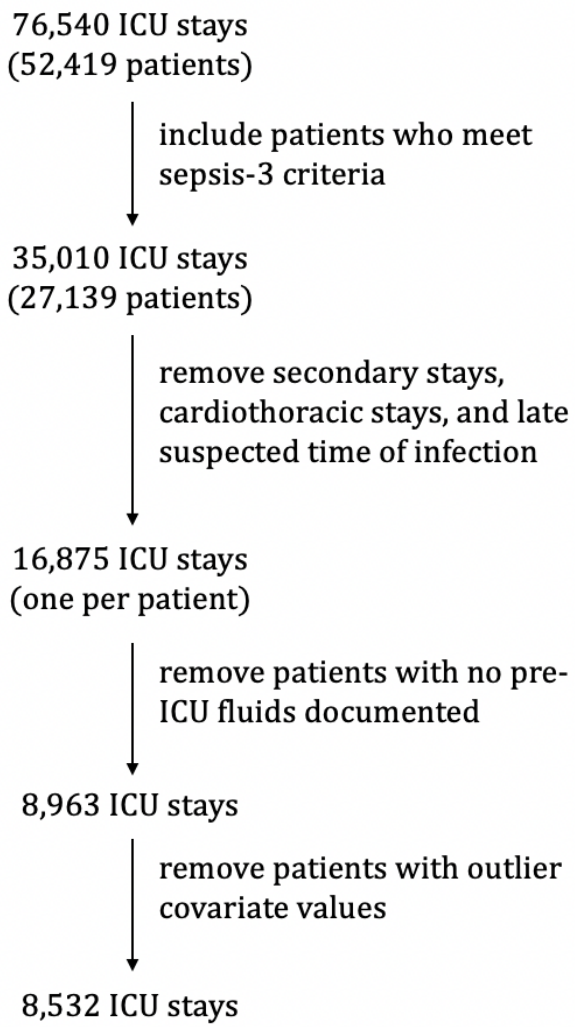

Figure 7-1: Process for constructing the sepsis cohort.

7.1.5 Interventions

Patients experiencing septic shock are typically treated with resuscitation fluids, vasopressors, or a combination of both. With fluids, the primary question of interest is the volume that should be given to the patient at each time interval, while with vasopressors, clinicians are more concerned with when to start administration. As such, we focused on varying fluid volume and vasopressor onset as the two main intervention variables in our experiments.

The fluids given to a patient can be classified as either treatment or maintenance (i.e. background). In our study, we were mainly interested in varying the amount

of treatment fluids, the list of which is provided in Table A.7. Furthermore, we specifically considered boluses in our counterfactual intervention strategies, which are fluids administered at a rate of 250mL/hr or greater [15]. To avoid potential confounding, all other fluids documented for a patient were treated as background and captured under a maintenance fluid variable separate from the bolus variable (Table A.6).

The vasopressors we considered were epinephrine, norepinephrine, dopamine, vasopressin, and phenylephrine. For the same dosage amount, different vasopressors may cause different levels of blood vessel constriction, so their rates were standardized during data extraction to allow comparability [15]. Boluses were modeled with a single variable representing the dosage amount at each timestep while vasopressors were modeled with a binary variable indicating whether treatment was given at that timestep or not.

7.1.6 Outcomes of Interest

The main outcomes of interest in our study were fluid overload and in-hospital mortality. To assess fluid overload, we identified events commonly associated with the condition, including administration of diuretics, onset of dialysis, initiation of mechanical ventilation, and chest X-ray findings of pulmonary edema. For in-hospital mortality, we directly extracted death and release indicators from the table of hospital admissions in MIMIC-IV. During training of G-Net, we directly modeled diuretics, dialysis, mechanical ventilation, and pulmonary edema as binary variables. To represent when a patient terminated their stay due to either death or release from the ICU, we modeled a generalized end-of-stay variable. We then used a separate outcome model, described in Section 8.1.3, to predict whether the patient experienced in-hospital mortality or was discharged alive. The end-of-stay variable denotes a *censoring event*: as soon as it takes on value 1 at timestep t_c, the patient is censored and no training data for that patient is available as input to our model for timesteps $t > t_c$. Consequently, during testing time, we also stopped simulating additional timesteps if the end-of-stay variable is predicted to be 1.

7.2 Extraction of Select Covariates

While most of the variables used in our study were extracted previously by the contributors of MIMIC-IV and are publicly available, there were additional covariates that we derived as part of our work. These include pre-ICU fluids, pulmonary edema labels, and the Elixhauser comorbidity score. In this section, we discuss the extraction of this data.

7.2.1 Pre-ICU Fluids

Since our intervention strategies included boluses as a treatment of interest, it was important to account for pre-ICU fluids as a potential confounder in decisions involving administration of additional fluids. In the MIMIC-IV database, we extracted pre-ICU fluids from one of two tables: *inputevents* and *eMAR* (electronic medication administration records). If patients had fluids documented in both tables, we only used data from *inputevents* since the records were most likely duplicates. From *eMAR*, only fluids administered within 72 hours prior to ICU admission were regarded as pre-ICU fluids.

In addition to treatment fluids (Table A.7) documented prior to ICU admission, we also considered fluids given in the post-anesthesia care unit (PACU) or operating room (OR) as contributing to the total pre-ICU volume. All PACU-related intake was added to the total pre-ICU volume, while OR-related intake was only counted as pre-ICU if (a) the patient did not previously come from the surgical intensive care unit (SICU) or (b) the patient previously came from the SICU but the OR intake was documented within 24 hours of ICU admission. Fluids administered in the PACU and OR include crystalloids, colloids, packed red blood cells, platelets, and fresh frozen plasma.

7.2.2 Pulmonary Edema Indicator

A common method for diagnosing the presence of pulmonary edema is via chest radiograph images, which are interpreted by radiologists in radiology reports. These

radiology reports are typically unstructured, requiring the use of advanced natural language processing (NLP) and machine learning techniques to extract meaningful information. In our study, we obtained labels for pulmonary edema status using CheXpert, a state-of-the-art labeler that automatically detects the presence of 14 different observations in radiology reports, one of which is pulmonary edema [13].

The first step for obtaining these labels was to de-identify the radiology reports in MIMIC-IV to remove personal health information using a combination of deep learning and rules-based NLP techniques. Specifically, we employed an architecture known as bidirectional encoder representations from transformers (BERT), as well as the *pydeid* module for annotating and removing personal identifiers, to accomplish this task. Afterwards, we filtered out all radiology reports associated with chest X-rays and applied the CheXpert model to these documents.

For each radiology report, CheXpert outputs either 1, 0, or -1 for each of the 14 observations, indicating presence, absence, or possible presence of that observation [13]. For our purposes, we treated both 1 and -1 as indicating presence of edema. To determine if a patient developed edema prior to their ICU stay, we considered findings from the most recent radiology report documented within 72 hours prior to ICU admission. If there was a positive finding, then the patient was considered to have pulmonary edema upon ICU admission. We chose 72 hours as a cutoff since the condition typically takes up to three days to resolve.

7.2.3 Elixhauser Score

The Elixhauser comorbidity index is a method for estimating patient comorbidity based on the International Classification of Diseases codes (ICD), specifically ICD-9 and ICD-10 [9]. A number of different conditions are factored into the calculation of the comorbidity score, including heart failure, renal failure, cancer, and obesity, though the weights for each condition differ depending on the exact algorithm used. Our study employed the van Walraven method, where a higher score indicates a more severe degree of comorbidity. The Elixhauser scores were computed for each patient using records of their diagnoses recorded in MIMIC-IV.

7.3 Data Preprocessing

Real-world data presents a number of challenges not present in traditional academic datasets used in many machine learning research applications or synthetic data generated from a simulator such as CVSim. For example, while patients produced by CVSim possess covariates documented frequently and at regular intervals, there is no such guarantee that clinical data collected from the ICU is as neat. Consequently, a number of preprocessing steps were required to format our data in a manner suitable for modeling.

7.3.1 Binning Strategy

When employing RNN frameworks for time-series prediction, observations are typically represented as sequences with fixed-width time steps. However, datasets in the real-world are often collected at different frequencies across different patients and variables and are thus seldom as neat as the ones prepared in synthetic environments. To adapt the MIMIC clinical data for sequential deep network modeling, we processed time-varying variables into discrete bins. Some measurements, such as laboratory values (e.g. creatinine, BUN, bicarbonate, etc.), are generally recorded once per day while other measurements, such as vital signs (e.g. heart rate, blood pressure, etc.), are taken once every hour. Because of this, we chose a bin size of one hour. If more than one measurement was documented in a particular hour for a particular variable, values were averaged to produce a single aggregate value.

Using this binning scheme poses an issue, however, as we lose the temporal ordering between any interventions and covariates administered and measured, respectively, in the same hourly bucket. This is important as covariate values recorded after a treatment action may reflect responses to that treatment; placing the covariate in the same bucket as the treatment would eliminate the possibility of our model recognizing any causal effects present. To circumvent this problem, the following re-binning procedure was used for covariates and outcomes recorded in the same hour h as the intervention:

1. If the covariate or outcome was documented prior to the intervention, we left the value in bucket h.

2. If the covariate or outcome was documented after the intervention, we re-assigned the value to the next bucket $h + 1$.

Note that with outcomes, we were chiefly interested in the *onset* of each outcome, as opposed to the mere presence or absence, at each hour. This is because once an outcome is deemed to have occurred, it is likely for it to continue across multiple time steps (e.g. a patient placed on mechanical ventilation is usually ventilated for more than a single hour). Thus, for outcomes, we applied the re-binning procedure only to the start times of the outcome and did not make any adjustments at subsequent hours.

As we considered two treatments, boluses and vasoporessors, in our study, it was possible for both interventions to be administered at different times in the same hour. If this occurred, we re-binned covariates on the *second* treatment. In other words, all values observed before the later treatment action (including measurements recorded between the two treatment actions) were left in bucket h, while measurements recorded after the second treatment action were pushed to bucket $h + 1$.

An example of the strategy described above is illustrated in Figure 7-2. Say there is a patient admitted to the ICU at 11:05, who has a laboratory value documented at 13:20 and a vital sign recorded at 13:50. Since the time of these measurements occurs between the second and third hours with respect to admission time (hours 2.25 and 2.75 for the laboratory and vital observations, respectively), they would be placed into hourly bin 3 provided that no interventions are administered in the same time step. Now let's introduce two treatment actions for this patient: a bolus administered at 13:11 (hour 2.1) and vasopressors administered at 13:35 (hour 2.5). In this case, we would re-bin with respect to the time of vasopressor administration. The laboratory measurement would remain in hourly bin 3 but the vital sign would be reassigned to hourly bin 4, to allow our model to learn the appropriate causal relationships between the interventions, covariates, and outcomes.

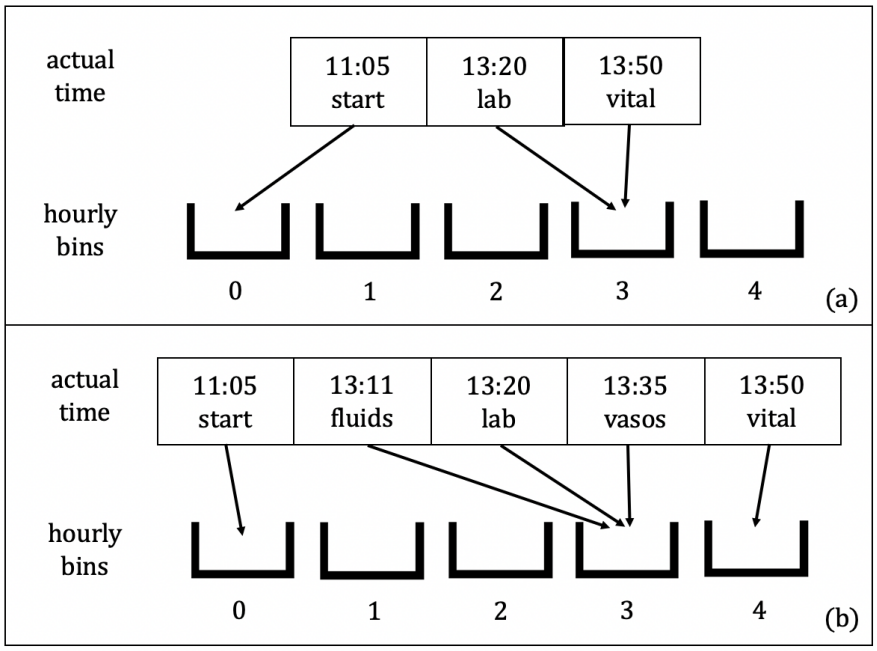

Figure 7-2: Rebinning procedure for covariates without (a) and with (b) treatment administered in the same hourly bin. In the diagram, *fluids* refer to fluid boluses while *vasos* indicate vasopressors.

7.3.2 Irregularly Sampled Data

Not all clinical data is collected frequently (i.e. at every hour) or consistently (i.e. at the same time interval), resulting in irregularly sampled data with many missing values, also known as NaNs. For a variable whose measurement in a given hour is NaN, we take the most recent measurement in a preceding hour and use that as the value for this hour, a method known as forward-filling imputation (Figure 7-3). This approach makes sense as variables are typically only measured when the clinician believes there has been a change in its value; otherwise, it is reasonable to assume that the measurement is constant. It follows, then, that variables that tend to change more frequently are also measured more frequently.

To forward-fill NaN values present at the start of the ICU stay, we used the most recent measurement in the 24 hours preceding admission. The 24-hour window was chosen since many variables, with the exception of vital signs, are recorded once per day and assumed to remain constant until the next measurement. While vital signs are measured more frequently, we did not have to worry about NaNs at the start of

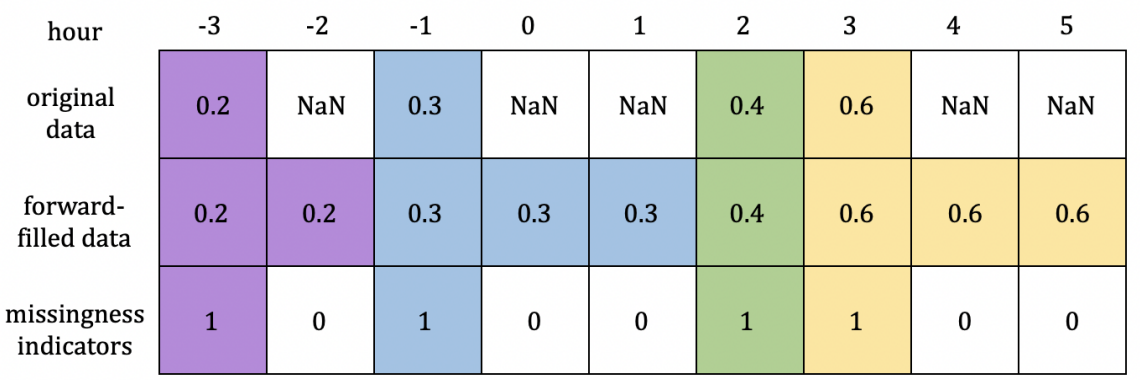

Figure 7-3: Forward-filling imputation for missing data. Only data from timestep $t = 1$ onward was used during the experiments, but values between $t = -24$ and $t = 0$, if they were present, were considered while forward-filling data at the beginning of the ICU stay.

the ICU stay because our study defined ICU admission as the time of the first heart rate recording, and it is expected that other vital signs are documented along with the heart rate. For patients who did not have any recent measurements preceding ICU admission, we simply median-filled NaN values at the beginning of the stay, using the median value of the covariate from the original (i.e. non-forward-filled) training dataset. We found that using the median yielded improved results compared to using the mean, as it is less subject to distortion by outliers.

The only values in our dataset that were not forward-filled were related to fluid balance, including boluses, maintenance fluids, and urine output. In addition, vasopressors, diuretics, dialysis, and mechanical ventilation, which were all recorded with a start and end time in MIMIC-IV, were zero-filled during hours outside of the documented time intervals.

7.3.3 Indicators for Missing Data

Recall that for many variables, clinicians record a new value only when they have reason to believe that there was a change in that variable. For example, following antibiotics treatment, a new laboratory culture may be ordered to check for decreased presence of microbes in the patient. This suggests that missing data is not missing at random, and that the pattern of documentation itself can provide information

about patient status. Previous studies showed that in addition to the forward-filling procedure described above, introducing binary indicators for each variable to represent missingness can potentially improve performance of RNNs on a multi-label sequence classification task [20]. For a given variable, its associated indicator takes on value 1 if a measurement was recorded in that hour or 0 if the value was originally NaN and required forward-filling (Figure 7-3). The model would be expected to learn a relationship between the indicator variables and their corresponding covariates, specifically that the covariate value should only change at this timestep compared to the previous if the indicator variable is predicted to be 1.

In our work, we explored a similar approach to modeling missingness as Lipton et al. [20]. We found that while introducing indicators for missingness lowered validation loss on the one-step-ahead prediction task during training, it did not improve simulation results when evaluated against a hold-out ground-truth test set. Given that adding the indicator variables nearly doubled the number of inputs to our model, it is reasonable that training performance increased simply due to an increase in predictors. Unfortunately, since each missingness indicator was incorporated as a separate additional box in the model, we hypothesize that error propagation was amplified during simulation time, leading to decreased testing performance. Because of this finding, we ultimately did not include indicators for missingness in the experiments presented in Chapter 8 on the MIMIC data.

7.3.4 Additional Data Transformations

Similar to the CVSim experiments, the continuous-valued inputs to our models were normalized as $x' = \frac{x-\bar{x}}{\sigma(x)}$, where $\bar{x}$ and $\sigma(x)$ were both derived from the training dataset. For covariates with log-normal distributions, we first took the logarithm of the values prior to calculating the mean, standard deviation, and normalized values. For covariates with log-normal distributions that could take on nonpositive values, we used the standard normalization procedure described above, as logarithms are only meaningful when applied to quantities greater than 0. Binary variables were treated as binary in the MIMIC study (as opposed to categorical in the CVSim

experiments). Finally, we introduced a nonlinear representation of time by applying a cubic spline transformation to the (unmodeled) hour covariate. This was to allow for better comparability between the linear and LSTM models.

Chapter 8

Assessing Predictive Performance of G-Net on the Sepsis Cohort

With real-world datasets, the only information we have available is that which was generated under the observational regime; we don't have access to outcomes had the clinicians followed an alternative treatment strategy. In other words, there are no "ground-truth counterfactual datasets" analogous to the CVSim datasets D_{c1} and D_{c2} against which we can compare the results of G-Net's counterfactual predictions. As such, to evaluate G-Net on the sepsis dataset, we tested its predictive abilities under the observational regime via "predictive check" experiments. Predictive checks allow us to assess how well our model estimates population-level covariate distributions under observational treatment strategies, which is necessary to gain confidence about its predictions under counterfactual strategies.

8.1 Experimental Setup

In this segment of our study, we were interested in estimating the effects of sepsis treatments administered in the first 24 hours of the ICU stay on patient prognosis later in the stay. G-Net was used to simulate forward patient covariates for up to 24 timesteps, after which patient history was fed into separate classifier models to estimate the occurrence of various outcomes of interest in the first 72 hours and in-

hospital mortality. The goal of the predictive check was to ensure that G-Net learned the correct covariate distributions in the observational cohort, which was evaluated by comparing the G-Net simulated trajectories and the predicted outcome prevalences against the ground-truth dataset.

We divided our 8,532-patient cohort into training, validation, and testing sets using an 80-10-10 split, respectively, and trained G-Net on a one-step-ahead prediction task with the training set as input and the validation set for hyperparameter tuning. During testing time, G-Net was provided with the patient's baseline physiological state at ICU admission (i.e. their covariates at timestep $t = 1$) and tasked with predicting trajectories at timesteps $t > 1$. Unlike the CVSim experiments, where the counterfactual treatment strategies were predefined, the sepsis predictive checks included treatment variables (bolus volume and vasopressor indicator) as part of the simulated covariate trajectories. This was necessary as we were evaluating the predictive ability of G-Net in these experiments.

For each patient in the testing set, we generated $M = 10$ Monte Carlo simulations according to Algorithm 1. This yielded a total of $M \times N$ simulations predicted for the testing set. Since our study focused on intervention strategies during the first 24 hours of a sepsis patient's ICU stay, the maximum length for a simulated trajectory was 24 timesteps (up to 23 simulated timesteps in addition to 1 baseline timestep), though it was possible for a patient's stay to be predicted to end before the 24-hour cutoff due to the presence of censoring variables. For patients whose stays were not predicted to end by 24 hours, we simply stopped simulating after the 24th timestep.

Following simulation, we used the forecasted patient histories to estimate the onset of various outcomes of interest within the first 72 hours of the ICU stay, in addition to in-hospital mortality. Note that for these outcomes, we were not so much interested as to *when* they occurred within the 72 hours so much as to *whether* they occurred at all. We built individual classifier models for predicting each outcome given the covariate and treatment history in the first 24 hours as input. These outcome models, described in Section 8.1.3, were trained using the same training and validation datasets as G-Net. A diagram of the high-level experimental setup is provided in Figure 8-1.

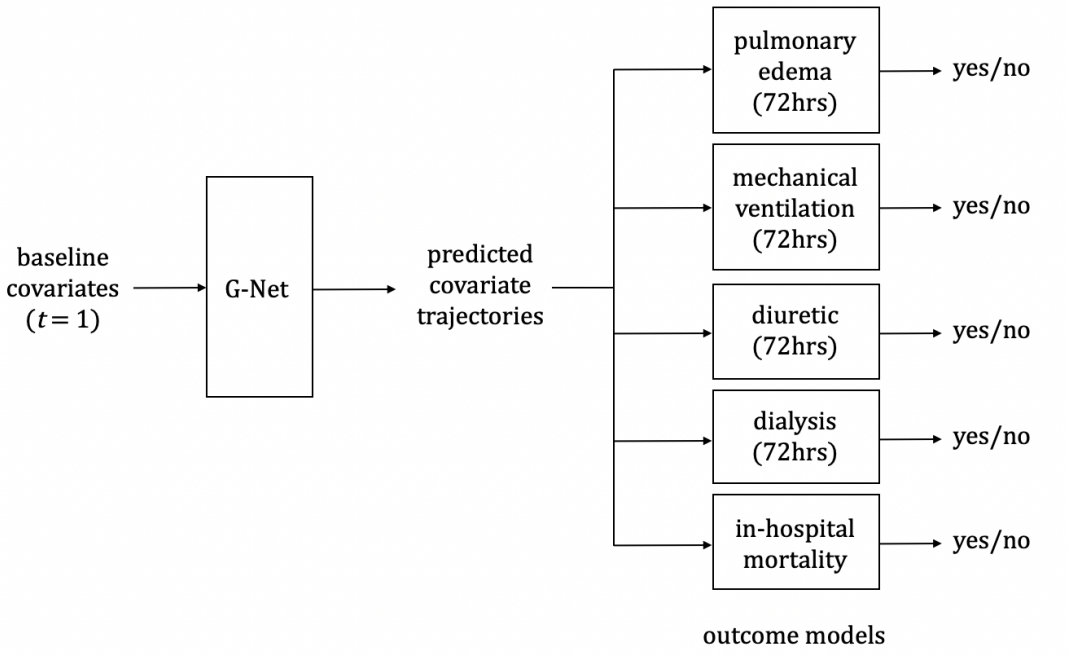

Figure 8-1: Experimental setup for the sepsis experiments. G-Net was used to simulate covariate trajectories in the first 24 hours of the ICU stay; given the covariate trajectories as input, predictions for various outcomes of interest were then produced by the individual outcome models.

8.1.1 Details on Implementation & Training of G-Net

We focused on a d-box model of G-Net for the set of predictive check experiments using the MIMIC sepsis cohort; that is, we modeled each covariate L_t^c with a separate box, an architecture known as *one variable per box*. Given the results from the CVSim experiments (as presented in Section 6.3), we opted to focus on refining (*LSTM2*), which employs LSTMs for each covariate box with no representational layer. While (*LSTM3*) demonstrated slightly lower RMSE during testing, the additional representational layer used in this implementation added a degree of time complexity that outweighed the mild performance gain provided by the model. As a baseline, we also constructed a one variable per box version of (*Linear*). Similar to the CVSim experiments, the models adapted for the sepsis study were fit to a one-step-ahead prediction task during training time and used to simulate covariates forward during testing time. However, with the lack of a shared representation layer, we were able to optimize the loss for each box individually during training, allowing G-Net to learn

the distributions of the covariates more accurately. In this chapter, we will refer to the LSTM model as ($LSTM$) and the linear model as ($Linear$).

In principle, the ordering of covariate groups should not affect model performance according to the assumptions of G-computation, but in practice, we found that changing the variable ordering could lead to differences in the simulation performance. Due to the method of covariate rebinning we employed in our study (as discussed in Section 7.3.1), the model required the covariate and outcome boxes to precede the treatment boxes in the prediction order; boxes for the censoring variables, which only included end-of-stay in our experiments, were placed last. This means, for example, that the input to the end-of-stay box at timestep t of simulation includes patient history up to and including timestep $t-1$, along with the predicted values at timestep t for the covariates, treatments, and other outcomes (see Section 4.2 for more details). The final ordering we used for our model was determined empirically and is given by the order presented in Table A.6. Static variables were inputted as predictors to each box but not modeled in our experiments.

Along with the covariate ordering, another challenge of working with real-world data was that patients in our cohort did not necessarily stay in the ICU for the same amount of time, resulting in variable-length inputs to our models. In order to support batching during training, we padded all trajectories to the same length ($t = 24$, which was the maximum length of any trajectory in our dataset) using a dummy value. Batching is helpful as it produces a more stable gradient estimate for backpropagation and also speeds up the training process. During the forward pass, we inputted in the entire padded trajectory, but only computed the loss on timesteps with actual measurements. At testing time, we simply stopped simulating once a positive instance of a censoring variable was encountered for a given patient, or once we reached timestep $t = 24$. Because of the different timesteps at which a censoring variable could be sampled, it was necessary that we simulate one patient at a time instead of in batches. Lack of batching ability also explains why we only generated $N = 10$ simulations per test patient rather than $N = 100$ as in the CVSim experiments.

8.1.2 Model & Training Parameters

The hyperparameter space we used to tune our model is displayed in Table 8.1. These values are on a per-box basis, meaning that each box in the model could have a different optimal parameter set. During training, we used the Adam optimizer with early stopping for a maximum of 100 epochs and performed the experiments on NVIDIA Tesla V100 SXM2 GPUs. The early stopping window was 10 steps and the stop tolerance was 0.001, and we employed a batch size of 32.

Table 8.1: Hyperparameter search space for G-Net in MIMIC experiments. *Denotes optimal parameter settings that were shared across all boxes in the model, where "optimal" is defined according to the criteria discussed in Section 8.2.

	Hyperparameters	**Search Range**
Linear	Learning Rate	0.001, 0.01
LSTM 2	Number of Layers	2*, 3
	Hidden Dimension	16, 32, 64, 128
	L2 Penalty	1e-4, 1e-5, 1e-6*
	Dropout (LSTM Layers)	0.0, 0.1*, 0.2
	Learning Rate	0.001*, 0.01

8.1.3 Outcome Model

Provided treatment for sepsis in the first 24 hours of the ICU stay, we were interested in predicting whether a patient experienced various adverse outcomes at any point during the first 72 hours of the stay, as well as overall in-hospital mortality. For this task, we built and trained separate models for the following outcomes: development of pulmonary edema, onset of diuretics, onset of dialysis, initiation of mechanical ventilation, and death versus release from the hospital, though death versus release was not restricted to the same 72-hour window as the other outcomes. Each model accepts as input patient history in the first 24 hours of the ICU stay and outputs a binary indicator predicting whether the patient experienced that outcome. Additionally, we imposed an arbitrary ordering on the outcomes so that the predictions made by earlier models in the ordering were included as input to later models.

The outcome models were developed using the MIMIC sepsis cohort with an identical training-validation-testing split as G-Net (as described in Section 8.1), and they were tested on the ground-truth held-out test set before being applied to the trajectories simulated by G-Net for the patients in the test set. At prediction time, the pulmonary edema, diuretics, dialysis, and mechanical ventilation indicators were automatically set to 1 if the patient had experienced the outcome during the first 24 hours of their ICU stay, and they were set to 0 if the patient had ended their stay before 24 hours without experiencing the outcome prior to leaving. When training the models for these outcomes, we also excluded patients who had already experienced that outcome in the first 24 hours. The end-of-stay indicator was handled somewhat differently from the others variables, as it indiscriminately captured both death and release during simulation and required the outcome model to predict the ultimate outcome. As such, all patients were included during training and testing of the in-hospital mortality outcome model.

The hyperparameter space used in tuning the outcome models is presented in Table 8.2. We tested both linear and RNN (using LSTM layers) implementations, with an early stopping window of 5 steps, a stop tolerance of 0.001, a batch size of 64, and an L2 penalty of 1e-6, and found that the RNN-based models yielded better performance. We also explored different orderings of the outcomes.

Table 8.2: Hyperparameter search space for outcome model *Denotes optimal parameter settings that were shared across all outcome models.

	Hyperparameters	**Search Range**
Linear	Learning Rate	0.001, 0.01
RNN	Number of Layers	2, 3
	Hidden Dimension	32, 64
	L2 Penalty	1e-6
	Dropout (LSTM Layers)	0.0*, 0.1
	Learning Rate	0.001*, 0.01

8.2 Evaluation

To evaluate the performance of G-Net, we focused on two aspects: (1) accuracy of simulated population-level covariate trajectories in comparison to ground-truth trajectories under the observational regime, and (2) similarity in predicted outcome prevalence for the simulated patients compared to the ground-truth outcome prevalence. Both procedures are described in more detail below.

8.2.1 Qualitative Analysis Using Covariate Trajectories

As in the CVSim evaluation, we plotted population-level covariate trajectories simulated for patients in the test dataset and analyzed them against the ground-truth population-level covariate trajectories. If the simulated trajectories closely follow the actual trajectories, it is reasonable to consider the model well-performing and to expect the counterfactual predictions made by the model to hold some validity. For continuous variables, we averaged the values at each timestep across patients, while for non-censoring binary variables, we calculated the proportion of patients at each timestep who had the variable set to 1. For censoring variables, we plotted a cumulative percentage of patients who had encountered a positive instance of the variable over time.

Due to the presence of these censoring variables, we noted that some patients might not have had measurements for all 24 timesteps. To address patients who ended their stay early when computing the population-level trajectories, we simply excluded them from the denominator at later timesteps so that the average covariate value at timestep t only took into account patients who were still in the ICU.

8.2.2 Quantitative Analysis Using Outcome Prevalence

The presence of censoring variables presented a challenge in quantifying the degree of error in the G-Net simulations, as it was possible for the predicted trajectory for a patient to have a different number of timesteps compared to the actual trajectory. In this case, it was unclear how to define the RMSE at timesteps in which the predicted

trajectory had already ended but the actual trajectory still reported covariate values, or vice versa. As such, we were precluded in the sepsis predictive check experiments from computing RMSE over time in the same manner as in the CVSim experiments, where all patients were observed for equal-length stays.

Outside of RMSE, an alternate proxy for quantitatively analyzing G-Net performance is to look at the proportion of outcomes of interest predicted for the simulated patients in comparison to the actual proportion of those outcomes in the test set patients. That is, for each outcome, can we train a classifier to predict whether a patient will develop that outcome given their covariate history? Additionally, if we use the trajectories simulated by G-Net as the inputted covariate history to this classifier, is the predicted percentage of patients who have this outcome the same as the actual percentage of patients in the ground-truth dataset? The proportions should be approximately equal if the simulated trajectories are accurate estimations of the actual trajectories, and this finding would provide further evidence that G-Net is able to predict average effects of treatment strategies in our given cohort of sepsis patients. This approach was suitable for our study since we were primarily interested in aggregate performance at the population level rather than individual predictions.

8.3 Results

We implemented a one variable per box model of G-Net and explored the use of GLMs and LSTMs for the boxes. No representational layer was employed in these experiments to reduce complexity. We first fit G-Net using the 6,825 patients in the training set, with the 853 patients in the validation set for hyperparameter tuning. Given observed covariate history at baseline (i.e. ICU admission at $t = 1$), we then simulated covariates forward up to $t = 24$ and predicted various outcomes of interest following simulation. The treatment variables were included as part of the simulation in order to evaluate the predictive ability of G-Net under the observational regime.

8.3.1 Population-Level Covariate Trajectories

The population-level trajectories for selected covariates is presented in Figure 8-2, which compares the predictive abilities of the (*Linear*) and (*LSTM*) implementations of G-Net applied to the sepsis cohort. Note that the values shown for bolus and maintenance fluids, as well as urine output, represent the total volume recorded or predicted at each hour. In contrast to the CVSim experiments, we found that the LSTM model did not always outperform the GLM; rather, the performance varied depending on the covariate. For example, while the trajectories for vasopressors, BUN, and heart rate predicted by (*LSTM*) are shown to be closer to ground truth than (*Linear*), the opposite is true for bicarbonate, mechanical ventilation, and SOFA score.

Observing the variability in performance between the two models, we experimented with a "hybrid" version of G-Net (hereon referred to as (*Hybrid*) in this chapter) combining the trained boxes from the (*Linear*) and (*LSTM*) implementations. This was possible given that each box was trained and optimized separately from the other boxes, so there was no practical barrier preventing us from using the output of an LSTM-based box as the input to a linear-based box, or vice versa, during simulation time. The only constraint was that the boxes in (*Linear*) and (*LSTM*) had to be ordered in the same sequence, because the input to box j at time t required the outputs of the preceding boxes $(\hat{L}_t^0, \ldots, \hat{L}_t^{j-1})$; that is to say, regardless of what model type was used for box j, the box must have been provided access to the covariates from boxes 0 to $j-1$ at time t.

To construct the hybrid model, we compared the trajectories of the (*Linear*) and (*LSTM*) predictions relative to ground-truth for each covariate and selected the best-performing box to be used in the hybrid construct (Table A.8). The resulting model employed during simulation comprised a mosaic of GLM and LSTM boxes. It should be noted that we used the simulated trajectories, rather than the one-step-ahead validation loss, to determine level of performance. For almost all covariates, the LSTM boxes achieved lower validation losses at training time compared to the

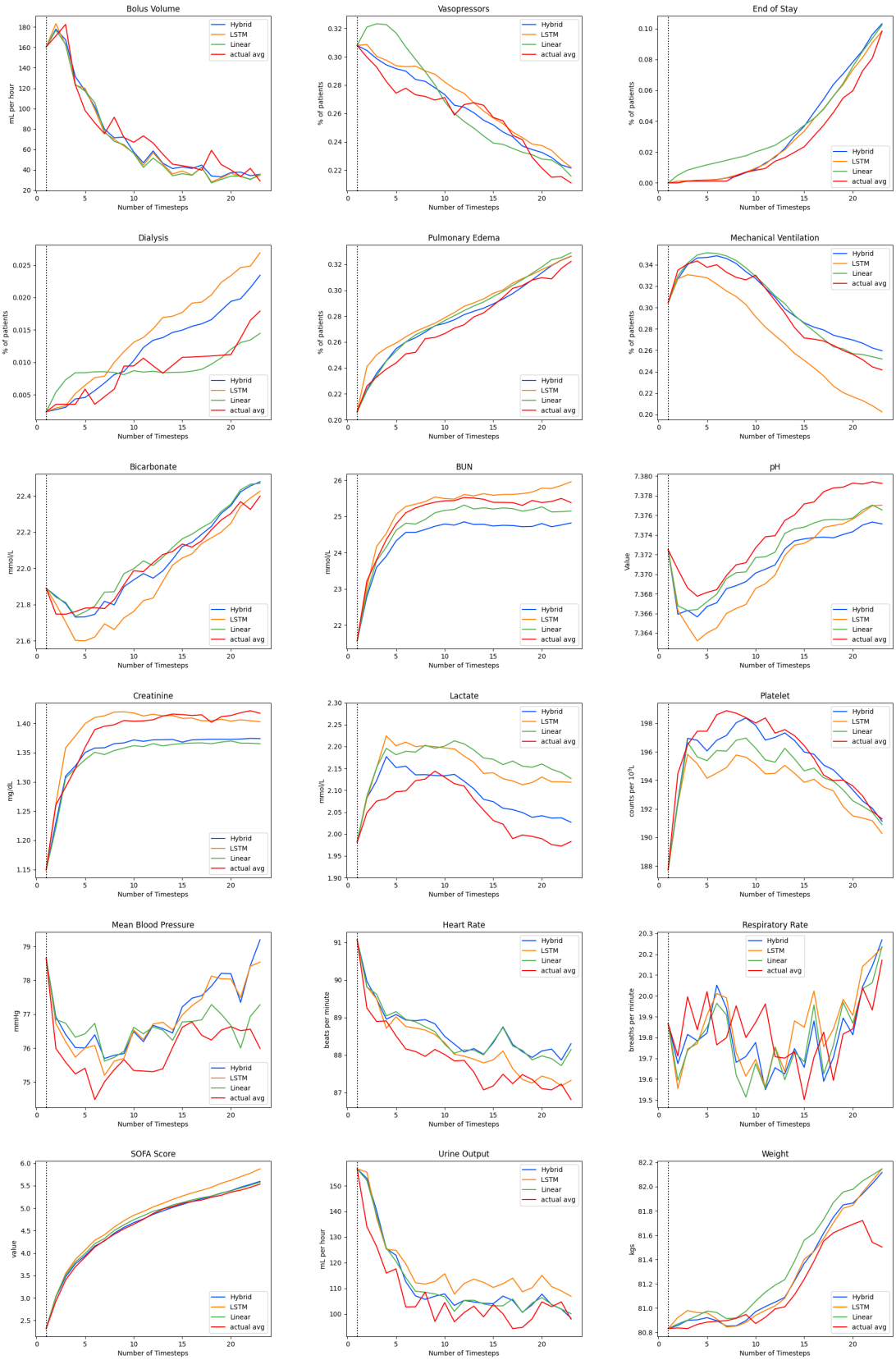

Figure 8-2: Simulated and ground-truth population-level trajectories for selected covariates in the predictive check experiments.

GLMs, but this did not necessarily translate to better performance at testing time as illustrated by the results in Figure 8-2. This finding is not unreasonable as the settings at testing time (multi-step-ahead simulation) and training time (one-step-ahead prediction) were not identical, and it is reminiscent of the results of applying MLP to the CVSim dataset, where we showed that the performance of MLP exceeded GLM during training but was considerably worse during testing.

Our rationale for the hybrid model was that combining the best-performing boxes for each covariate would help improve the overall predictions made by G-Net. Not only would this increase the accuracy of the covariate predictions individually, but recall that the inputs to box j included the outputs of boxes 0 through $j-1$; this suggests that an increase in the accuracy of the covariate predictions earlier in the simulation ordering can also help performance of covariates later in the ordering. On the flip side, we recognize that poor predictions made in prior boxes may also negatively affect predictions of downstream covariates. This is one disadvantage of the one variable per box framework, because while it allows for more robust estimation of covariate distributions, it also increases the room for error during simulation: errors that can be propagated and amplified over time as they are passed through the sequence of boxes in the model.

The predictions made by (*Hybrid*) are plotted in Figure 8-2 in blue. For the most part, we see that the hybrid implementation performs on par with or better than the all-linear or all-LSTM models, a difference that is particularly noticeable for covariates such as lactate and platelet.

8.3.2 Outcome Model Predictions

The outcome models were trained using the same training and validation datasets as G-Net and evaluated using the ground-truth trajectories from the testing dataset. Instead of purely looking at accuracy, we were interested in comparing the percentage of patients predicted to experience each outcome compared to the actual percentage of patients who experienced that outcome. Given that the patients in the testing dataset were assumed to be from the same distribution as the patients in the training and

validation datasets, we expected predictions using ground-truth trajectories from the testing dataset to be accurate estimations of population-level outcome prevalences.

The results for the best-performing outcome models, which all employ LSTMs, are presented in Table 8.3. In this case, "best-performing" was defined by the validation loss at training time, which makes sense since the tasks during training and testing were identical. Note that the order of outcomes provided in the table was empirically determined to produce the most accurate predictions.

Table 8.3: Proportion of patients in the test set experiencing in-hospital mortality and other outcomes of interest within the first 72 hours of the ICU stay. The estimated percentages (columns 3 and 4) were produced by the best-performing outcome models and the simulated trajectories were generated under the (*Hybrid*) implementation of G-Net.

Outcome	**Actual Proportion**	**Proportion Predicted from Ground-Truth Trajectories**	**Proportion Predicted from Simulated Trajectories**
Pulmonary Edema	0.459	0.447	0.462
Mechanical Ventilation	0.450	0.465	0.463
Diuretic	0.269	0.249	0.287
Dialysis	0.045	0.054	0.053
In-Hospital Mortality	0.129	0.155	0.158

Once we validated the outcome classifiers on the ground-truth test set trajectories, we applied the LSTMs to the trajectories simulated by the (*Linear*), (*LSTM*), and (*Hybrid*) models in order to estimate the proportion of patients that experienced each of the outcomes conditioned on their covariate history predicted by G-Net. If the estimated proportions are similar to the actual proportions, then we can more confidently say that the trajectories simulated by G-Net come from the same distribution as trajectories in the ground-truth dataset and that G-Net is accurately modeling the given population. Of the trajectories simulated by the different G-Net implementations, we found that the outcome percentages estimated using the hybrid-simulated trajectories most closely aligned with the actual percentages, which supports the findings in Figure 8-2 regarding the respective performances of the three models. The results of the outcome predictions are displayed in Table 8.3.

From the table shown above, we observe that onset of dialysis and in-hospital mortality appear to be moderately overestimated compared to the actual percentages seen in the ground-truth test set. However, we also observe that the proportions predicted from the ground-truth versus simulated trajectories are very similar, suggesting that these inaccuracies do not necessarily indicate poor simulation performance by G-Net, but rather suggest room for improving the architecture of the respective outcome models. In fact, the percentages predicted from the simulated trajectories are fairly similar to the percentages predicted from the ground-truth trajectories for a majority of the outcomes, with the exception of onset of diuretics where the discrepancy is larger. Even so, the predicted proportions are able to match the actual proportions to a reasonable degree. Additional analyses should be conducted to confirm there is no statistically significant difference in the predictions made using the ground-truth versus simulated trajectories.

Chapter 9

Predicting Counterfactual Treatment Effects in Sepsis Patients

Having obtained promising results from the predictive check experiments, we hypothesized that G-Net could be used to reliably estimate covariate trajectories under alternative interventions for which we do not have ground-truth data for. In this chapter, we explore two counterfactual strategies of interest and report the outcomes predicted by G-Net under these strategies.

9.1 Counterfactual Strategies

To devise interventions that were relevant, interesting, and not likely to violate the positivity assumptions required by g-computation, we considered regimes implemented by established clinical trials studying the early treatment of sepsis. Specifically, we focused on adapting interventions employed in the CLOVERS and ProCESS trials [1, 30] and developed two counterfactual strategies to test: a *conservative* and a *liberal* strategy. The conservative strategy is based on the CLOVERS study and involves conservative use of treatment fluids, while the liberal strategy is based on the ProCESS study and employs treatment fluids more liberally. Details about these trials, as well as how we modified them for our experiments, are provided below. Note that when testing the counterfactual interventions, we only borrowed the treatment

protocols without adopting the other study design criteria set forth in the CLOVERS and ProCESS trials.

9.1.1 Conservative Intervention Strategy

The conservative intervention regime defined in our study was based on the Crystalloid Liberal or Vasopressors Early Resuscitation in Sepsis (CLOVERS) clinical trial, which compared the use of restrictive versus liberal fluids in the treatment of hypotensive patients with suspected sepsis infection during the first 24 hours of their ICU stay [30]. In the restrictive fluids intervention arm, patients were treated with only vasopressors to achieve a mean arterial pressure between 65mmHg and 75mmHg, while in the liberal fluids intervention arm, patients were administered fluid boluses to increase blood pressure. Additional details about the inclusion/exclusion criteria, outcome measures, and treatment strategies can be found on the study website [30]. All patients admitted into the CLOVERS study were required to have received a minimum of 1L and a maximum of 3L of fluid prior to enrollment.

With clinical guidance, we modified the CLOVERS restrictive fluids arm for our study as follows (Figure 9-1): for a patient with blood pressure below 65mmHg at time t, a 500mL bolus was administered if the total volume of fluids (including both treatment and maintenance) they received up until that time point did not exceed X liters and if they did not have any signs of fluid overload; otherwise, vasopressors were administered. While $X = 1$ in the CLOVERS study, we tested values $X = 1, 3$ and 5 in our experiments and we did not exclude any patients from our cohort during simulation if their pre-ICU fluid level was above or below X. Patients who received more than X liters of pre-ICU fluids were simply not given additional fluids during treatment, while patients who received less than X liters of pre-ICU fluids were administered boluses until they reached or surpassed X liters, after which they were switched to vasopressors. Fluid overload was defined in our study as the presence of pulmonary edema, administration of diuretics without mechanical ventilation, onset of dialysis, or initiation of mechanical ventilation [30]. Maintenance fluids were always set to 0 as per the CLOVERS guidelines.

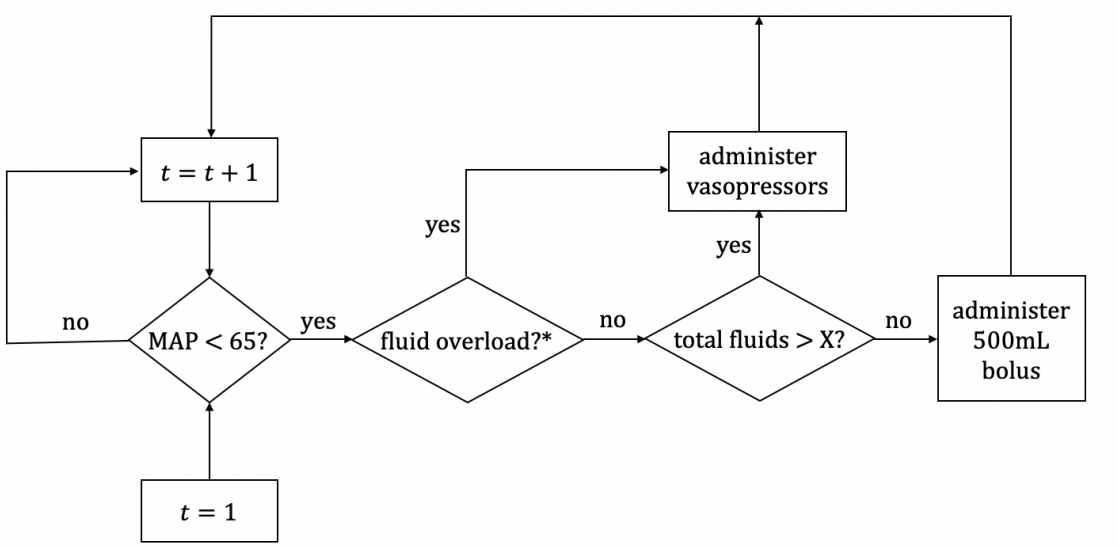

Figure 9-1: Procedure for the conservative counterfactual regime. *Fluid overload is defined as one of the following: pulmonary edema, diuretics without mechanical ventilation, dialysis, or mechanical ventilation.

9.1.2 Liberal Intervention Strategy

The liberal intervention strategy was developed using the Protocol-Based Care for Early Septic Shock (ProCESS) trial as a guide. The investigators of this study were interested in examining various protocol-based strategies for sepsis treatment, including the so-called protocol-based "standard strategy," which involves more liberal use of fluids than the CLOVERS restrictive fluid intervention arm [1]. All patients admitted into the study had received a minimum of 2L of fluids, or were given additional fluids to achieve the 2L baseline unless they were determined to be fluid overloaded by the treating clinician. Patients with systolic blood pressure below 100mmHg were treated with 500-1000mL fluid boluses if they were not fluid overloaded and with vasopressors if they were. More information about the trial can be referenced in the publication by the ProCESS investigators [1].

While the ProCESS standard strategy was only defined for the first 6 hours of sepsis treatment, we extended it to 24 hours for our experiments. Similar to the trial, patients in our study with systolic blood pressure below 100mmHg were treated with 1L fluid boluses if they did not display evidence of fluid overload and vasopressors

if they did. At each timestep, treatment was administered if systolic blood pressure was below 100mmHg; otherwise 250mL maintenance fluids were administered. No maintenance fluids were administered during timesteps in which boluses or vasopressors were used or if the patient was fluid overloaded. During the first timestep in which a patient required treatment, if they had not yet received the 2L minimum and did not present signs of fluid overload, they were provided with a 1L fluid bolus or a larger quantity if needed to reach the 2L threshold. Fluid overload under the liberal counterfactual strategy was defined similarly as the conservative treatment regime. The protocol is outlined in Figure 9-2.

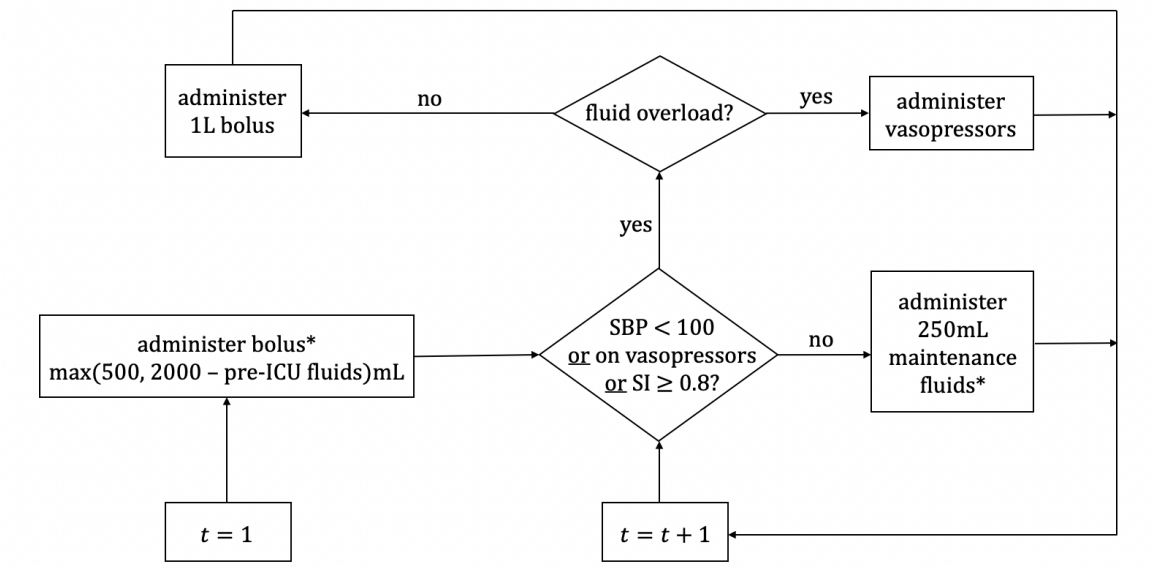

Figure 9-2: Procedure for the liberal counterfactual regime. *Administered only if the patient is not fluid overloaded, defined in the same manner as in the conservative regime.

9.2 Evaluation

Due to the lack of a ground-truth dataset to compare to, we relied on clinical expertise to determine if the predicted covariate trajectories averaged over the population were physiologically plausible and logically consistent with the different counterfactual strategies. For example, we might expect the incidence of pulmonary edema to

be higher under the modified ProCESS strategy compared to CLOVERS because the former employs treatment fluids more liberally than the latter. Note that the treatment variables (treatment bolus and vasopressor onset) are no longer being predicted under the counterfactual experiments, as they are computed based on the strategies of interest. However, we can also plot the computed treatment trajectories and compare them to the observational treatment trajectories to check that the treatments are being administered correctly. That is, we would expect the average amount of fluid bolus administered to be greater under the liberal regime and smaller under the conservative regime than the observational regime; furthermore, the prevalence of vasopressors would be expected to be greater under the conservative regime than the liberal regime.

9.3 Results

We used the (*Hybrid*) version of G-Net to simulate covariate trajectories for patients in the testing set under the conservative and liberal strategies, in order to estimate the effects of these strategies on downstream outcomes. Select covariates are presented in Figure 9-3, allowing for comparison of the two counterfactual regimes, where $X = 1$ in the conservative regime. In order to interpret these results, we must assume that there is no unobserved confounding; while an inherent limitation of observational studies is the inability to guarantee adjustment for all confounders, we believe that we accounted for the most import variables that drive fluid and vasopressor treatment decisions in sepsis.

Looking at the trajectories given in Figure 9-3, we note that the volume of fluids administered is much higher under the liberal counterfactual regime than either the observational or conservative counterfactual regimes. Furthermore, the volume is highest at the first timestep, which reflects patients receiving enough fluid to reach the 2L minimum required by the protocol. At later timesteps, the bolus amount under the liberal strategy continues to exceed the observational, while the amount under the conservative strategy remains below the observational. This pattern is also

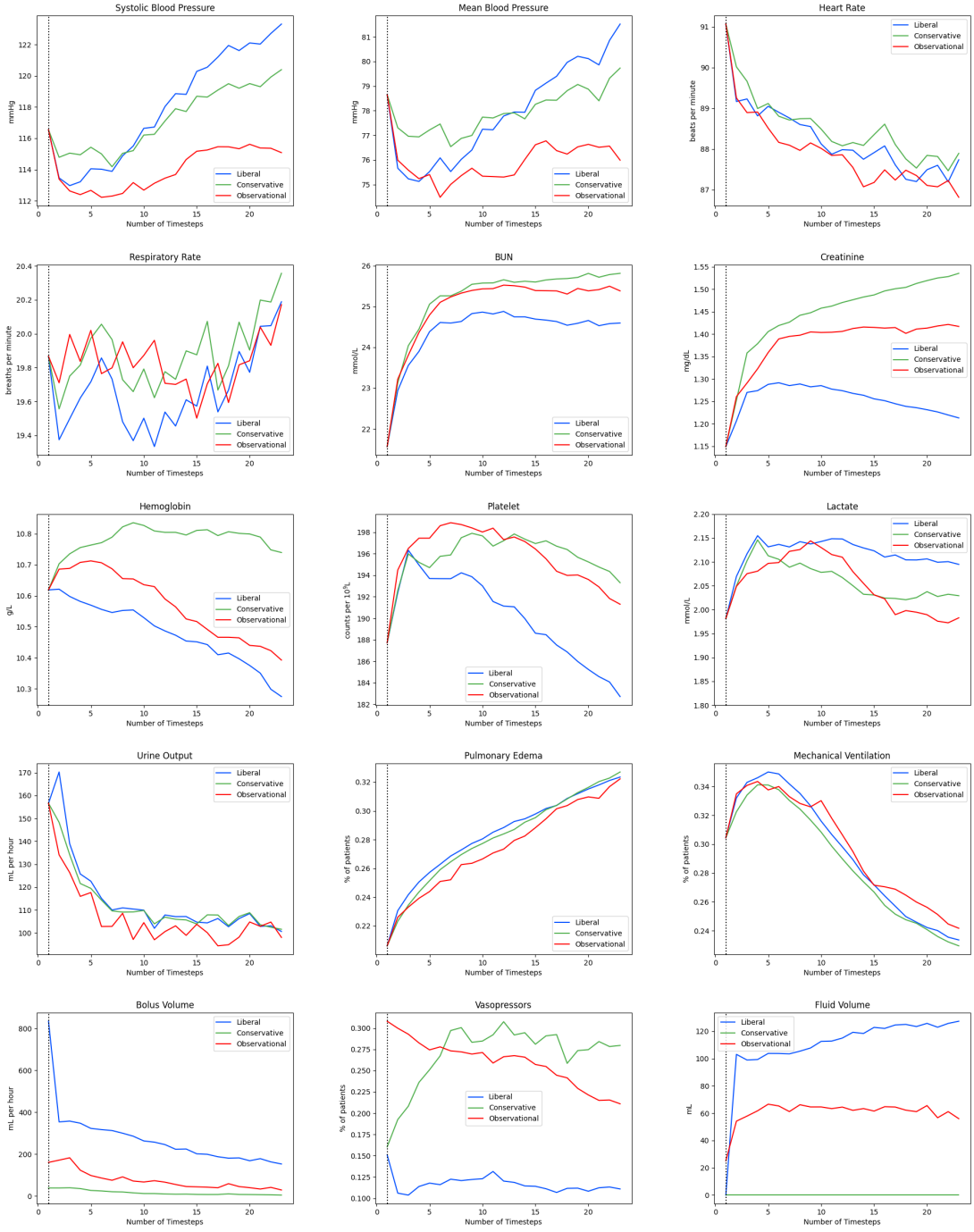

Figure 9-3: Population-level trajectories for selected covariates predicted under counterfactual treatment regimes, with the exception of bolus volume, maintenance fluids, and vasopressor indicator, which are calculated deterministically. Ground-truth covariate trajectories under the observational regime are also plotted for reference. Note that the results displayed here for the conservative regime use a fluid cap of $X = 1$ liter.

seen with maintenance fluids, which are not used in the conservative strategy and administered more generously in the liberal strategy.

The percentage of patients on vasopressors, meanwhile, is generally higher under the conservative strategy than the observational, which is in turn higher than the liberal. Notably, there are fewer patients on vasopressors at earlier timesteps under the conservative strategy than the observational, indicating that patients may not have yet reached the fluid cap and are still being treated with boluses; however, the level of vasopressors quickly increases and remains high for the rest of the simulation. These observations are expected based on how we defined the two counterfactual interventions in Section 9.1 and suggest that the interventions were implemented correctly during simulation.

Turning to the predicted trajectories, we see that both counterfactual regimes are able to maintain blood pressure at comparable levels above the observational regime, with the liberal regime achieving slightly higher blood pressure at later timesteps in the simulation. The effects of the alternative interventions on heart rate and respiratory rate are similar to the observational; similarly, there is not much difference in lactate levels and urine output between the three strategies.

With respect to laboratory values, the levels of hemoglobin, BUN (blood nitrogen urea), and creatinine are noticeably lower under the liberal regime than the conservative, and somewhat lower under the liberal regime than the observational. This may be explained by the fact that administering large amounts of resuscitation fluids increases the bloodstream fluid volume, which in turn decreases the concentration of various substances in the body [21]. It follows, then, that limiting the use of fluids in the conservative regime may increase the apparent levels of these substances compared to the observational. Additionally, it should be noted that acute increase in BUN and creatinine are both markers for acute renal failure [12]. These compounds are normally eliminated from the body by the kidneys, but hypotension can lead to decreased renal perfusion and reduced renal filtration. Treatment for sepsis is expected to reverse these effects by increasing blood pressure, leading to greater waste clearance. The decreased levels of BUN and creatinine under the liberal regime sug-

gests that administering resuscitation fluids could potentially restore kidney function more effectively than either the observational or conservative interventions.

Interestingly, the predicted trajectories for pulmonary edema and mechanical ventilation appear to be similar under the conservative and liberal strategies, and even slightly higher than the observational. Initially, this seems somewhat surprising as these outcomes are markers for fluid overload, and we would have expected them to be more common under a treatment regime employing liberal fluids, such as the modified ProCESS intervention. One possible hypothesis for this observation is that the downstream effects of increased fluid administration are not manifested until later in the ICU stay, after the first 24 hours of treatment. Indeed, once we obtained the simulated trajectories under the counterfactual regimes, we applied the outcome models to these trajectories and found that markers of fluid overload, as well as in-hospital mortality, were predicted to occur more frequently under the liberal regime compared to the conservative regime (Table 9.1).

Table 9.1: Predicted prevalence of various outcomes of interest under the conservative and liberal counterfactual strategies. The predicted prevalences using simulated trajectories under the observational regime are also provided for reference.

Outcome	**Observational**	**Conservative**	**Liberal**
Pulmonary Edema	0.462	0.445	0.535
Mechanical Ventilation	0.463	0.450	0.474
Diuretic	0.287	0.299	0.311
Dialysis	0.053	0.054	0.069
In-Hospital Mortality	0.158	0.147	0.160

The most prominent difference in predicted outcome prevalence between the two counterfactual regimes is seen pulmonary edema, which is estimated to develop in a larger proportion of patients under the liberal regime compared to the conservative. This is reasonable given that greater volumes of fluid are administered under the liberal regime while urine output is predicted to be approximately equal across strategies; together, these observations lead to a net fluid balance that is much higher in patients receiving the liberal intervention [22].

Also consistent with the prediction of increased pulmonary edema is the fact that the percentages of patients initiating mechanical ventilation, diuretics, and/or dialysis are also somewhat increased under the liberal regime in comparison to the observational; on the other hand, the percentages under the conservative regime are similar to or slightly less than the observational. A mild decrease in in-hospital mortality is seen under the conservative strategy, which aligns with previous studies reporting that placing caps on resuscitation fluids in the early treatment of sepsis may help improve patient prognosis [31]. Moreover, there appears to be no difference in the death rate between the observational and liberal counterfactual strategies, which is a finding also reported by the ProCESS study [1].

9.4 Limitations

In the future, additional research may be undertaken to explore other counterfactual regimes and refine the ones presented in this thesis. While we based the conservative and liberal strategies on existing clinical trials (CLOVERS and ProCESS, respectively), it was necessary to simplify and modify the protocols to some extent to fit into our study design and modeling framework. For example, we operationalized fluid overload as the presence of pulmonary edema, the onset of diuretics or dialysis, or the initiation of mechanical ventilation, as those were the covariates we were able to extract from the MIMIC database. The ProCESS and CLOVERS trials, on the other hand, also included observations like jugular venous distention, rales, and/or bilateral crackles in their criteria. To increase the validity of our models for real-world applications, it is important that we find ways to model counterfactual strategies as closely and accurately as possible as to how they might be implemented in actual clinical settings.

Another limitation of our study pertains to the interpretation of the population-level covariate trajectories in Figure 9-3. For a given covariate, the differences between trajectories under the two counterfactual regimes do not strictly represent causal effects, because these differences depend on who still remains in the ICU at each

timestep. Consider the average decrease in platelet levels seen in patients receiving the liberal treatment versus the conservative. With high probability, the set of patients left in the ICU at timestep $t > 1$ who were receiving the liberal treatment was not the same set of patients left in the ICU at timestep $t > 1$ who were receiving the conservative treatment. Because we were no longer comparing the same exact cohort of patients at each timestep after baseline, the observed differences between the two populations could have been due to characteristics of the patients themselves rather than a causal effect of the intervention strategy used.

On the other hand, this isn't to say that the plots in Figure 9-3 are unable to provide any insight into patient outcomes under alternative intervention strategies. If the rate of end-of-stay in the first 24 hours is low under the different treatment scenarios, then these plots may still reasonably approximate the effects of counterfactual regimes on the covariates and outcomes shown. This is because a low rate of end-of-stay implies that the patient population at the end of 24 hours is largely the same as the patient population at the start, and since we started with the same cohort of patients admitted to the ICU under each counterfactual regime, it follows that the cohort of patients remaining in the ICU after the follow-up period must also be similar across regimes. Even in this case, however, it still must be noted that the observed effect isn't necessarily a causal one. Ultimately, prospective studies and randomized clinical trials will be required to obtain more conclusive results and further clarify the appropriate dosage and timing of treatment. In the meantime, the work presented here provides an instructive example for how g-computation can potentially be used to support clinical decision-making and explore multiple treatment strategies efficiently, perhaps to inform future experimental studies.

Chapter 10

Conclusion

Treating sepsis is challenging due to the heterogeneity in patient responses and the potential for developing adverse outcomes elicited by the very interventions (i.e. fluids and vasopressors) used to stabilize the condition. In real-world clinical settings, physicians can only observe the set of outcomes associated with the treatment that they actually administered to the patient and thus do not have access to outcomes that might have happened had they taken a different course of action. Given the difficulty in determining optimal interventions for sepsis patients, it would be useful to be able to test various interventions before selecting one to administer. This is precisely the goal of counterfactual prediction, where we aim to estimate the effects of alternative treatment regimes on patient covariate trajectories provided information under the observational regime. While there are many methods that can be used to carry out this task, our study focused on g-computation, as it is particularly suited for handling inputs that are high dimensional under interventions that are dynamic and time-varying.

In this thesis, we introduced G-Net, a flexible recurrent neural network approach to g-computation for estimating outcomes under counterfactual treatment strategies. The model was first evaluated using synthetic data generated from a well-established program simulating the cardiovascular system, from which we could obtain both an observational and multiple counterfactual datasets for training and testing, respectively. During our experiments, we explored a number of different architectures and

showed that LSTMs yielded more accurate population-level counterfactual predictions compared to GLM and MLP baselines. Notably, the advantage of LSTMs increased as the number of simulation timesteps increased, illustrating their ability to learn complex relationships between time-varying covariates and capture their long-range dependencies.

After validating the performance of G-Net on synthetic data, we adapted one of the LSTM-based architectures for analyzing fluid and vasopressor treatment strategies in a real-world cohort of sepsis patients. It was necessary to modify the model to address issues with missing and irregularly sampled values seen in real-world datasets. Given the lack of counterfactual information, we assessed the predictive abilities of G-Net on a held-out test set under the observational regime and demonstrated that the model is able to accurately simulate forward most covariates at testing time. The simulated trajectories also led to predicted outcome probabilities that were similar to the ground-truth probabilities. When we applied G-Net to analyzing outcomes under counterfactual treatment strategies derived from protocols used in widely known clinical trials studying the early treatment of sepsis, we found that G-Net made logical and clinically plausible predictions on covariate trajectories.

Future extensions of this work may be interested in employing other architectures for G-Net such as time-series generative adversarial networks, or adding temporal attention mechanisms to improve performance on datasets where the covariate dependencies span across larger ranges of time. Performance may also be improved by modifying the training procedure so that the task at training time versus testing time are more similar. In our study, G-Net was trained on a one-step-ahead prediction task but evaluated on its ability to generate multi-step-ahead simulations, and we hypothesize that incorporating the simulation loss during model optimization can help boost testing accuracy. Given that neural networks require a significant volume of data for adequate learning, it may also be useful to acquire a larger cohort for future experiments. Finally, we note that the distribution of the Monte Carlo simulations produced by G-Net in this work constitute an estimate of uncertainty about a counterfactual prediction; however, our current architecture does not provide uncertainty

estimation about the G-Net parameter estimates themselves. In the future, we may want to investigate techniques for incorporating model uncertainty in counterfactual outcome prediction.

We hope that the work presented in this thesis can ultimately help improve the treatment outcomes of sepsis patients in the ICU. Using simulated data from a mechanistic model, we successfully demonstrated the ability for G-Net in estimating counterfactual outcomes under alternative treatment strategies, with improved qualitative and quantitative results over previous models. While we primarily focused on clinical applications in our work, we recognize that G-Net is a powerful tool that can be adapted for a variety of scenarios in which one would like to predict downstream effects of alternative courses of action.

Appendix A

Tables

Table A.1: Input parameters to CVSim and their corresponding ranges. *Cannot be lower than zero-pressure filling volume.

Input Covariate	**Range**
Zero-Pressure Filling Volume	500 - 3,500
Total Blood Volume*	500 - 6,500
Nominal Heart Rate	40 - 160
Total Peripheral Resistance	0.1 - 1.3
Arterial Compliance	0.4 - 1.1
Pulmonary Arterial Compliance	2.0 - 3.4
Pulmonary Microcirculation Resistance	0.4 - 1.00

Table A.2: Output parameters of CVSim. Covariates in bold and denoted with an asterisk (*) are the covariates used in the G-Net experiments.

Output Covariate	Abbreviation
Left Ventricle Pressure*	**LVP**
Left Ventricle Flow*	**LVQ**
Left Ventricle Volume	LVV
Left Ventricle Contractility*	**LVC**
Right Ventricle Pressure*	**RVP**
Right Ventricle Flow*	**RVQ**
Right Ventricle Volume	RVV
Right Ventricle Contractility*	**RVC**
Central Venous Pressure*	**CVP**
Central Venous Flow	CVQ
Central Venous Volume	CVV
Arterial Pressure*	**AP**
Arterial Flow*	**AQ**
Arterial Volume	AV
Pulmonary Arterial Pressure	PAP
Pulmonary Arterial Flow	PAQ
Pulmonary Arterial Volume	PAV
Pulmonary Edema*	**PE**
Pulmonary Venous Pressure	PVP
Pulmonary Venous Flow	PVQ
Pulmonary Venous Volume*	**PVV**
Heart Rate*	**HR**
Arteriolar Resistance*	**AR**
Venous Tone*	**VT**
Total Blood Volume*	**TBV**
Intra-thoracic Pressure	PTH
Mean Arterial Pressure*	**MAP**
Systolic Blood Pressure*	**SBP**
Diastolic Blood Pressure	DBP

Table A.3: Sepsis-3 cohort characteristics. *Denotes proportion of patients released from the ICU in the first 24 hours (not from the hospital).

Characteristic	***n***
Number of ICU stays	8,532
Mean age	65.26
Median age	67.0
Male	4,683
Race – white	5,184
Race – black	698
Race – other	2,453
Death rate (first 24hrs)	1.75%
Release rate* (first 24hrs)	11.74%
In-hospital mortality	13.68%

Table A.4: Outlier values for specific covariates used to exclude patients at baseline in the sepsis experiments.

Covariate	**Maximum Value**	**Units**
Heart rate	250	beats/min
Diastolic Blood Pressure	200	mmHg
Systolic Blood Pressure	250	mmHg
Mean Blood Pressure	220	mmHg
BUN	100	mmol/L
Weight	200	kgs
pCO2	150	mmHg
Urine output	8,000	mL
Pre-ICU Fluid Amount	10,000	mL
Platelet	1500	counts/10^9L
pO2	600	mmHg
Base excess	50	mmol/L
Calcium	80	mg/dL

Table A.5: MIMIC static variables. All variables were used as inputs to our models.

Variable Name	**Variable Type**	**Units**
Age	Continuous	years
Gender	Binary	N/A
Pre-ICU Fluid Amount	Continuous	mL
Elixhauser Score	Continuous	N/A
End Stage Renal Failure	Binary	N/A
Congestive Heart Failure	Binary	N/A

Table A.6: MIMIC time-varying variables. All variables were used as inputs to our models, and boluses and vasopressors were also intervention variables. *Refers to maintenance fluids (not an intervention).

Variable Name	**Variable Type**	**Units**
Heart Rate	Continuous	beats/min
Diastolic Blood Pressure	Continuous	mmHg
Systolic Blood Pressure	Continuous	mmHg
Mean Blood Pressure	Continuous	mmHg
Temperature	Continuous	°C
SOFA Score	Treated as Continuous	N/A
Platelet	Continuous	counts/10^9L
Hemoglobin	Continuous	g/dL
Calcium	Continuous	mg/dL
BUN	Continuous	mmol/L
Creatinine	Continuous	mg/dL
Bicarbonate	Continuous	mmol/L
Lactate	Continuous	mmol/L
pO2	Continuous	mmHg
sO2	Continuous	%
spO2	Continuous	%
pCO2	Continuous	mmHg
Total CO2	Continuous	mEq/L
pH	Continuous	Numerical[1,14]
Base excess	Continuous	mmol/L
Weight	Continuous	kgs
Respiratory Rate	Continuous	breaths/min
Fluid Volume*	Continuous	mL
Urine Output	Continuous	mL
Pulmonary Edema Indicator	Binary	N/A
Diuretics Indicator	Binary	N/A
Dialysis Indicator	Binary	N/A
Mechanical Ventilation Indicator	Binary	N/A
Bolus Volume	Continuous	mL
Vasopressor Indicator	Binary	N/A
End of Stay Indicator	Binary	N/A

Table A.7: Resuscitation fluids employed in the treatment of sepsis patients.

Fluid	**Category**
Albumin 5%	Colloids
Albumin 25%	Colloids
Hetastarch (Hespan) 6%	Colloids
Dextran 40	Colloids
Dextran 70	Colloids
NaCl 0.9%	Crystalloid Bolus
NaCl 0.45%	Crystalloid Bolus
NaCl 3% (Hypertonic Saline)	Crystalloid Bolus
Lactate Ringer (LR)	Crystalloid Bolus
D5 1/2NS	Crystalloid Bolus
D5 1/4NS	Crystalloid Bolus
D5N5	Crystalloid Bolus
D5LR	Crystalloid Bolus
Fresh Frozen Plasma	FFP Transfusion
PACU FFP Intake	FFP Transfusion
Packed Red Blood Cells	RBC Transfusion
PACU Packed RBC Intake	RBC Transfusion
Platelets	RBC Transfusion
PACU Platelet Intake	RBC Transfusion

Table A.8: Boxes used to model individual covariates in the hybrid implementation of G-Net. The ordering of boxes within the hybrid model is the same as in Table A.6.

LSTM	**Linear**
Heart Rate	Fluid Volume
Diastolic Blood Pressure	Urine output
Systolic Blood Pressure	SOFA Score
Mean Blood Pressure	Temperature
Lactate	Platelet
pH	pO2
Hemoglobin	Respiratory Rate
Calcium	BUN
Base Excess	Bicarbonate
pCO2	spO2
Total CO2	Pulmonary Edema Indicator
Creatinine	Diuretic Indicator
Weight	Mechanical Ventilation Indicator
sO2	
Dialysis Indicator	
Vasopressor Indicator	
Bolus Volume	
End of Stay Indicator	

Bibliography

[1] A randomized trial of protocol-based care for early septic shock. *New England Journal of Medicine*, 370(18):1683–1693, 2014. PMID: 24635773.

[2] M.A. Alaa, M. Weisz, and M. van der Schaar. Deep counterfactual networks with propensity-dropout. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[3] O. Atan, J. Jordan, and M. Van der Schaar. Deep-treat: Learning optimal personalized treatments from observational data using neural networks. In *Proceedings of AAAI*, 2018.

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

[5] Centers for Disease Control and Prevention (CDC). Sepsis: Clinical information.

[6] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3504–3512. Curran Associates, Inc., 2016.

[7] M.W. Dünser, J. Takala, A. Brunauer, and J. Bakker. Re-thinking resuscitation: leaving blood pressure cosmetics behind and moving forward to permissive hypotension and a tissue perfusion-based approach. *Critical Care*, 17(5):326, October 2013.

[8] C.J. Espinosa-Almanza, O. Sanabria-Rodríguez, I. Riaño-Forero, and E. Toro-Trujillo. Fluid overload in patients with septic shock and lactate clearance as a therapeutic goal: a retrospective cohort study. *Revista Brasileira de Terapia Intensiva*, 32(1):99–107, January-March 2020.

[9] A. Gasparini. Comorbidity scores.

[10] T. Heldt, R. Mukkamala, G.B. Moody, and R.G. Mark. CVSim: An open-source cardiovascular simulator for teaching and research. *Open Pacing, Electrophysiology & Therapy Journal*, 3:45–54, 2010.

[11] M.A. Hernan and J.M. Robins. *Causal inference: What if*. Chapman & Hall CRC, first edition, 2011.

[12] International Society of Nephrology. Kdigo clinical practice guideline for acute kidney injury. *Official Journal of the International Society of Nephrology*, 2(1), March 2012.

[13] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, and et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 580–597, July 2018.

[14] A.E.W. Johnson, J. Aboab, J.D. Raffa, T.J. Pollard, R.O. Deliberato, L.A. Celi, and D.J. Stone. A comparative analysis of sepsis identification methods in an electronic database. *Critical Care Medicine*, 46(4):494–499, April 2018.

[15] A.E.W. Johnson, L. Bulgarelli, T.J. Pollard, S. Horng, L.A. Celi, and R. Mark. Mimic-iv (version 0.4). *PhysioNet*, 2020.

[16] R. Kato and M.R. Pinsky. Personalizing blood pressure management in septic shock. *Annals of Intensive Care*, 5(1):41, December 2015.

[17] H.I. Kuttab, J.D. Lykins, K. Hughes, M.D. amd Wroblewski, E.P. Keast, O. Kukoyi, J.A. Kopec, S. Hall, and M.A. Ward. Evaluation and predictors of fluid resuscitation in patients with severe sepsis and septic shock. *Critical Care Medicine*, 47(11):1582–1590, November 2019.

[18] R. Li, S. Shahn, J. Li, M. Lu, P. Chakraborty, D. Sow, M. Ghalwash, and L.H. Lehman. G-net: A deep learning approach to g-computation for counterfactual outcome prediction under dynamic treatment regimes, 2020.

[19] B. Lim, A. Alaa, and M. van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[20] Z.C. Lipton, D.C. Kale, and R. Wetzel. Directly modeling missing data in sequences with rnns. In F. Doshi-Velez, J. Fackler, D. Kale, B. Wallace, and J. Wiens, editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 253–270, Northeastern University, Boston, MA, USA, August 2016. PMLR.

[21] M.J. Maiden, M.E. Finnis, S. Peake, S. McRae, A. Delaney, M. Bailey, and R. Bellomo. Haemoglobin concentration and volume of intravenous fluids in septic shock in the arise trial. *Critical Care*, 22(118), May 2018.

[22] M.L.N.G Malbrain, N. Van Regenmortel, B. Saugel, B. De Tavernier, P.J. Van Gaal, O. Joannes-Boyau, and et al. Principles of fluid management and stewardship in septic shock: it is time to consider the four d's and the four phases of fluid therapy. *Ann Intensive Care*, 8(1):66, 2018.

[23] L. Mayaud. *Prediction of mortality in septic patients with hypotension.* PhD dissertation, University of Oxford, Department of Engineering Science, January 2014.

[24] A.I. Naimi, S.R. Cole, and E.H. Kennedy. An introduction to g methods. *International Journal of Epidemiology*, 46(2):756–762, December 2017.

[25] C.J. Paoli, M.A. Reynolds, M. Sinha, M. Gitlin, and E. Crouser. Epidemiology and costs of sepsis in the united states—an analysis based on timing of diagnosis and severity level. *Critical Care Medicine*, 46(12):1889–1897, December 2018.

[26] A. Rhodes, L.E. Evans, W. Alhazzani, M.M. Levy, M. Antonelli, R. Ferrer, and et al. Surviving sepsis campaign: International guidelines for management of sepsis and septic shock: 2016. *Critical Care Medicine*, 2017.

[27] E. Rivers, B. Nguyen, S. Havstad, J. Ressler, A. Muzzin, B. Knoblich, B. Peterson, M. Tomlanovich, and Early Goal-Directed Therapy Collaborative Group. Early goal-directed therapy in the treatment of severe sepsis and septic shock. *New England Journal of Medicine*, 345(19):1368–1377, November 2001.

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

[29] P. Schulam and S. Saria. Reliable decision support using counterfactual models. In *Neural Information Processing Systems (NIPS).*, 2017.

[30] W.H. Self, M.W. Semler, R. Bellomo, S.M. Brown, B.P. deBoisblanc, and June) Exline M.C. (2018, February 2020. Crystalloid liberal or vasopressors early resuscitation in sepsis (clovers).

[31] Z. Shahn, N.I. Shapiro, P.D. Tyler, D. Talmor, and L. H. Lehman. Fluid-limiting treatment strategies among sepsis patients in the icu: a retrospective causal analysis. *Critical Care*, 24(62), February 2020.

[32] M. Singer, C.S. Deutschman, C.W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, and et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801–810, February 2016.

[33] S.L. Taubman, J.M. Robins, M.A. Mittleman, and M.A. Hernán. Intervening on risk factors for coronary heart disease: an application of the parametric g-formula. *International Journal of Epidemiology*, 38(6):1599–1611, December 2009.

[34] N. Tomašev, X. Glorot, J.W. Rae, M. Zielinski, H. Askham, A. Saraiva, and et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572(7767):116, 2019.

[35] C. Xiao, E. Choi, and J. Sun. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428, 06 2018.

[36] Y. Xu, Y. Xu, and S. Saria. A Bayesian Nonparametric Approach for Estimating Individualized Treatment-Response Curves. In *Proceedings of the 1st Machine Learning for Healthcare Conference (PLMR)*, volume 56, pages 282–300, 2016.

[37] J. Yoon, J. Jordan, and M. Van der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *ICLR*, 2018.