Personalized Treatment Response Prediction under Dynamic and Time-Varying Treatment Strategies for Sepsis Patients

by

Megan Su

S.B. in Computer Science and Engineering, Mathematics, Massachusetts Institute of Technology (2022)

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

February 2024

Authored by:	Megan Su Department of Electrical Engineering and Computer Science February 5, 2024
Certified by:	Roger G. Mark Distinguished Professor, Thesis Supervisor
Certified by:	Li-wei H. Lehman Research Scientist, Thesis Supervisor
Accepted by:	Katrina LaCurts Senior Lecturer Chair, Master of Engineering Thesis Committee

Personalized Treatment Response Prediction under Dynamic and Time-Varying Treatment Strategies for Sepsis Patients

by

Megan Su

Submitted to the Department of Electrical Engineering and Computer Science
on February 5, 2024 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 medical emergency in which the body responds improperly to an infection, and is typically treated with intravanous fluids and vasopressors. However, administering the right balance is often difficult because adverse outcomes can be caused by both excessive and insufficient treatment. There have been many clinical trials done in the past to investigate the optimal regime for treating sepsis, however these studies have resulted in inconclusive results and often take a long time to conduct. Thus, personalized treatment response prediction under dynamic time-varying treatment strategies can be a very useful tool for clinicians when deciding what treatment strategy to administer to a patient.

This thesis builds on G-Net, a deep sequential modeling framework for g-computation that has been evaluated on response prediction under dynamic and time-varying strategies on the population level. Utilizing real-world data collected from the intensive care unit (ICU), we evaluate the performance of various deep learning implementations of G-Net on individual-level response prediction and compare their performances on prediction under the observational treatment regime. We then apply G-Net to counterfactual prediction under alternative regimes of interest and show that G-Net is able predict patient covariates and outcomes that are physiologically plausible and match clinical intuition. Our work showcases the potential of G-Net as a tool for personalized treatment response prediction to aid clinicians in determining optimal therapy for sepsis patients in the ICU.

Thesis supervisor: Roger G. Mark
Title: Distinguished Professor

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

Acknowledgments

First, I would like to thank my direct supervisor Li-Wei Lehman for providing guidance and direction for not only the project, but more importantly my identity as a researcher. Coming in with little research experience, she taught me from ground up how to ask the right questions, tackle difficult problems, and interpret my results beyond the surface level. She has been a wonderful mentor and helped me grow during my MEng career.

I would also like to thank my supervisor Professor Roger Mark for allowing me to be a part of the Laboratory for Computational Physiology. He has inspired me to think more deeply about the context of the research I am doing.

In addition, I would like to say thank you to Professor Zach Shahn and Dr. Elias Baedorf-Kassis. Professor Shahn has helped me better understand and addressed my questions regarding the causal inference portion of project. Having no background in medical studies, Dr. Elias Baedorf-Kassis provided much insight on the clinical setting of the project and what implications the results have clinically.

Also a huge thank you to Stephanie Hu, who was extremely helpful in my onboarding process and infinitely patient with all the questions I had about the G-Net repository and previous work done despite her busy schedule. I also would like to mention and thank Hong Xiong and Feng Wu, with whom I've had the pleasure of working with during my MEng journey.

Finally, I am so grateful for my friends and family for all the support they've given me throughout my journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sepsis is a life-threatening medical emergency in which the body responds improperly to an infection, and is typically treated with a balance between intravanous fluids and vasopressors. However finding the correct balance can be difficult. Vasopressors are a type of drug that constrict the patient's blood vessels to raise blood pressure, but clinicians tend not to favor its usage due to its aggressive nature. On the other hand, fluids are typically the default treatment strategy, but over-administration of fluids can lead to other adverse outcomes such as pulmonary edema. Recent clinical trials have shown no significant difference in mortality rate when comparing fluid-liberal and fluid-conservative treatment strategies, making it difficult to decide what treatment strategy to use for a patient.

Furthermore, in the real world, we can only observe the patient conditions and patient outcomes under the treatment that was administered, known as the observational regime. One cannot know what would have occurred if an alternative treatment strategy was chosen instead, but we often wonder what may have happened. This is especially true for clinicians who are deciding what treatment plan to use for a patient and don't have the luxury of testing the different plans before their decision, such as clinicians who work with sepsis patients. This is because sepsis patients often react differently to the same treatment, so being able to predict an particular patient's response to different regimes would be extremely useful. Thus, predicting sepsis patient response to treatment strategies is both an important area of study and the focus of our work.

The rest of this chapter will detail g-computation, a causal inference approach for counterfactual prediction, describe G-Net, a deep-learning implementation of g-computation that this work builds on, list the individual contributions made in this work, and outline of the rest of this thesis.

1.1 Counterfactual Prediction and G-computation

Counterfactual prediction is the task of predicting patient conditions and outcomes under an alternative treatment strategy given observed patient history. In our work, we use patient covariate trajectories to represent patient history. Sepsis patients often require complex treatment strategies that require multiple treatment decisions dependent on past history at any given time. These kinds of treatments are known as *dynamic* and *time-varying* treatment

strategies. *Time-varying* refers to treatments that make decisions at different times while *dynamic* refers to treatments where the treatment administered at each time depends on past history. Counterfactual prediction is especially difficult for dynamic and time-varying interventions due to time dependent confounders, when interventions depend on time-varying covariates which are in turn influenced by past treatments.

Oftentimes, sepsis patients need to be administered large amounts of fluid to increase blood pressure, but administering too much fluids can lead to fluid overload and other adverse effects. Thus, fluids are balanced with vasopressor administration, which can be a very aggressive form of therapy. Treatment plans for sepsis patients will most likely require deciding whether to give the patient fluids at any given time, and how much to give depending on their observed history, making these interventions time-varying and dynamic.

Thus, we employ g-computation, a causal inference technique used to estimate counterfactual outcomes under dynamic and time-varying treatments[1]. G-computation first learns the distribution of patient covariates, such as heart rate and blood pressure, conditioned on the patient's past history of covariates and treatments, then estimates patient counterfactual outcomes by simulating forward using the learned distribution under counterfactual treatment strategies. One question that remains is how exactly we can learn these distributions. This leads us to G-Net, a deep learning implementation of g-computation.

1.2 G-Net

Given the high predictive power machine learning and deep learning techniques have demonstrated in the past, such tools are a perfect fit for implementing g-computation. Specifically, we can use machine learning and deep learning to learn the conditional distributions of patient covariates and predict them forward.

Proposed by Li et al., G-Net implements g-computation using recurrent neural networks (RNNs) to learn the conditional distributions used to simulate forward[2]. RNNs are designed to capture sequential or temporal dependencies for sequential or time-series data, which reflects conditioning on the patient's past history. Li et al. proposed a *two-box* architecture, in which the continuous variables were jointly modeled with one neural network and the categorical variables were jointly modeled with another.

A recent thesis by Hu further refined the original work on G-Net. Hu additionally introduced the *one-variable-per-box* architecture, in which each covariate was modeled individually by its own model, and adapted G-Net for prediction using real-world data. However Hu's evaluations focused on estimating treatment effects on the population level for clinical data [3]. Our work is more focused on evaluation of predicting time-varying treatment response on the individual level for clinical data.

1.3 Contributions

Our work builds on previous works from Li et al. and Hu [2], [3] by doing a more rigorous evaluation of G-Net for prediction on the individual level. In our study, we investigated a few more model architectures for implementing G-Net. We then tested and validated

its performance via the observational check by assessing how well it was able to predict the observed care. Finally, we evaluated its performance on the task of counterfactual prediction. Thus the contribution of this thesis are the following:

1. **Exploring different model architectures for G-Net.** We introduced new model architectures used to implement G-Net. Specifically, while including the generalized linear models (GLMs) and Long-Short Term Memory (LSTM) models used in Hu's work[3], we additionally included basic recurrent neural networks (RNNs), RNNs implemented with gated recurrent units (GRUs), and a hybrid model that had a mix of one of these four architectures.

2. **Evaluation of G-Net for personalized treatment response.** We more rigorously evaluated G-Net on individual-level predictions for real-world data by introducing new measures to quantitatively assess G-Net's performance in personalized prediction under observational treatment regimes. Specifically, we investigate G-Net's performance when varying-length of patient's history was used as input to the model in predicting future outcomes of individual patients. We introduce an RMSE metric to measure the model predictive performance for continuous variables over time, while accounting for the over- and under-prediction in the trajectory length between the actual and the predicted due to death or discharge. Furthermore, we evaluate G-Net's performance in predicting the presence of various adverse clinical outcomes at an individual patient level using AUC scores as the evaluation metric.

3. **Counterfactual prediction for sepsis patients.** We use G-Net to predict sepsis patients' response under alternative fluids-limiting strategies, with varying degrees of fluids caps on a patient's total volume of bolus received during the first 24-hours in the ICU. Our fluids-limiting strategies are adapted from the recently conducted randomized controlled trial, Crystalloid Liberal or Vasopressors Early Resuscitation in Sepsis (CLOVERS)[4]. Finally, we present an individual case study from the MIMIC-IV [5]database to illustrate the potential use of G-Net for personalized treatment response prediction on the individual level.

1.4 Thesis Outline

The next chapter will give an overview of recent works using g-computation and deep learning techniques for counterfactual prediction. The following chapter will give some more background on g-computation. Chapter 4 will describe our methods and Chapter 5, will then describe our experiments, including the implementation and training of G-Net and an overview of the MIMIC dataset. Chapter 6 will present the results of our predictive check experiments, and Chapter 7 will present our counterfactual prediction experiments and results. We will conclude with a summary of our results and discussion for future work in Chapter 8.

Chapter 2

Related Works

What regime is optimal for treating sepsis patients has been a question of interest for clinicians working with sepsis patients for years. Patients often display heterogeneous reactions to similar treatment plans, making it often difficult to decide which treatment strategy would be optimal for a given patient. A recent clinical trial found no statistically significant difference in patient mortality between a more restrictive fluid strategy compared to a more liberal fluid strategy [6]. More clinical trials need to be conducted in order to potentially discover optimal treatment strategies for patients suffering from sepsis, but clinical trials can often be lengthy, and conclusions can only be drawn for the specific treatment strategies investigated.

A natural tool to turn to is machine learning and deep learning, techniques known for their predictive capabilities. In particular with the rise of RNN-style architectures that allow for capturing temporal dependencies, there has been much interest in applying various neural networks on the task of estimating treatment effects of different treatment strategies. This chapter will review some past work that has been done on investigating counterfactual prediction using machine learning.

2.1 Machine Learning for Counterfactual Prediction

There has been much work done on the application of machine learning for counterfactual prediction. In 2020, Bica et al. proposed the Counterfactual Recurrent Network (CRN) that built treatment-invariant representations of a patient's history that was then used for counterfactual prediction. These representations, built via domain adversarial training, was designed to tackle bias from time-varying confounders [7]. Prior to Bica et al.'s work, Lim et al. demonstrated that marginal structural models implemented with RNNs (known as recurrent MSMs) performed better than linear models and traditional MSM approaches on the task of counterfactual prediction [8]. Other recent works by Atan et al., Alaa et al., and Yoon et al. have applied deep learning to counterfactual prediction given data under the observed regime [9]–[11].

Although these techniques show promising results for counterfactual prediction, some are limited to counterfactual prediction for static time-varying treatment strategies. In particular, we are interested in *dynamic* treatment strategies as well. Additionally, MSMs rely

on different modeling assumptions compared to G-computation, allowing G-computation to handle higher-dimensional patient history. Additionally, G-computation allows for estimating the distribution of counterfactual outcomes using a time-varying counterfactual strategy, which is not as easily done using MSMs.

Thus this work turns to G-computation, and specifically continues the work of Hu's thesis that expanded on Li et al.'s work on on G-Net, a recurrent network implementation of G-computation for dynamic and time-varying treatment regimes [2], [3] .

2.2 G-computation and G-Net

The G-computation algorithm, described more in detail in section 3.1, learns conditional covariate distributions (conditioned on patient history) observed in the data via regression models. Previous studies have typically used generalized linear models (GLMs) as the regression model of choice, but theoretically, more complex models can be used as well. Previously, work by Shulam et al. implemented a continuous time version of G-computation using Gaussian processes (GPs), although they focused on static time-varying counterfactual strategies when designing their model [12]. Xu et al. also used GPs for individual patient-level treatment response predictions, but they limited their evaluation to predictive checks and did not evaluate model performance on counterfactual predictions [13]. Additionally, GPs are not very scalable with the size of the data and has high time-complexity with an increase in the variables used and modeled. Recurrent neural networks (RNNs) are much more scalable to higher-dimensional data, and have shown high performance on various time-series regression tasks [14], [15].

Thus, we build on G-Net, a sequential deep learning implementation of G-computation used to estimate the conditional distribution of covariates conditioned on patient history under dynamic, time-varying treatment strategies [2]. However, their work was only evaluated on synthetic datasets. Although Hu's work adapted G-Net for a real-world dataset, Hu primarily focused on evaluation on the population level rather than the individual level[3]. This work focuses on G-Net evaluation on real-world data at the individual level.

Chapter 3

Problem Setup and G-computation

Counterfactual prediction is particularly useful in situations where a decision needs to be made under uncertain circumstances by predicting the effects of each possible action. Depending on the circumstance, there are many methods that could be used for counterfactual prediction. This work in particular focuses on counterfactual prediction for sepsis patients, requiring the method of choice to be able to handle dynamic and time-varying regimes that is conditioned on high-dimensional patient covariate histories. This makes g-computation, one such method for counterfactual prediction, a perfect candidate for our setting.

G-computation is a causal inference approach to modelling counterfactual prediction on dynamic and time-varying strategies[1]. The g-computation framework consists of two primary steps. The first step is learning the conditional distributions of covariates conditioned on the history of covariate values and treatment actions. The second step then estimates counterfactual outcomes by using Monte Carlo methods to simulate forward in time from the learned distributions.

This chapter will first explain the G-Computation framework in more detail then discuss how it can be applied to our setting.

3.1 Problem Setup

We want to measure the effect of different treatment regimes g on a set of outcomes in a setting where a set of patient covariates influence both the treatment administered and the outcomes experienced by the patient. Our strategies of interest are dynamic and time-varying, meaning the treatment administered at any time step t depends on a patient's history of covariates and treatment actions up until $t-1$. G-computation can used to predict the effects of multiple counterfactual strategies of interest by simulating their effects on both the patient's covariates and outcomes.

Let us define:

- $t \in \{0, \ldots, K\}$: a discrete-valued time step, with K being the end of followup. This work uses $K = 24$.
- A_t: the observed treatment action at time t

- Y_t: the observed value of the outcome(s) at time t
- L_t: 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 $X_0, \ldots, X_t$ (for example, $\bar{L}_t$ would denote patient covariate history up to and including timestep t)

We denote a dynamic, time-varying treatment strategy g as a collection of functions $g = \{g_0, \ldots, g_K\}$, such that g_t maps patient history, a combination of past covariate values and treatment actions, to 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 prior to the intervention administered at time t. In other words, the treatment given at time step t will depend on the patient's past covariate values up to and including time step t as well as the sequence of treatment actions up to and including time step $t-1$.

Given a strategy g, we denote $Y_t(g)$ as the patient outcomes observed at time t as a result of following treatment strategy g from baseline [16]. For example, let us consider a patient who has had $m-1$ time steps of observed treatment actions $\bar{A}_{m-1}$. We would like to predict the effects of administering a strategy of interest g starting from time step m until K. We then denote the counterfactual outcome at time t as $Y_t(\bar{A}_{m-1}, \underline{g}_m)$, where $t \geq m$.

The goal of counterfactual prediction in our setting is estimating the expected counterfactual outcomes for a patient

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

conditioned on the patient's observed covariate history through time m and observed treatment action history through time $m-1$, under a specified counterfactual treatment strategy g. It is additionally 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 $t \geq m$. Notice that we obtain the expectation and distribution respectively over the population if we do not condition on the patient's history $(\bar{L}_m, \bar{A}_{m-1})$ in Equations 3.1 and 3.2.

3.2 G-computation Framework

Using the problem setup described above, we now define the G-Formula and the assumptions used in g-computation.

3.2.1 G-Formula

In order to estimate Equations 3.1 and 3.2 via G-computation, some assumptions must hold [16]. Specifically:

- **Consistency:** the counterfactual outcome is the same as the observed outcome when the counterfactual regime is the observed regime.

$$Y_t(\bar{A}_t) = Y_t \text{ for } t \in \{0, \ldots, K\}$$

- **Sequential Exchangeability:** there is no unobserved confounding of the treatment at any time. In other words, anything that would factor into the treatment decision is observed at every hour.

- **Positivity:** the counterfactual strategy g has a non-zero probability of actually being administered. This is to avoid the situation of conditioning on events with zero probability.

If these assumptions hold true, we can rewrite Equation 3.2 as

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

for $t = m$. In other words, the conditional distribution of the counterfactual outcome is equivalent to the conditional distribution of the observed outcome given a patient's history and the counterfactual treatment strategy in question.

As $t > m$ increases, Equation 3.3 becomes more complicated with the need to adjust for time-varying confounding. Let us denote $X_i, \ldots, X_j$ for any arbitrary variable X as $X_{i:j}$. Then under the assumptions above, the g-formula yields:

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

This equation captures four components that the outcomes at time $t > m$ depends on:

- the observed patient covariate history (for time steps up to m).

- the observed treatment actions (under the observational treatment strategy) up to and including time step $m - 1$.

- the counterfactual treatment actions from time step m onwards.

- the estimated effects of the counterfactual regime on the patient's covariates starting from time step $m + 1$ onwards.

3.2.2 G-computation Method

Due to difficulties solving for the closed form of the integral in Equation 3.4, in practice we use Monte Carlo simulations as an approximation instead. Specifically, we first simulate a population under the counterfactual regime of interest. Then we use the empirical outcome distribution as an estimation of the actual outcome distribution.

In order to simulate a trajectory for a patient under a counterfactual regime of interest, we sample from the joint distribution $p(L_t|\bar{L}_{t-1}, \bar{A}_{t-1})$ for each timestep $t \in \{m, \ldots, K\}$. We then do this process n times to obtain the simulated population, from which the empirical distributions of the outcomes gives us a way to approximate Equation 3.2 at each time step t. The average of the sample draws for each time step t gives us an estimate of the conditional expectations from Equation 3.1. We can then use the averages as predictions for $Y_t(\bar{A}_{m-1}, \underline{g}_m)$.

Chapter 4

Methods

To utilize the g-computation method, we first need to learn the conditional distribution $p(L_t|\bar{L}_{t-1}\bar{A}_{t-1})$ of the patient covariates at each time t dependent on the patient's history. Then we can use that learned distribution to simulates forward the data to estimate the counterfactual outcomes under different regimes. The G-Net framework leverages the power of sequential deep learning models to carry out the g-computation method [2]. In particular, RNN-style models were the choice of model having shown superior performance on problems involving high-dimensional sequential data, which is a fit for the time-varying covariate data we use.

In order to perform g-computation, G-Net works in two primary stages. First, through training, we fit our networks of choice to enable us to sample from $p(L_t|\bar{L}_{t-1}\bar{A}_{t-1})$. Then, we simulate forward according to the method described in Section 3.2.2 by using our trained model to sample the covariates at each time step. The rest of this chapter will explain the G-Net framework, describe the experiments we conducted and our evaluation metrics, and introduce our dataset of choice.

4.1 G-Net Framework

The first stage in g-computation is learning the covariate conditional distributions. When learning these conditional distribution, we can separate the vector of covariates L_t into p disjoint groups, where each group can be separately modelled. Specifically, if we set $p = 1$, we would model all the covariates simultaneously and approximate the covariates' joint distribution. If we set $p > 1$, we can model covariates separately. This can be desirable in situations where the distributions of the covariates may be different, such as if the data contains both continuous covariates and categorical covariates. In our study, we set p to be d, our total number of covariates of interest. In other words, for each covariate, we have a box that models its conditional distribution. We call this approach the *one-variable-per-box* approach.

Due to our choice of the one-variable-per-box architecture, we can train and optimize each box independently of one another. This also means at each given time step, we can impose an (arbitrary) ordering in which the covariates are to be predicted. Then at time t, for the covariate j in the prediction order, we can condition on all covariates up until time

$t-1$ and all covariates until $j-1$ at time t. Namely, we learn $p(L_t^j|\bar{L}_{t-1}\bar{A}_{t-1}L_t^0 \dots L_t^{j-1})$ for each covariate. Then, we can use the probability identity

$$\begin{aligned} p(L_t|\bar{L}_{t-1}\bar{A}_{t-1}) =& p(L_t^0|\bar{L}_{t-1}\bar{A}_{t-1}) \times p(L_t^1|\bar{L}_{t-1}\bar{A}_{t-1}L_t^0) \\ & \times \cdots \times p(L_t^d|\bar{L}_{t-1}\bar{A}_{t-1}L_{t\,t}^{0\,d-1}) \end{aligned}$$

to simulate our final $p(L_t|\bar{L}_{t-1}\bar{A}_{t-1})$. Specifically, we can simulate our final predicted $p(L_t|\bar{L}_{t-1}\bar{A}_{t-1})$ by simulating our covariates in the arbitrary order we impose on them during training.

4.2 G-Net Architecture

At each time t, G-Net is trained to predicted L_t, the covariates at time t conditioned on $\bar{L}_{t-1}\bar{A}_{t-1}$, the patient's covariate history and treatment history. In the setting described in Section 4.1, if we set $p=1$, we would have 1 box $f^0(;\Lambda^0)$ with parameters Λ^0 learned during training and input $(\bar{L}_{t-1}\bar{A}_{t-1})$.

Then for any arbitrary p, we would have p boxes to model each p groups of covariates. Let us use the convention L_t^i to denote the prediction of box i. Then for the first box $f^0(;\Lambda^0)$, the input is just $\bar{L}_{t-1}\bar{A}_{t-1}$, the patient's covariate and action history. The output L_t^0 is then also used as input to the next box. Then for arbitrary box i, $L_t^i = f^i(\bar{L}_{t-1}\bar{A}_{t-1}\hat{L_t^0} \dots \hat{L_t^{i-1}};\Lambda^i)$. In other words, the input to box i is the concatenation of the patient's history with the predictions from the previous $i-1$ boxes.

There are many configurations possible to implement G-Net with. This work focuses on the one-variable-per-box implementation, where we set $p=d$. As such, each covariate distribution is learned and modeled by its own box.

4.2.1 G-Net Training

G-Net is fit to the one-step-ahead prediction task, which will provide us with estimates of $E[L_t^j|\bar{L}_{t-1}\bar{A}_{t-1}L_{t\,t}^{0\,j-1}]$ for all t and j, the conditional expectations of covariate j at time t. Then using these conditional distributions, we can simulate from $p(L_t^j|\bar{L}_{t-1}\bar{A}_{t-1}L_{t\,t}^{0\,j-1})$ as described in Section 3.2. The following sections will discuss choices made during the training process.

Teacher-Forcing Strategy

We trained G-Net on the one-step-ahead prediction task using a *teacher-forcing* strategy. Specifically, at each time step t for box j in the arbitrary ordering, we input the ground truth for all covariates up until time $t-1$ as well as the ground truth for all covariates up until $j-1$ at time t. This was chosen in contrast to the *student-forcing* strategy, where we input the ground truth for all covariates up until time $t-1$ and the **predicted values** for all covariates up until $j-1$ at time t, to learn the conditional distributions that exist in the observed data.

Loss Function

As in classic machine learning, optimizing the right choice of loss function and employing effective gradient descent techniques can enable efficient training of our model to predict the covariates at each time t. We choose to optimize each box individually, rather than jointly, to allow for greater flexibility while training. Additionally, we choose to average over all time steps over all patients per batch to reduce bias related to a variable number of time steps. We choose binary cross-entropy loss (BCE) for binary variables and mean squared error (MSE) for continuous variables.

G-Net Simulation

At each time step for each covariate, we simulate $L_t^j | L_t^0 \dots L_t^{j-1} \bar{L}_{t-1} \bar{A}_{t-1} \sim \hat{E}[L_t^j | L_t^0 \dots L_t^{j-1} \bar{L}_{t-1} \bar{A}_{t-1}] + \epsilon_t^j$, where $\hat{E}[L_t^j | L_t^0 \dots L_t^{j-1} \bar{L}_{t-1} \bar{A}_{t-1}]$ is our model's prediction. If covariate j is continuous, we choose ϵ_t^j to be randomly drawn from an empirical set of residuals $L_t^j - \hat{L}_t^j$ that are generated from our validation dataset. If the covariate is binary, we instead take a Bernoulli draw characterized by the output of our model, assuming the ouput of our model is in the range $[0, 1]$. This can be done via an activation function, such as the sigmoid activation.

For each simulation, we simulate forward until the patient has a full 24-hour trajectory, or until the patient is predicted to have ended their stay. This occurs when death or release is predicted for the patient of interest.

4.3 Experimental Overview

We want to perform personalised outcome prediction for sepsis patients. In particular, we want to perform counterfactual prediction on the individual level for sepsis patients under alternative fluid therapies. However, the effects of fluid treatment do not always emerge immediately. Oftentimes, outcomes due to fluid therapies, such as pulmonary edema, are observed a period of time after the initial treatment.

Thus we first apply the G-Net framework on our cohort of interest to predict forward a patient's covariates and outcomes experienced until 24 hours post ICU admission, as mentioned in the previous section. If the patient is predicted to have already experienced the outcomes of interest in the 24 hour study period, we do not need to predict further.

If the patient has not been predicted to have experienced certain outcomes, which may be due to the delayed onset of outcomes due to fluid therapies, we want to then predict whether or not a patient will ultimately experience those outcomes.

4.3.1 72 Hour Outcome Prediction Framework

For each outcome of interest, we want a binary classifier $h^j(;\Gamma^j)$ with learned parameters Γ^j to predict whether a patient will experience that outcome of interest within 72 hours of ICU admission given the patient's 24 hour covariate trajectories. Specifically, we would like to know Y^i, the binary outcome label representing whether a patient experiences outcome i, conditioned on $\bar{L}_{24} \bar{A}_{24}$. We impose an almost arbitrary ordering for prediction of the 72

hour outcomes - the only rule we impose is that release and mortality are predicted last in no particular order. Similarly to the 24-hour case, input to each 72-hour model for outcome j is the 24-hour patient covariate and treatment action history $\bar{L}_{24}\bar{A}_{24}$ as well as the binary outcome labels prior to j in the ordering, $Y^0 \dots Y^{j-1}$. As with the 24-hour case, we use the teacher-forcing strategy and BCE loss to train the 72 hour outcome binary classifiers.

4.4 Evaluation

To choose the best-performing model for our counterfactual analysis and to showcase the reliability of our models, we need to choose relevant and meaningful metrics of evaluation. In general, we were interested in measuring how well our models were able to predict and simulate forward the conditional covariate distributions, as well as how well our models were able to predict patient outcomes. Thus we chose two quantitative measures for evaluating our models.

4.4.1 Individual-Level RMSE

Our first evaluation measure is the Root Mean Squared Error (RMSE) on the normalized values, used as a quantitative measure of the difference between our predicted covariate trajectories and the ground truth covariate trajectories. This metric was applied solely to our assess our G-Net framework for the 24 hour period.

We start by averaging the N Monte Carlo simulations per covariate per patient to consolidate our simulations into a representation of the patient's final predicted covariate trajectories. Thus for each patient, we have a mean $\bar{L}_t^j$. We then use the RMSE formula, with N_c as the number of patients and f being our model,

$$\sqrt{\frac{1}{N_c(K-m)d}\sum_{i=1}^{N_c}\sum_{t=m}^{K}\sum_{h=1}^{d}(L_{ti}^{h,CF} - \hat{L}_{ti}^{h,CF}(f))^2} \tag{4.1}$$

to calculate RMSE over time steps m to K[2].

RMSE Padding Choice

A complication arises due to the potentially non-uniform length in predicted trajectories. This is because a patient can experience death or discharge during the first 24 hours, which could result in either *over-prediction*, where a Monte Carlo simulation predicts an end of stay on a time step *after* the patient had actually ended his or her stay, or *under-prediction*, where a Monte Carlo simulation predicts an end of stay on a time step *before* the ground truth. In order to make sure our RMSE results are valid and meaningful, we decided handle the cases of over and under-prediction by padding the missing data with an ad-hoc selected value. For both the predicted and actual trajectories, we pad the missing values for trajectories that have experienced release with population mean. Specifically, a covariate's post-release timesteps are padded with the population mean of that covariate. To account for death, we

decided to use the minimum value observed in the dataset for each covariate for padding. However, a few of our covariates are represented in their log-normalized form, meaning we take the log of the raw value and normalizing via the mean and standard deviation of the log values. This poses an issue if the minimum value observed is negative or 0 because the log of a value less than or equal to 0 is undefined. Thus, for covariates that used log-normalization, we padded the post-death trajectories with 0.0001 if the observed minimum was negative or 0, which would give the padded values a defined value. These padding values were intended to represent the patient's physiological status change.

4.4.2 AUC-ROC Score

To evaluate our model's ability to predict whether a patient experienced certain outcomes, we use the area under the receiver operating characteristic (ROC) curve (AUC) metric. We choose to use this assessment primarily to measure our model's ability to predict whether a patient will experience an outcome of interest. An AUC score of 0.5 is congruent with a fair coin flip and an AUC score of 1.0 indicates the ability to perfectly discriminate between positive and negative labels. We additionally used the DeLong's assessment to find 95% confidence intervals[17]. This metric was used to assess both 24 hour outcome prediction and 72 hour outcome prediction.

4.5 MIMIC Dataset Overview

For our study, we use the MIMIC IV dataset, which contains real-world ICU data sourced from the electronic health record of the Beth Israel Deaconess Medical Center [5]. We will now specify how we selected our cohort and introduce the covariates we used in our study.

4.5.1 Cohort Selection

We select ICU patients that met the Sepsis-3 criteria, which identifies sepsis and septic shock identifying sepsis and septic shock using the most up-to-date definitions [18]. Under the Sepsis-3 criteria patients are classified as septic if they have an episode of suspected infection **and** have a Sequential Organ Failure Assessment (SOFA) score of 2 points or more [18], [19]. An episode of suspected infection can be characterized by either an antibiotic administration followed by a sampled culture within 24 hours of the antibiotic or a sampled culture followed by an antibiotic administration within 72 hours of the culture. The time of the first event to occur (between antibiotic and culture sampling) is used as the onset time of suspected infection. Our cohort of interest included only patients over the age of 16 at the time of sepsis onset and had no missing data. Our final dataset yielded 27,139 patients with 35,010 ICU stays related to sepsis.

We exclude patients whose time of suspected infection was more than 24 hours post ICU admission due to our selection of ICU admission as the start time of our study and $K = 24$. We also excluded patients who were admitted following cardiac, vascular, or trauma surgery due to their inherently different mortality risk compared to other ICU patients [19]. We only included the first stay for any patient, whether they had one or multiple ICU stays,

to avoid repeated measurements. We also excluded patients who did not have any recorded documentation regarding fluids administered before their ICU admission. Given most sepsis patients are expected to receive fluid therapy of some sort prior to ICU admission, we assume the lack of pre-ICU fluid documentation is more likely to be due to erroneous documentation rather than the patient actually not being administered any fluids. In the case this was true, the inclusion of these patients would most likely cause confounding issues.

After generating our population of interest, we did one last filtering step for patients who had outlier values of certain measurements as determined with reference to a clinician. Specifically, we capped the amount of pre-ICU fluid administration at 10L because these patients were unlikely to be administered even more fluids under observed care. Given our interest in variations in fluid treatment strategies, including these patients in our cohort would not be meaningful.

The final size of our study population was 8,721 patients with one sepsis-related ICU stay each. During our experiments, we used 6,976 (80%) patients in the training set, 872 (10%) in the validation set, and 873 (10%) patients in the testing set. Table 4.1 presents an overview of our Sepsis-3 cohort of interest by describing the distribution of patients in our cohort with more detail, which helps us frame our work. Table 4.2 presents the population-level proportion of patients who experienced various outcomes of interest. This helps give context to the results of our experiments. In particular, we notice that these proportions are similar to those of our test set, which will be presented in Chapter 6, indicating that the distribution of outcomes experienced in our test set is fairly similar to the entire population.

Table 4.1: Population statistics describing our entire study cohort, including age, gender, and race.

Characteristic	***n***
Number of ICU stays	8,721
Age (μ, σ)	(65.31, 17.33)
Pre-ICU Fluids (μ, σ)	(2.87L, 1.99L)
Male	56.30%
Race – white	66.40%
Race – black	8.78%
Race – other	24.81%
CHF	3.47%
ESRD	25.59%

Table 4.2: Proportion of our cohort of interest that experiences each outcome within 24 hours and 72 hours of ICU admission respectively. *Denotes release from ICU (not from the hospital).

Outcome	24 Hour	72 Hour
Pulmonary Edema (%)	32.85	44.36
Mechanical Ventilation (%)	42.46	45.08
Diuretics (%)	15.46	27.19
Dialysis (%)	2.56	4.55
Release* (%)	11.49	86.00
In-Hospital Mortality (%)	1.83	14.00

4.5.2 Patient Covariates

We selected covariates that would be both important for determining sepsis treatment decisions and are typically monitored in the ICU as predictors for our model. We also included potential confounders. We used similar covariates to the ones used by both Li et al.[2] and Hu [3]. We had a combination of unmodeled static covariates, such as basic demographic information, comorbidities, and pre-ICU fluids, as well as dynamic and modeled variables, such as SOFA score, vital signs, and urine output [2], [3]. A comprehensive list is provided in Tables A.1 and A.2.

Chapter 5

Experiments

Our study's primary focus is evaluating the G-Net framework for predicting individual-level dynamic and time-varying treatment response on clinical data. To do this, we first investigated our model's performance via the *predictive check*, where we evaluated our model's ability to predict the observed outcomes under the observed regime. This showcases the reliability of our model's predictions and provides metrics to select a model for counterfactual prediction. Then we evaluate our best-performing model on individual-level counterfactual prediction. The rest of this chapter will specify how we implemented and carried out the methods described in Chapter 4 and give an overview of the evaluations we performed.

5.1 G-Net Experiments

In this section, we will discuss the details of our experiments. We start by introducing the architectures we used to implement G-Net with. We then describe the different simulations we conducted for the predictive check for both the 24-hour and 72-hour cases.

5.1.1 G-Net Model Implementation

In order to perform g-computation, we first need to select model architectures that will allow us to learn and sample from the conditional covariate distributions. We chose five different architectures to implement G-Net with, four of which use the same model for all boxes, and one of which uses a combination of the architectures for its boxes. The latter approach is possible because each box is independently trained in the *one-variable-per-box* approach. Specifically, we have the following:

1. **Linear** where all of the boxes used generalized linear models (GLMs)
2. **RNN** where all of the boxes used basic recurrent neural networks (RNNs).
3. **GRU** where all of the boxes used RNNs using gated recurrent units (GRU).
4. **LSTM** where all of the boxes used Long-Short Term Memory (LSTM) networks.
5. **Hybrid** where each box was implemented using either a GLM, LSTM, RNN, or GRU.

We decided to test a variety of RNN architectures due to their use of a hidden state to store memory, which mimics conditioning on the entire history of a patient by capturing time-varying correlations. We noticed that some there was not one architecture that consistently higher performance, so we decided to create a hybrid model in which each covariate is modeled by the "best" architecture for that covariate. Our final hybrid model consisted of the architecture with the highest validation performance out of the other four models per box. For the boxes predicting binary variables, we additionally included a sigmoid activation layer.

5.1.2 G-Net Training

We used the Adam optimizer as our gradient descent algorithm of choice with MSE loss for the continuous covariates and BCE loss for the binary variables to train our models. For each box, we performed a grid search across a space of hyper-parameters, and selected the hyper-parameters that resulted in the model with the highest validation score per box. Figure A.4 details the hyper-parameter space we searched through per model.

Then for each box, we selected the box with the highest validation score from our Linear, RNN, GRU, and LSTM models to comprise our final Hybrid model, which is detailed in Figure A.3.

5.1.3 G-Net 24-Hour Predictive Check

In order to do our 24 hour predictive check, we first simulated forward patient covariate trajectories. In order to mitigate the effects of randomness during our simulation process and better approximate the conditional distribution using our learned conditional expectations, we repeat the simulation process for a total of $n = 100$ Monte Carlo simulations per test patient.

Time Delays

To more rigorously evaluate the G-Net framework, this study also investigated our model's performance on the task of individualized prediction conditioned on variable number of time steps. Thus we additionally experimented with different simulation start times post ICU admission. When simulating with start time k, we first feed time steps 1 to $k-1$ of the patient's ground truth covariate trajectories into our model before initiating our simulation process. The remaining predictions are conditioned on $k-1$ time steps of observed ground truth data. This emulates the setting in which a clinician has observed a patient for $k-1$ time steps in the ICU and would like to predict the patient's condition in the future conditioned on what has been observed thus far. We used $k = 2$, 4, 6, and 8 in our experiments.

RMSE

For our RMSE calculations, we took the average across each time step for each continuous covariate to get a final predicted covariate trajectory for each patient that we used for our RMSE calculations. For different time delays k, we set $m = k$ in Equation 4.1.

5.1.4 Data Preparation and Analyses for Outcome Prediction

To calculate our 24 hour AUC score, we first filter out patients who have ended their ICU stay from time steps 1 to $k-1$. This is because we would have already observed what outcomes the patient had experienced up until the end of their stay, and not only would including these patients artificially inflate our evaluation, it doing so would not have meaning in the use case of predicting what a patient will experience in the future up until 24 hours of care in the ICU.

For each Monte Carlo simulation per patient with simulation start time k, we label the trajectory with 1 if the outcome indicator at any time step from k to 24 was 1. Then for each patient, we take the proportion of Monte Carlo simulations with a label of 1 to be the patient's predicted probability of experiencing the outcome. The ground truth labels were generated by determining if the outcome indicator at any time step from k to 24 was 1 in the patient's ground truth trajectory, much like what was done for each Monte Carlo simulation. The final predicted probability was then compared against the ground truth labels to find the final AUC score.

5.2 72 Hour Outcome Prediction

Additionally, as mentioned in Section 4.3, oftentimes the effects of certain treatments are not observable immediately, so we extend our assessment to predicting various clinical outcomes within 72 hours of ICU admission via a binary classifier.

5.2.1 72 Hour Outcome Binary Classifier Training

We used the same training, validation, and testing splits used to train our G-Net models to train our 72-hour outcome binary classifiers, which we chose to implement using an LSTM model with sigmoid activation. The outcome ground-truth labels were generated by looking for the presence of the outcome in each patient's 72-hour ICU data. For each k, we looked for the presence of the outcome from time steps k to 72 to generate their final label. During training for each outcome model, we then exclude patients who had already ended their stay or experienced the outcome of interest during time steps 1 to $k-1$.

We performed a grid search across a space of hyper-parameters that are documented in Figure A.5.

5.2.2 72 Hour Outcome Prediction Testing

During our 72-hour outcome predicting process, we keep the imposed arbitrary ordering used during training. Similar to the 24 hour case, for each outcome, we not only input the 24 hour predicted trajectories as input to the model, but also the outcomes predicted previously in the imposed order. For example, we input $\bar{L}_{24}\bar{A}_{24}$, into our first 72 hour outcome binary classifier model to get $h^0(\bar{L}_{24}\bar{A}_{24};\Gamma^0)$. In the 24-hour case, we chose to use a Bernoulli sample characterized by the predicted value to convert to a binary label, however in this case, a random sampling would only add unwanted noise.

Instead, we use the ROC curve of the validation data set. We input the ground truth $\bar{L}_{24}\bar{A}_{24}$ and $Y^0 \dots Y^{j-1}$ of the validation data as input to our model, and compared the predicted $\hat{Y}^j$ to the ground truth Y^j to find the ROC curve. Then we picked the optimal threshold to use, indicated by the lowest false positive and highest true positive rate (i.e. the most top-left point on the ROC curve), denoted p^j_{val}.

So for every outcome j, we threshold each predicted $h^j(\bar{L}_{24}\bar{A}_{24}; \Gamma^0)$ using p^j_{val} to obtain a predicted binary label $\hat{Y}^j$ to feed as input to the next model.

5.2.3 72 Hour Predictive Check

To perform our 72 hour predictive check, we used the $n = 100$ simulated trajectories from the 24-hour experiments. We use AUC as our metric for assessment. We will now describe how we conducted our 72 hour AUC evaluation.

Data Preparation and Analyses for 72 Hour Outcome Prediction

Same as in the 24 hour case, we filter for patients who have not experienced death or release for time steps 1 to $k-1$. We additionally filter out patients who experienced the outcome from hours 1 to $k-1$ as we did during training. For each patient, we first look at each of their $n = 100$ 24 hour Monte Carlo simulations. If the patient experienced the outcome of interest from time steps k to 24, which was determined by looking at the indicator variable in the trajectory like in the labelling of the 24 hour case, we label the simulation with probability 1. Otherwise, if the patient had ended their stay after time step k and did not experience the outcome during their stay, we labelled the simulation with probability 0. We input all other trajectories to our model to obtain a prediction in the range $[0, 1]$.

This gave us $n = 100$ 72 hour predicted probabilities $h^j(\bar{L}_{24,i}\bar{A}_{24,i}\hat{Y}^0_i \dots \hat{Y}^{j-1}_i) = p^j_i$ for each outcome of interest. We then took the average of the predicted probabilties to get a final predicted probability $\bar{p} = \frac{\sum_{i=1}^n p^j_i}{n}$ per patient. These predicted probabilities were then compared against the patient's 72 hour ground truth labels to calculate the final AUC.

Chapter 6

Results of Predictive Check of G-Net on Sepsis Cohort

In this section, we will present the results of the predictive check of our G-Net models on the sepsis cohort. The predictive check will serve as an evaluation of our model's capabilities and give context to the counterfactual experiments due to the lack of ground truth for quantitative evaluation on counterfactual predictions. It will also allow us to compare different model architectures to select the highest performer for our counterfactual setting. We start with the 24-hour qualitative and quantitative results, then finish with the 72 hour results.

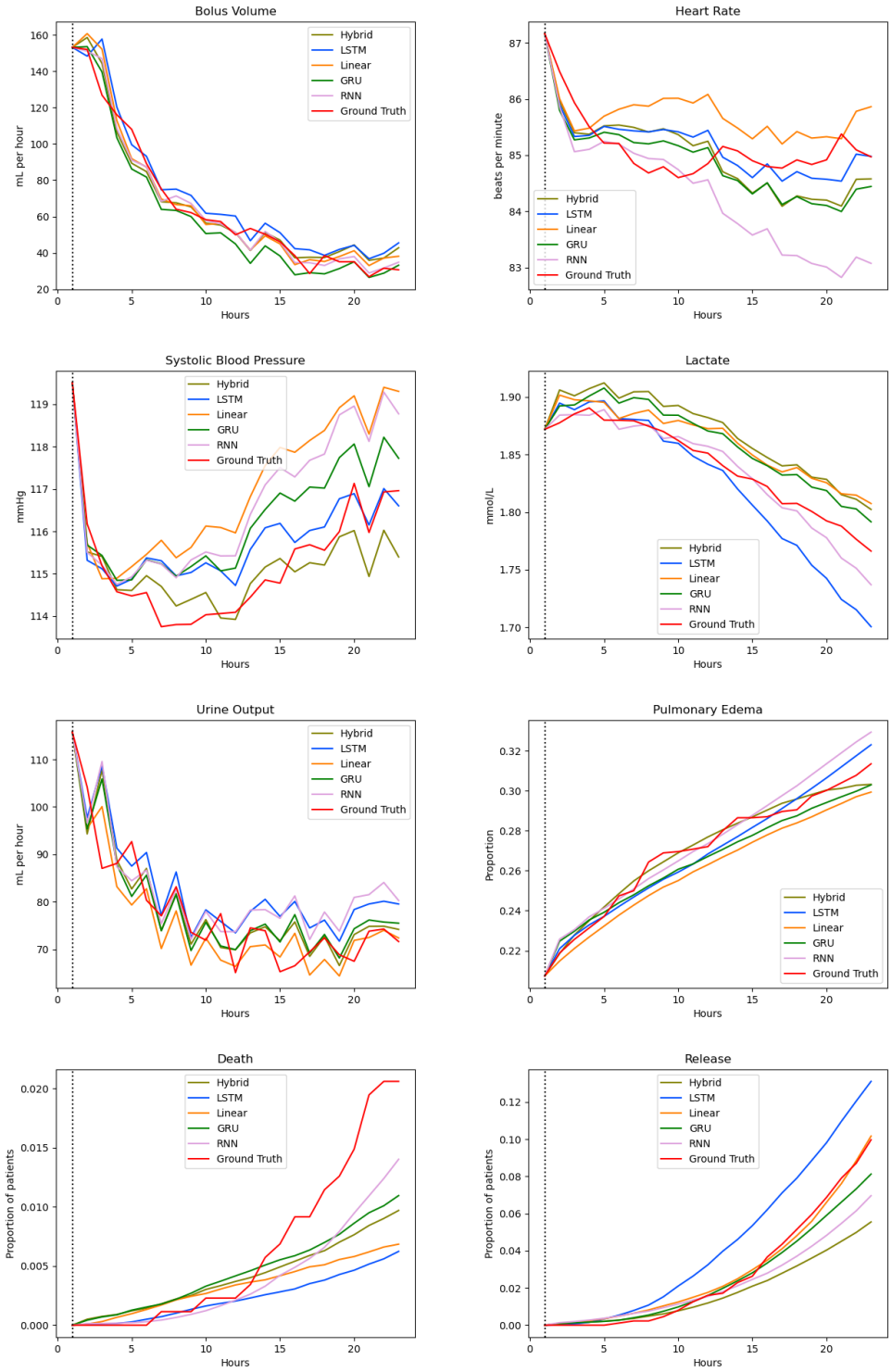
6.1 24 Hour Population-level Trajectories

We first qualitatively assessed our results via population-level trajectory plots, a few of which are displayed in Figure 6.1. The trajectories were generated by aggregating all $n = 100$ Monte Carlo simulations per patient in our test set (873 total patients). To qualitatively evaluate our model, we visually compared our predicted population-level average trajectories against the population-level average ground truth trajectories. If our predictions are accurate on the individual level, we would expect the population-level trajectories to be similar as well.

We caution that while interpreting these plots, the reader should keep in mind that the trajectories have variable length depending on if a patient experienced death or release, thus the number of simulations averaged across is not necessarily the same across every time step. This is particularly true when using these results to frame the counterfactual results that will be presented in Chapter 7.

We see that the general trends of the true population average are well-reflected in the predicted population average, however as the time step increases, the models' predictions tend to deviate more from the ground truth. This is to be expected due to the compounding error that is accumulated throughout time. Additionally, the RNN-style architectures seem to perform better than the Linear model for many of the covariates, which is reflected in the results from the 24 hour RMSE, which will be presented in section 6.2.

Figure 6.1: Comparison of simulated population-level trajectories under different architectures of G-Net compared to the ground truth population-level trajectories at $k = 2$. The Death and Release plots are cumulative.



6.2 24 Hour Individual-level RMSE

We now present the 24 hour individual level RMSE results, which helps us evaluate each model's ability to predict the continuous covariates. Table 6.1 presents the RMSE calculation done, with the padding strategy described in Section 4.4.1. This means the numerical value of the RMSE results evaluate both the model's prediction to predict continuous covariates similarly to the ground truth and over- and under-prediction, which is difficult to tease apart without further analysis.

Table 6.1: 24-hour individual-level RMSE, evaluating the performance of various G-Net architectures in predicting the 24-hour patient continuous covariate trajectories. Predictions start at time k and is conditioned on all covariates from time step 1 to $k-1$, as well as the previous covariates predicted in the arbitrarily imposed prediction order. Time steps post-release are padded with population average, and time steps post-death are padded with either the population minimum or 0.0001 for covariates that are log-normalized. This table appears in [20].

Model	k = 2	k = 4	k = 6	k = 8	k = 10
Linear	5.949	6.232	6.792	6.570	6.584
GRU	**5.663**	6.226	6.211	6.373	6.699
RNN	5.842	6.226	6.053	6.223	6.490
Hybrid	5.758	**5.999**	6.220	6.269	6.642
LSTM	5.864	6.286	**5.921**	**5.765**	**6.439**

Although the RMSE values are not standardized against any external metric and therefore do not carry any inherent meaning, we can compare amongst the RMSEs per model to evaluate model performance. In particular, we notice that the Linear model does not outperform its recurrent architecture counterparts for any of the delay times. This can be attributed to the RNN-style models' architecture that contains a hidden state that can capture temporal dependencies. In our setting, where each L_t^j is dependent on $\bar{L}_{t-1}\bar{A}_{t-1}$, the capability to store and use previously seen information would lead to more accurate predictions. Additionally, due to the more complex structure of the LSTM, we expect the LSTM to be able to capture more complex dependencies in the data. However, due to its complexity, it also may need to see more data before it becomes advantageous over the other models, explaining why it performs better for the larger values of k. Overall, the LSTM performs the best under the RMSE metric.

6.3 24 Hour Outcome AUC-ROC

Next, we will present the 24 hour AUC results, which evaluates the model's predictive capabilities for the binary outcome indicator variables. We note that this evaluation is not as fine-grained as the RMSE calculations because the predicted label is determined by whether the model predicts an outcome of interest at all from time steps k to 24, rather

than a per-time-step evaluation, as the RMSE does. To frame our findings, we first present an overview of the 24-hour test population used in our evaluations in Table 6.2.

Table 6.2: Size of set of patients N_c contributing to the 24 hour AUC calculations for each delay k, as well as proportion of contributing patients experiencing outcomes of interest. [1]Refers to release from Hospital.

k	N_c	**Edema**	**MV**	**Diuretics**	**Dialysis**	**Release**[1]	**Death**
2	873	30.24%	38.72%	15.35%	1.95%	11.45%	2.29%
6	873	29.67%	35.97%	13.52%	1.83%	11.45%	2.29%
8	870	29.77%	33.79%	12.76%	1.84%	11.26%	2.18%
10	868	29.72%	32.03%	12.56%	1.73%	11.06%	2.19%

Table 6.3: Predictive Check. Performance of G-Net in predicting clinical outcomes within the first 24 hours of ICU admission at varying simulation start times. Values reported are AUCs and 95% confidence intervals produced using DeLong's method [17]. [1]Refers to release from Hospital.

Model	k	**Edema**	**MV**	**Diuretics**	**Dialysis**	**Release**[1]	**Death**
Linear	2	0.85 (0.82, 0.88)	0.89 (0.87, 0.92)	0.74 (0.69, 0.79)	0.78 (0.67, 0.90)	**0.67** (0.61, 0.73)	0.49 (0.37, 0.62)
GRU	2	**0.86** (0.83, 0.89)	**0.91** (0.88, 0.93)	0.77 (0.73, 0.82)	0.88 (0.79, 0.98)	**0.67** (0.61, 0.73)	0.68 (0.55, 0.81)
RNN	2	0.85 (0.82, 0.88)	**0.91** (0.89, 0.93)	0.75 (0.70, 0.8)	**0.93** (0.86, 1.00)	0.66 (0.6, 0.72)	0.61 (0.48, 0.75)
Hybrid	2	0.85 (0.82, 0.88)	0.90 (0.88, 0.92)	0.78 (0.74, 0.83)	0.90 (0.81, 0.98)	**0.67** (0.61, 0.74)	0.66 (0.52, 0.79)
LSTM	2	**0.86** (0.84, 0.89)	0.90 (0.88, 0.92)	**0.82** (0.77, 0.86)	0.92 (0.84, 1.00)	0.66 (0.6, 0.72)	**0.74** (0.62, 0.87)
Linear	6	0.89 (0.86, 0.91)	0.93 (0.91, 0.95)	0.72 (0.66, 0.77)	0.87 (0.78, 0.97)	**0.68** (0.62, 0.74)	0.45 (0.33, 0.57)
GRU	6	**0.90** (0.87, 0.92)	0.94 (0.92, 0.96)	**0.82** (0.78, 0.87)	0.92 (0.85, 1.00)	0.67 (0.61, 0.73)	0.65 (0.51, 0.78)
RNN	6	0.88 (0.86, 0.91)	0.94 (0.92, 0.96)	0.78 (0.74, 0.83)	**0.96** (0.90, 1.00)	0.64 (0.58, 0.7)	0.64 (0.51, 0.77)
Hybrid	6	0.88 (0.85, 0.91)	0.94 (0.92, 0.96)	0.81 (0.77, 0.86)	0.95 (0.89, 1.00)	0.67 (0.61, 0.73)	0.68 (0.55, 0.81)
LSTM	6	0.89 (0.87, 0.92)	**0.95** (0.94, 0.97)	0.81 (0.76, 0.86)	0.95 (0.89, 1.00)	0.65 (0.59, 0.71)	**0.85** (0.74, 0.96)
Linear	8	**0.91** (0.89, 0.94)	0.93 (0.91, 0.95)	0.73 (0.68, 0.79)	0.97 (0.91, 1.00)	**0.70** (0.64, 0.76)	0.38 (0.26, 0.5)
GRU	8	0.89 (0.87, 0.92)	0.95 (0.93, 0.97)	0.79 (0.74, 0.84)	0.92 (0.83, 1.00)	**0.70** (0.64, 0.76)	0.69 (0.56, 0.82)
RNN	8	**0.91** (0.88, 0.93)	0.95 (0.93, 0.97)	0.78 (0.72, 0.83)	**0.98** (0.93, 1.00)	**0.70** (0.64, 0.76)	0.68 (0.55, 0.82)
Hybrid	8	0.90 (0.87, 0.92)	0.95 (0.93, 0.97)	0.79 (0.74, 0.84)	0.95 (0.88, 1.00)	**0.70** (0.63, 0.76)	0.65 (0.52, 0.79)
LSTM	8	**0.91** (0.88, 0.93)	**0.96** (0.94, 0.97)	**0.80** (0.75, 0.85)	0.90 (0.82, 0.99)	**0.70** (0.64, 0.76)	**0.78** (0.66, 0.91)
Linear	10	0.92 (0.9, 0.94)	0.94 (0.92, 0.96)	0.75 (0.69, 0.8)	**0.99** (0.95, 1.00)	0.71 (0.65, 0.77)	0.56 (0.43, 0.7)
GRU	10	0.92 (0.9, 0.95)	**0.95** (0.93, 0.97)	0.80 (0.75, 0.85)	0.92 (0.83, 1.00)	**0.72** (0.66, 0.78)	0.67 (0.54, 0.81)
RNN	10	0.92 (0.89, 0.94)	**0.95** (0.94, 0.97)	0.78 (0.73, 0.83)	0.96 (0.90, 1.00)	0.71 (0.65, 0.77)	**0.78** (0.65, 0.90)
Hybrid	10	0.92 (0.90, 0.95)	**0.95** (0.93, 0.96)	0.81 (0.76, 0.86)	0.95 (0.88, 1.00)	0.70 (0.64, 0.76)	0.75 (0.62, 0.87)
LSTM	10	**0.93** (0.91, 0.95)	**0.95** (0.93, 0.97)	**0.82** (0.77, 0.87)	0.98 (0.94, 1.00)	0.68 (0.62, 0.75)	0.69 (0.56, 0.83)

According to Table 6.3, we see that G-Net implemented with Linear tends to be outperformed by its RNN-structure counterparts, which is expected as mentioned in the previous section, and the LSTM implementation performs the best overall. We also note that as the delay increases, the AUCs tend to increase as well. This can be attributed to the additional ground truth we are conditioning on as we increase k.

We also see a relatively lower performance for in-hospital mortality and release from ICU. This could be due to the final nature of the two outcomes. The other outcomes are potentially reoccurring, and oftentimes if a patient experiences it earlier on in their ICU stay, they are more likely to experience it in the future as well. Our model may have observed

and learned this from the training data distribution. On the other hand, death and release occur only once, and there are likely more nuances in the covariate and treatment data to predict either death or release, making it more difficult for our model to predict.

6.4 72 Hour Outcome AUC-ROC

We now examine the 72 hour outcome AUC results, which evaluates our model's ability to predict whether a patient will experience outcomes of interest further down the line. Potential error could stem from two sources - the error in the simulated trajectories as well as the error for the 72 hour binary classifier. Thus, as a reference, we also include the AUC for the predictions when the ground truth trajectories are fed into our binary classifier. We ask the reader to keep in mind when interpreting the ground truth 72 hour outcome prediction performance that if the outcome occurred from time steps k to 24, the prediction is not reliant on the binary classifier and may be artificially inflating the performance. Thus they are presented and intended to be used as reference only.

We first present some population statistics in Table 6.4 to frame our results.

Table 6.4: Size of set of patients contributing to the 72 hour AUC calculations for each outcome at each delay k, as well as proportion of contributing patients experiencing each outcome. [1]Refers to release from Hospital. [2]Refers to In-Hospital Mortality.

k	**Edema**	**MV**	**Diuretics**	**Dialysis**	**Release**[1]	**Death**[2]
2	(873, 43.64%)	(873, 44.90%)	(873, 27.38%)	(873, 4.70%)	(873, 84.77%)	(873, 15.23%)
6	(666, 26.13%)	(561, 14.26%)	(829, 23.40%)	(864, 4.04%)	(873, 84.77%)	(873, 15.23%)
8	(649, 24.35%)	(548, 12.41%)	(813, 22.14%)	(861, 3.48%)	(870, 84.75%)	(870, 15.25%)
10	(632, 22.63%)	(536, 10.63%)	(803, 21.42%)	(856, 3.15%)	(868, 84.83%)	(868, 84.83%)

Table 6.5: Performance of G-Net in predicting clinical outcomes within the first 72 hours of ICU admission at varying simulation start times. Model refers to the G-Net implementation that generated the 24 hour trajectories used for labelling and input to the 72 hour binary classifier. Values reported are AUCs and 95% confidence intervals produced using DeLong's method [17]. [1]Refers to release from Hospital. [2]Refers to In-Hospital Mortality. [3]Ground truth trajectory as input to 72 hour outcome classifier model.

Model	k	**Edema**	**MV**	**Diuretics**	**Dialysis**	**Release**[1]	**Death**[2]
Actual[3]	6	0.87 (0.84, 0.91)	0.92 (0.88, 0.96)	0.87 (0.83, 0.90)	0.97 (0.93, 1.00)	0.87 (0.84, 0.89)	0.83 (0.79, 0.88)
Linear	6	0.67 (0.62, 0.72)	**0.71** (0.64, 0.77)	0.67 (0.62, 0.72)	0.87 (0.80, 0.95)	**0.76** ((0.72, 0.79)	**0.75** ((0.71, 0.80)
GRU	6	**0.68** (0.63, 0.73)	0.67 (0.60, 0.74)	0.71 (0.66, 0.75)	0.89 (0.82, 0.96)	0.74 ((0.70, 0.78)	0.74 ((0.69, 0.79)
RNN	6	0.65 (0.60, 0.70)	**0.71** (0.64, 0.78)	0.70 (0.65, 0.74)	0.90 (0.83, 0.97)	0.75 ((0.72, 0.79)	**0.75** ((0.70, 0.80)
Hybrid	6	**0.68** (0.63, 0.73)	0.68 (0.61, 0.74)	0.71 (0.66, 0.75)	**0.91** (0.84, 0.97)	0.74 ((0.70, 0.78)	0.73 ((0.68, 0.78)
LSTM	6	0.67 (0.62, 0.72)	**0.71** (0.64, 0.77)	**0.72** (0.68, 0.76)	**0.91** (0.84, 0.98)	0.74 ((0.70, 0.78)	0.74 ((0.69, 0.79)
Actual[3]	8	0.86 (0.82, 0.90)	0.9 (0.86, 0.95)	0.86 (0.82, 0.89)	0.95 (0.90, 1.00)	0.88 (0.85, 0.90)	0.84 (0.79, 0.88)
Linear	8	**0.69** (0.64, 0.74)	0.65 (0.58, 0.72)	0.67 (0.62, 0.72)	0.90 (0.83, 0.97)	**0.78** ((0.75, 0.82)	**0.78** ((0.74, 0.83)
GRU	8	0.67 (0.62, 0.72)	0.72 (0.65, 0.79)	0.71 (0.66, 0.76)	0.91 (0.84, 0.98)	0.76 ((0.72, 0.80)	0.76 ((0.71, 0.81)
RNN	8	0.67 (0.62, 0.72)	0.72 (0.65, 0.79)	0.68 (0.63, 0.73)	0.90 (0.83, 0.98)	0.77 ((0.74, 0.81)	**0.78** ((0.73, 0.83)
Hybrid	8	**0.69** (0.64, 0.74)	0.71 (0.64, 0.79)	0.71 (0.66, 0.75)	**0.92** (0.85, 0.99)	**0.78** ((0.74, 0.81)	**0.78** ((0.73, 0.83)
LSTM	8	**0.69** (0.64, 0.74)	**0.73** (0.66, 0.8)	**0.72** (0.67, 0.77)	**0.92** (0.86, 0.99)	0.77 ((0.74, 0.81)	0.76 ((0.71, 0.81)

Overall, the LSTM performed the best for generating trajectories that accurately predict 72 hour outcomes. On average, the AUC scores for the 72 hour outcomes are lower than those of the 24 hour outcomes. Two factors could be at play here. First, there is compounding error - on top of the compounding error that the 24 hour outcomes have, the 72 hour outcome predictions have an additional source of error from the binary classifiers. In addition, in the 24 hour case, fewer patients were excluded from the calculations and the ground truth proportions are much closer to the population-level proportions compared to the 72 hour case, and the resulting AUCs are much higher. Additionally, if we consider the dialysis outcome, we note that much fewer patients were excluded from the calculations at the 72 hour level compared to the other outcomes and the proportion remains similar to the population-level proportion. In accordance, the AUC for dialysis is much more comparable to the 24 hour performance. Thus the performance evaluation for the 72 hour case should be considered in the context of the population used for evaluation.

Chapter 7

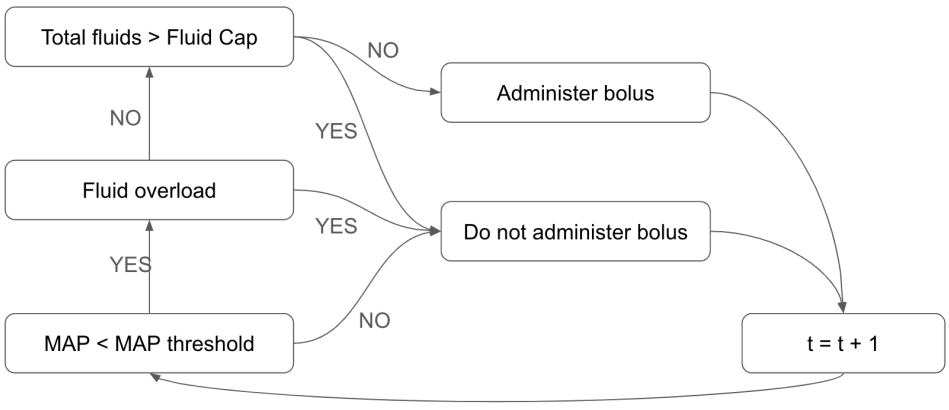
G-Net for Counterfactual Prediction

Given the promising results obtained from the predictive check, we proceed to estimate counterfactual covariate trajectories. This chapter will first describe the counterfactual strategies we used in our experiments then showcase our results via population-level trajectories and a case study of interest.

7.1 Counterfactual Strategies of Interest

A common initial treatment for sepsis patients is the administration of fluids to elevate blood pressure to normal levels in the patient. However, excess fluids can lead to adverse outcomes as a result of fluid overload, such as pulmonary edema. Clinicians will often have to make the decision of how much fluid to give to the patient, as well as decide how to balance fluids with vasopressors when deciding what treatment regime to administer to the patient. Thus we designed counterfactual strategies inspired by the CLOVERS [4] clinical trials.

Figure 7.1: Decision flow chart for our counterfactual fluid-conservative treatment regimes.



For all our fluid strategies of interest, we used a MAP threshold of 65mmHg and a 1L bolus every time bolus was administered. We define a patient being *fluid overloaded* if the patient has pulmonary edema, is on dialysis, or is on diuretics but not on mechanical

ventilation. We additionally stipulated that we would only administer bolus, so we don't administer maintenance fluids at all, with the exception of the fluid liberal strategy. For all counterfactual strategies, vasopressor administration was modeled as a confounding variable, and not explicitly controlled as part of our CF strategies.

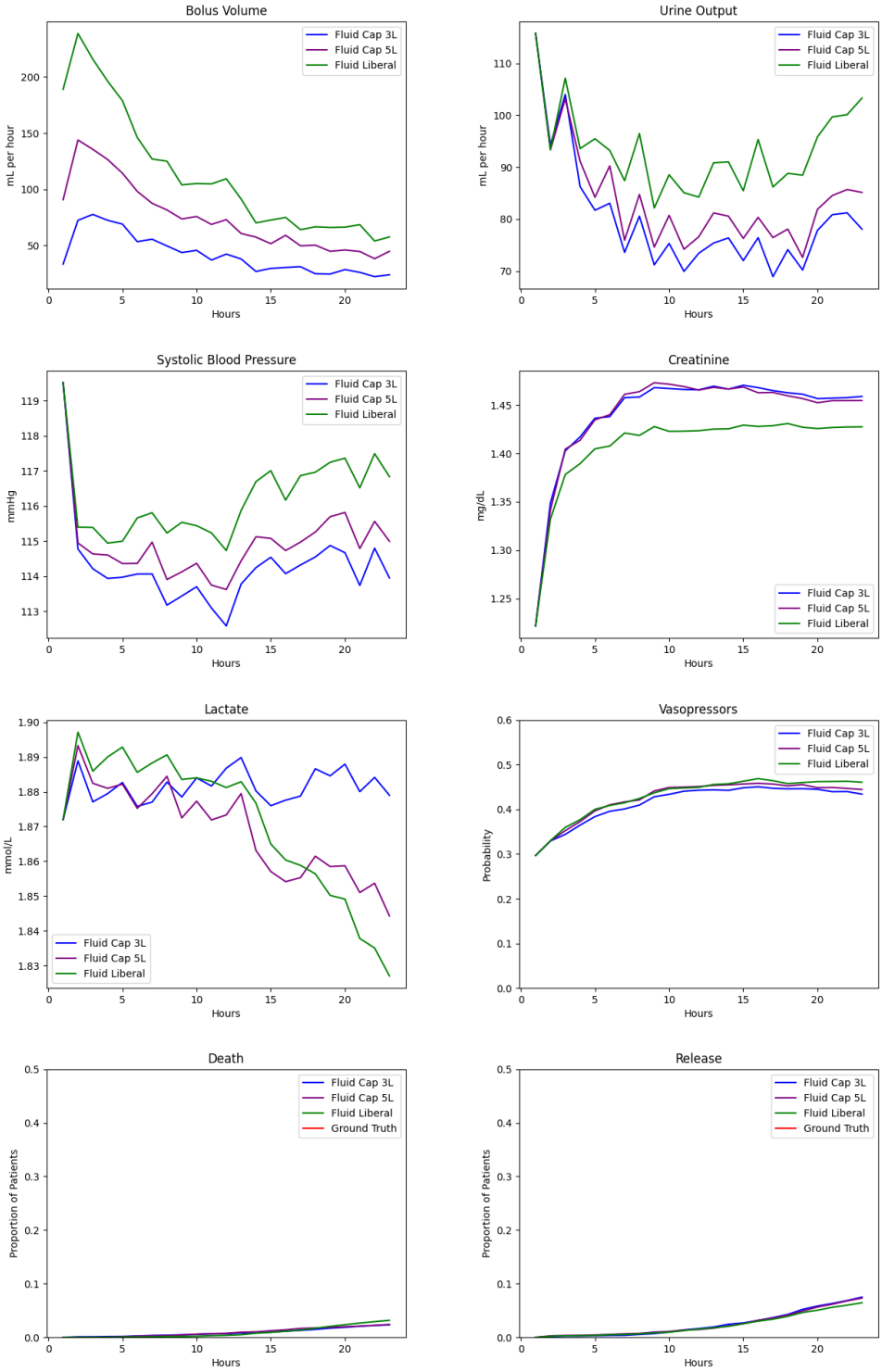
We chose to use the LSTM implementation of G-Net to simulate our counterfactual covariate trajectories. We simulated trajectories for the testing set patients under two fluid-conservative strategies and one fluid-liberal strategy. The fluid-conservative strategies followed the strategy described above with Fluid Cap values of 3L and 5L. The Fluid Liberal strategy similarly implemented the strategy described with no Fluid Cap with the exception of administering maintenance fluids as predicted under the predicted care.

7.2 24 Hour Population Trajectories

We present the simulated population-level trajectories for covariates in Figure 7.2 to compare the population-level predictions for the three strategies. $n = 5$ Monte Carlo simulations were produced for each patient in the test set (873 patients total). Each counterfactual strategy was initiated at time step 1 and covariate predictions started at time step 2. An average was taken across all Monte Carlo simulations for all patients to obtain the population-level trajectories shown.

Due to our lack of ground truth in counterfactual prediction for real-world data, the main criteria for G-Net assessment for counterfactual prediction is whether the resulting predictions follow clinical intuition and can be physiologically explained. We notice that under more fluid-liberal strategies, the bolus volume, urine output, and systolic blood pressure tends to increase, which aligns with physiological expectations. We additionally observe lower levels of predicted creatinine and lactate with more fluid-liberal strategies, which align with our expectations. We additionally note that the trajectories presented in Figure 7.2 should not be interpreted as evaluation of treatment effects. Similar to the observational check, due to early end of stay due to death or release, the length of each simulated trajectory is variable, and the number of trajectories factoring into the average population does not remain consistent throughout all 24 time steps.

Figure 7.2: Comparison of simulated population-level trajectories under different simulated counterfactual strategies using the LSTM model at $k = 2$.



7.3 Sample Case Study

We also examined our model's performance on an individual case through a case study. Many times, clinicians may wonder what would have happened to a patient if a different treatment regime was administered, so we selected a clinically interesting case study to demonstrate a potential use case.

Case Study Patient Introduction

Our patient of interest is a 75 year old female with a history of congestive heart failure (CHF), admitted to MICU/SICU with 4L of pre-ICU fluids. Within 4 hours after ICU admission, patient had developed a hypotensive episode (with mean arterial blood pressure < 60mmHg). The counterfactual strategies were applied starting at the end of hour 5 after ICU admission and covariate simulation was initiated for hour 6 after ICU admission.

Case Study Experiment Details

We primarily wanted to contrast the predicted counterfactual outcomes under a fluid conservative compared to fluid liberal regime. We chose the fluid conservative regime with Fluid Cap 3L and Fluid Liberal strategies presented in the previous section. As before, we choose the LSTM version of G-Net, due to its higher performance during the observational checks, to the task of individual-level counterfactual prediction for an individual in our test set.

Case Study Results

Figure 7.3 indicates a higher volume of bolus administered to the patient under the fluid liberal strategy compared to fluid conservative strategy. Furthermore, we see an increase in predicted probability of the patient being put on mechanical ventilation under the fluid liberal strategy compared to the fluid conservative strategy. This aligns with signs of fluid overload and excess fluids in the body, which is expected of a regime that administered more fluids compared to the fluid conservative strategy.

We can see that the trends from Figure 7.3 are clinically valid. For example, fluid liberal strategy is predicted to have lower creatinine concentration and slightly higher urine output than its fluid conservative counterpart. We additionally compare the predicted probability of 72 hour outcomes the patient will experience under the two regimes.

Figure 7.4 displays the predicted probabilities of this patient developing various adverse clinical outcomes under the observational, conservative and liberal fluid strategies respectively. We note that the patient is predicted to have a higher probability of experiencing most of the adverse outcomes of interest associated with fluid overload under the fluid liberal strategy compared to the conservative strategy.

We note here that the probabilities are not well calibrated due to the lack of calibration in our 72 hour binary classifiers. Therefore we focus on the difference between the two predicted values rather than the actual predicted value and see that the fluid liberal strategy consistently predicts higher probability of experience fluid overload outcomes, which is clinically consistent. However, as mentioned before, these results are not to be taken to measure

treatment effects due to lack of uncertainty quantification, and rather is to showcase an application of our work.

Figure 7.3: Comparison of predicted covariate trajectories for select patient under fluid liberal and fluid conservative counterfactual regimes. Dotted lines indicate cumulative plots.

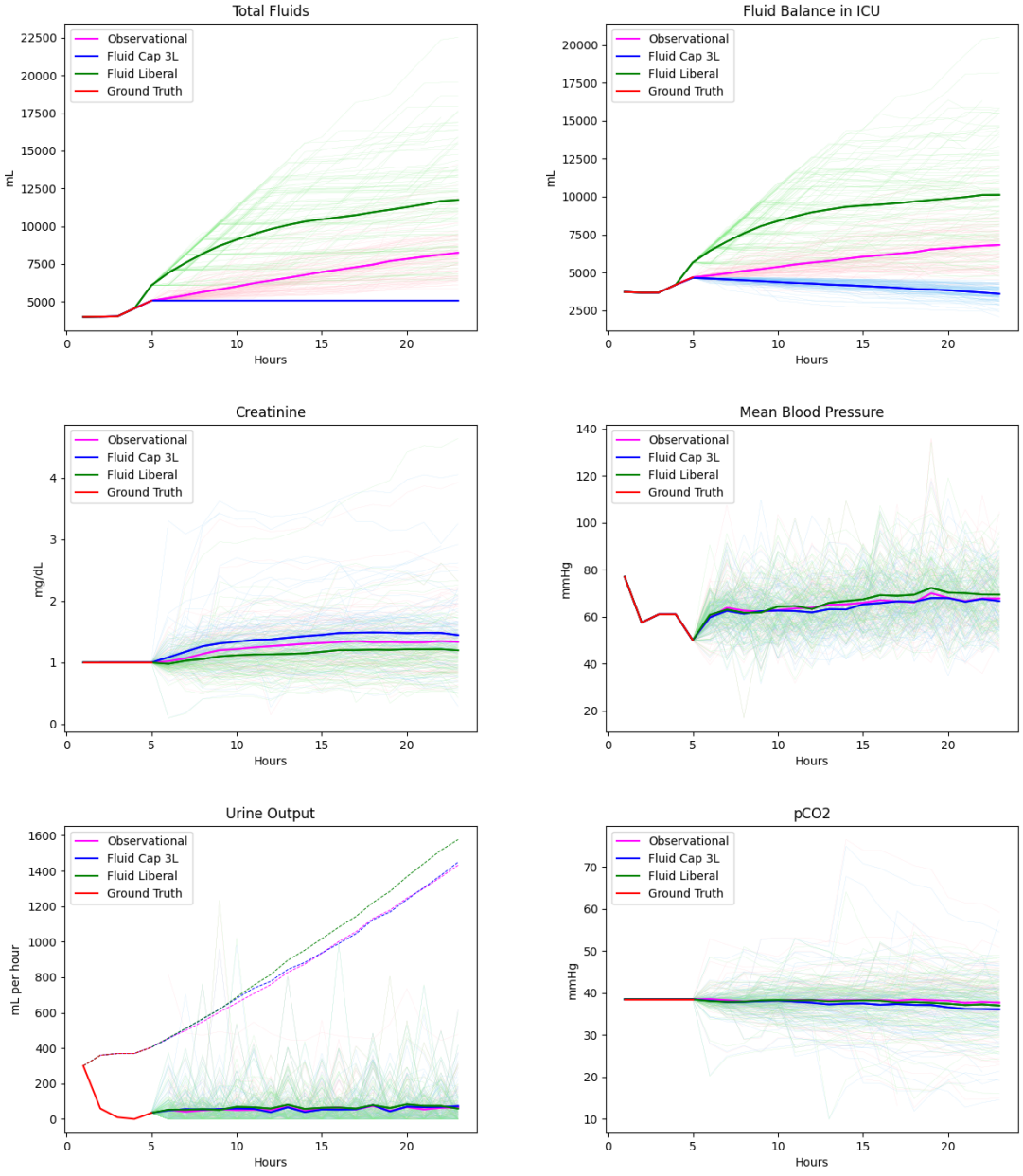
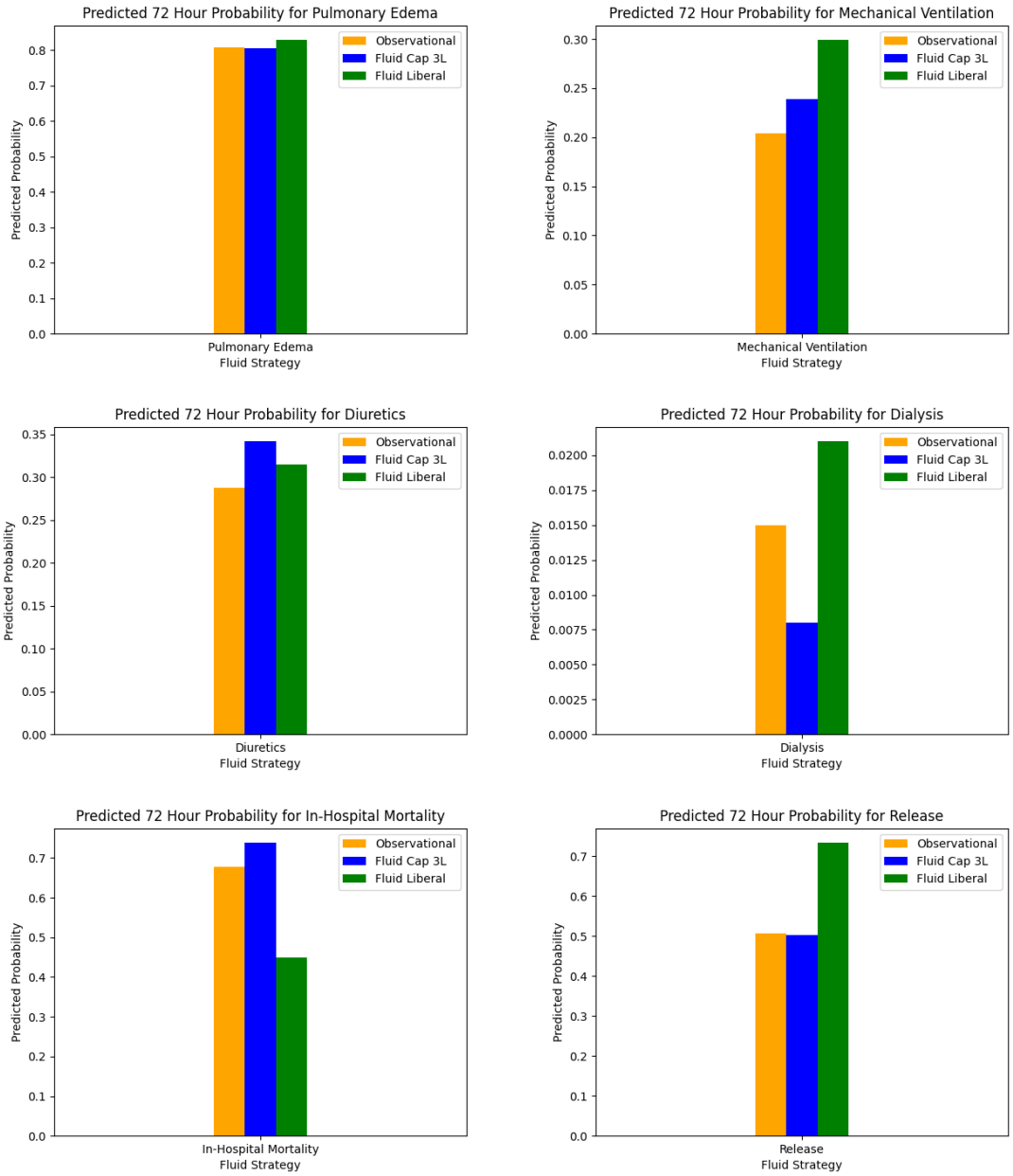


Figure 7.4: Predicted probability of the patient of interest experiencing adverse outcomes associated with fluid overload within 72 hours of ICU admission under the observational, fluid liberal, and fluid conservative regimes.



Chapter 8

Conclusion

Determining optimal care for sepsis patients is difficult due to the variety in response to the same treatment regime and to the potential adverse outcomes that can result from the treatment itself. In the real world, clinicians can only observe the set of outcomes that actually occurred, but it would be useful to be able to predict the outcomes of their choices of treatment before choosing one to administer. G-Net is a deep sequential learning framework based of g-computation that tackles this task of counterfactual prediction for sepsis patients. However, past work has primarily evaluated G-Net on population-level treatment effects.

This thesis focuses on personalized treatment response prediction under dynamic and time-varying treatment strategies by building on G-Net. First, we introduced the GRU and RNN implementation of G-Net and presented their performances against the GLM and LSTM implementations, and additionally presented a Hybrid version. We then evaluated the models in the predictive check, primarily focusing on assessment on the individual level. We find that the LSTM implementation seemed to perform the best on average – it performed the best for RMSE on most of the delay k settings and for the 24 hour AUCs as well. It also generated trajectories that resulted in highest 72 hour AUC for the majority of the outcomes of interest along different delay times.

After selecting the optimal performer based on the predictive check results, we apply the LSTM implementation on counterfactual prediction. We chose to adapt sepsis treatment strategies that appeared in recent clinical trials to fit our setting under modelling limitations. We found that our model's predictions were physiologically plausible and aligned with clinical intuition. However, a few limitations and open issues remain present in our current work.

8.1 Limitations

One open issue lies in the RMSE calculation. Because we are working with real world data in a problem setting that results in variable-length covariate trajectories, we need to adjust the RMSE calculation to account for over- and under-prediction of trajectories, i.e. when the predicted and the actual trajectories have different lengths. In our study, we choose to determine a standard length for all trajectories, then pad any missing values using a padded value that was chosen ad-hoc. Thus the current RMSE measures both error in predicting the continuous variables and over and under-prediction of the death or release of a patient.

More sensitivity analysis should be done to tease apart the respective contributions to the final RMSE scores.

Additionally, for the 72 hour AUC assessment, there was a dip in performance with respect to delay time k. As mentioned prior, this is most likely due to the inconsistent set of patients that factor into our evaluation, particularly with respect to the distribution of patients who ultimately experience the outcome in the set of patients factoring into the evaluation. A more stable and concise evaluation technique might be more useful and allow us to draw more concrete conclusions.

Additionally, more work can be done to refine the Hybrid model. In theory, because we selected the model with the best validation performance for each box, we expected better results from the Hybrid model. That's not to say the model with the best validation score will definitely perform best on the test set, but there may be better ways to produce a Hybrid model, such as including different combinations of models for the boxes during the validation stage.

Finally, there are many limitations regarding the counterfactual analysis. The first limitation is in the lack of uncertainty quantification, which can most likely be done using variational dropout or some bootstrapping techniques. This would be able to provide more meaningful results and potentially allow for us to draw more conclusions about the performance of our model on counterfactual prediction. Additionally, as mentioned previously, there may be confounding in the data itself. Specifically, the data that is measured and presented in the MIMIC IV dataset [5] may not have sufficiently granular data enough to test certain counterfactual regimes.

8.2 Future Works

This thesis used several deep sequential models to implement G-Net with, however they are by no means the current state-of-the-art models. More complex sequential models, such as Transformers and LLMs, have surfaced with recent works and are a great fit for the G-Net framework.

It would also be interesting to use representation learning in our framework to investigate if subgroups of people may react similarly to treatment regimes. This could be further extended to discover optimal treatment regimes for subgroups of patients, which would be clinically relevant as well.

8.3 Final Words

We hope that the individual-level evaluations of G-Net presented in this thesis showcase a clinically-relevant use case that will be able to aid clinicians in their choice of care and ultimately improve the outcomes of sepsis patients as a whole. This thesis focused on applying G-Net to counterfactual prediction for sepsis patients, but the flexibility of the G-Net framework makes its potential application space much broader.

Appendix A

Appendix

A.1 Tables

Table A.1: 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.2: 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
Minimum Mean Blood Pressure	Continuous	mmHg
Minimum Change in Mean Blood Pressure from Baseline	Continuous	mmHg
Minimum Mean Blood Pressure from Baseline	Continuous	mmHg
Minimum Change in Mean Blood Pressure from Previous	Continuous	mmHg
Temperature	Continuous	C
SOFA Score	Treated as Continuous	N/A
Change in SOFA Score from Baseline	Treated as Continuous	N/A
Change in SOFA Score from Previous	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
O2 Requirement Level	Continuous	N/A
Change in O2 from Baseline	Continuous	N/A
Change in O2 from Previous	Continuous	N/A
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
Change in Weight	Continuous	kgs
Respiratory Rate	Continuous	breaths/min
Fluid Volume*	Continuous	mL
Urine Output	Continuous	mL
Cumulative Edema	Binary	N/A
Pulmonary Edema Indicator	Binary	N/A
Diuretics Indicator	Binary	N/A
Dialysis Indicator	Binary	N/A
Mechanical Ventilation Indicator	Binary	N/A
Vasopressor Indicator	Binary	N/A
Bolus Volume	Continuous	mL
In-Hospital Mortality Indicator	Binary	N/A
Release Indicator	Binary	N/A
Cumulative Fluids	Continuous	mL
Fluid Balance in ICU	Continuous	mL

Table A.3: Chosen architecture for each box in the final Hybrid model based on validation score.

Variable	**Box Architecture**
Heart Rate	LSTM
Diastolic Blood Pressure	GRU
Systolic Blood Pressure	GRU
Mean Blood Pressure	GRU
Minimum Mean Blood Pressure	LSTM
Temperature	GRU
SOFA Score	GRU
Platelet	GLM
Hemoglobin	GLM
Calcium	GRU
BUN	LSTM
Creatinine	GRU
Bicarbonate	GRU
Lactate	LSTM
O2 Requirement Level	GRU
pO2	LSTM
sO2	GRU
spO2	LSTM
pCO2	GLM
Total CO2	GRU
pH	LSTM
Base excess	GLM
Weight	GLM
Respiratory Rate	GRU
Vasopressor Amount	LSTM
Maintenance Fluid Volume	GRU
Urine Output	GRU
Edema Indicator	GLM
Diuretic Indicator	GRU
Dialysis Indicator	GRU
Mechanical Ventilation Indicator	GRU
Vasopressor Indicator	GRU
Bolus Volume	GRU
Death Indicator	GRU
Release Indicator	GRU

Table A.4: G-Net Tuning Hyperparameters.

Hyperparameter	**LSTM**	**Linear**	**GRU**	**RNN**
Epoch	100	100	100	100
Stop Window	10	10	10	10
Stop tolerance	0.001	0.001	0.001	0.001
Batch size	32	32	32	32
Hidden dimension	[32, 64]	N/A	[32, 64]	[32, 64]
Number of Layers	[2, 3]	N/A	[2, 3]	[2, 3]
Learning rate	0.001	0.001	0.001	0.001
Weight decay (L2 penalty)	1e-6	1e-6	1e-6	1e-6
Dropout Probability	0.1	0.0	0.1	0.1

Table A.5: Binary Outcome Classifier Tuning Hyperparameters.

Hyperparameter	**Range**
Epoch	50
Stop Window	5
Stop tolerance	[0.001, 0.0001]
Batch size	64
Hidden dimension	[32, 64]
Number of Layers	[2, 3]
Learning rate	0.001
Weight decay (L2 penalty)	[1e-6, 1e-5]
Dropout Probability	0.0

References

[1] M. Hernan and J. Robins, *Causal inference: What if*, First. Chapman & Hall CRC, 2011, ISBN: 1420076167.

[2] R. Li, S. Hu, M. Lu, Y. Utsumi, P. Chakraborty, D. Sow, P. Madan, M. Ghalwash, Z. Shahn, and L. H. Lehman, "G-Net: A recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime," in *Proceedings of Machine Learning for Health*, vol. 158, 2021, pp. 280–297.

[3] S. Hu, *A recurrent network approach to g-computation for sepsis outcome prediction under dynamic treatment regimes*, Master's thesis, Available at https://dspace.mit.edu/handle/1721.1/140128, Cambridge, MA, Sep. 2021.

[4] W. Self, M. Semler, R. Bellomo, S. Brown, B. deBoisblanc, and J. Exline M.C. (2018 February - 2020, *Crystalloid liberal or vasopressors early resuscitation in sepsis (clovers)*. [Online]. Available: https://clinicaltrials.gov/ct2/show/NCT03434028.

[5] A. Johnson, L. Bulgarelli, L. Shen, *et al.*, "MIMIC-IV, a freely accessible electronic health record dataset," *Nature Scientific Data*, 2023.

[6] N. Shapiro, I. Douglas, R. Brower, S. Brown, M. Exline, A. Ginde, and et al., "Early restrictive or liberal fluid management for sepsis-induced hypotension," *New England Journal of Medicine*, vol. 388, pp. 499–510, 2023.

[7] 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.

[8] B. Lim, A. Alaa, and M. Van der Schaar, "Forecasting treatment responses over time using recurrent marginal structural networks," in *Neural Information Processing Systems (NIPS).*, 2018.

[9] 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.

[10] 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.

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

[12] P. Schulam and S. Saria, *Reliable decision support using counterfactual models*, 2018. arXiv: 1703.10651 [stat.ML].

[13] Y. Xu, Y. Xu, and S. Saria, *A bayesian nonparametric approach for estimating individualized treatment-response curves*, 2016. arXiv: 1608.05182 [cs.LG].

[14] H. Choi, C. Jung, T. Kang, H. J. Kim, and I.-Y. Kwak, "Explainable time-series prediction using a residual network and gradient-based methods," *IEEE Access*, vol. 10, pp. 108 469–108 482, 2022. DOI: 10.1109/ACCESS.2022.3213926.

[15] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Learning time series associated event sequences with recurrent point process networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 3124–3136, 2019. DOI: 10.1109/TNNLS.2018.2889776.

[16] J. 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*, vol. 7, no. 9-12, pp. 1393–1512, 1986.

[17] E. DeLong, D. DeLong, and D. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," *Biometrics*, vol. 44, pp. 837–845, 1988.

[18] M. Singer, C. Deutschman, C. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, and et al., "The third international consensus definitions for sepsis and septic shock (sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 801–810, Feb. 2016. DOI: https://doi.org/10.1001/jama.2016.0287.

[19] A. Johnson, J. Aboab, J. Raffa, T. Pollard, R. Deliberato, L. Celi, and D. Stone, "A comparative analysis of sepsis identification methods in an electronic database," *Critical Care Medicine*, vol. 46, no. 4, pp. 494–499, Apr. 2018.

[20] M. Su, S. Hu, H. Xiong, E. B. Kassis, and L. H. Lehman, "Deep sequential modeling for counterfactual sepsis outcome prediction under dynamic and time-varying treatment regimes," in *Proceedings of the AMIA 2024 Informatics Summit*, 2024.