






Forecasting Treatment Outcomes Over Time Using Alternating Deep Sequential Models

Feng Wu , Guoshuai Zhao , *Member, IEEE*, Yuerong Zhou , Xueming Qian , *Member, IEEE*, Elias Baedorf-Kassis, and Li-wei H Lehman , *Member, IEEE*

Abstract—Medical decision making often relies on accurately forecasting future patient trajectories. Conventional approaches for patient progression modeling often do not explicitly model treatments when predicting patient trajectories and outcomes. In this paper, we propose Alternating Transformer (AL-Transformer) to jointly model treatment and clinical outcomes over time as alternating sequential models. We leverage causal convolution in the self-attention mechanism of AL-Transformer to incorporate local spatial information in the sequence, thus enhancing the model's ability to capture local contextual information of the sequence. Additionally, to predict the sparse treatment, a constraint learned by a convolutional neural network (CNN) is used to constrain the sparse treatment output. Experimental results on two datasets from patients with sepsis and respiratory failure extracted from the Medical Information Mart for Intensive Care (MIMIC) database demonstrate the effectiveness of the proposed approach, outperforming existing state-of-the-art methods.

Index Terms—Machine learning, sequential models, time series forecasting, treatment outcome prediction, clinical decision making.

I. INTRODUCTION

PREDICTING a patient's future trajectory and their treatment needs can provide valuable information for personalized healthcare and clinical decision making. A substantial

Manuscript received 1 May 2023; revised 5 September 2023 and 11 October 2023; accepted 29 October 2023. Date of publication 9 November 2023; date of current version 21 March 2024. The work of Guoshuai Zhao was supported in part by the NSFC, China under Grants 61902309 and 62372364, and in part by the Fundamental Research Funds for the Central Universities, China under Grant xzd012022006. The work of Li-wei H Lehman was supported in part by the MIT-IBM Watson AI Lab and in part by the NIH under Grants R01EB030362 and R01EB017205. (*Corresponding author: Guoshuai Zhao.*)

Feng Wu is with the School of Software Engineering, Xi'an Jiaotong University, China, and also with the Institute for Medical Engineering and Science, Massachusetts Institute of Technology (MIT), USA.

Guoshuai Zhao was with the Massachusetts Institute of Technology (MIT), Cambridge, MA 02142 USA. He is now with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: guoshuai.zhao@xjtu.edu.cn).

Yuerong Zhou is with the School of Software Engineering, Xi'an Jiaotong University, China.

Xueming Qian is with the School of Information and Communication Engineering, Xi'an Jiaotong University, China.

Elias Baedorf-Kassis is with the Department of Anesthesia, Pain and Critical Care and the Division of Pulmonary and Critical Care, Beth Israel Deaconess Medical Center, Harvard Medical School, USA.

Li-wei H Lehman is with the Institute for Medical Engineering and Science, Massachusetts Institute of Technology (MIT), USA.

Digital Object Identifier 10.1109/TBME.2023.3331298

body of research has focused on the development and application of machine learning techniques in clinical outcome prediction. [1], [2], [3], [4], [5]. Deep learning, in particular, has emerged as a promising approach for automated extraction of complex data representations for end-to-end training. Recurrent neural networks (RNN) models and their variants, such as Long Short-Term Memory (LSTM), have demonstrated promising performance in modeling sequences and time series data [1], [3]. Transformers have more recently emerged as the state-of-the-art technique to model sequence data with complex temporal dependencies [5], [6], [7]. However, existing methods in clinical outcome prediction or disease progression modeling often do not model time-varying treatments when predicting patient trajectories and outcomes [4], [5], [8], [9]. Other prior works focused on predicting patients' need for treatment without modeling subsequent time-varying patient outcomes in a long horizon sequential treatment setting [10], [11], [12], [13]. Predicting patient outcomes under dynamic treatment regimes remain a challenging task, as treatments often depend on previous time-varying treatments and covariates.

We present a transformer-based architecture to jointly model the time-varying treatments and outcomes over time as alternating sequential models. We predict the treatments and the corresponding outcomes in alternating time steps sequentially, conditioned on the patient's past covariates and treatments. Our goal is to forecast the future trajectory of a patient under the predicted treatment sequence over time under the observational treatment strategies. Sequential modeling of patient outcomes and time-varying treatments using multivariate time series presents several challenges. The complex inter-dependency between treatment and outcome can be challenging to model, where treatment decisions are often influenced by past treatments and patient outcomes. In order to model the expected future trajectory of a patient under the observational treatment policy, an accurate sequential model for the time-varying treatment regime needs to be learned from the observational data. Additionally, in a longitudinal clinical setting, treatment variables are typically sparse in time relative to the outcome variables. Temporal modeling of these kinds of variables are challenging. For example, many patient outcome variables, e.g. heart rate, blood pressure and urine output, can be sampled hourly, but most of the treatments and medications are sparse in time points. Such sparse treatment data are difficult to model and predict.

To address the above challenges in forecasting future patient trajectories under the modeled observational policy, we

propose an alternating sequential model ALternating Transformer (AL-Transformer) to simulate forward the future treatments and outcomes of a patient, conditioned on the patient's past treatments and outcomes. Our approach uses an encoder to encode a patient's previous treatment and outcome sequences, and predict the patient's treatment outcome sequence at the current time. The decoder in AL-Transformer subsequently takes this predicted outcome sequence and the encodings of previous outcomes and treatments as input, to predict the treatment sequence of current time. Considering the treatment variables are sparse, we propose a sparsity constraint learned by a CNN module to constrain the sparse treatment output. Furthermore, to address the challenge of incorporating local context information into the self-attention mechanism employed by the original Transformer, we employ the causal convolutional self-attention mechanism to more effectively model the temporal context within the sequence.

Experimental results on two datasets from MIMIC-III [14], a real-world intensive care database, demonstrate that the proposed model outperforms the state-of-the-art baseline methods. In particular, our proposed AL-Transformer architecture which models time-varying treatments and outcomes as alternating sequential process with sparsity constraints significantly outperformed (with a reduction of 20.64% in mean absolute error) the baseline Transformer approach which models treatments and outcomes as multi-variate time series.

Our main contributions are summarized as follows.

- We introduce a CNN-based sparsity constraint that predicts the need for treatments in future time steps, a novel approach that enhances the accuracy of handling sparse medical treatment data.
- Our model adopts causal convolution in self-attention to better capture local temporal dynamics, significantly improving the prediction of patient outcomes at larger temporal scales.
- The AL-Transformer we proposed is a new architecture for simultaneously predicting patient treatments and outcomes. Extensive experiments on the real-world dataset demonstrate the superior performance over state-of-the-art methods, with a significant reduction in prediction error.

II. RELATED WORK

Treatment and Disease Progression Modeling: Prior works have proposed machine learning models for disease progression modeling and outcome prediction using multivariate clinical time series [4], [8], [9], [15], [16]. For example, [16], [17] used Gaussian Process based models for modeling clinical time series data. These prior works, however, typically do not account for time-varying treatments when modeling patient trajectories. Furthermore, Gaussian Processes are difficult to scale, and typically make strong assumptions on the model structure. In contrast, our approach is more flexible, and can update internal states of the model to predict trajectories of new patients. Other prior works focused on treatment or physician action prediction. For example, Ren et al. [10] used gradient boosting to predict urgent need for intubation in ICU patients.

Recent works Interpole [11] and Treatment-RSPN [13] modeled physician treatment decision dynamics using input-output hidden Markov model (IOHMM), and Treatment-RSPN [13] additionally also modeled response prediction, but these works focused on one-step ahead prediction instead of long horizon multi-step prediction. INPREM [18] proposed a Linear model with random gate to measure the uncertainty of predicted treatment. MED-BERT [19] utilized the pre-trained model and large language model to modeling the disease progression by text records in EHR. To address the issue of sparsity and irregularity in treatment sequences, STRaTS [20] employs a self-supervised Transformer architecture for modeling time series.

Treatment Effect and Counterfactual Prediction: Xu et al. [21] developed a Bayesian nonparametric method for estimating univariate treatment response curves from sparse observational time series. Soleimani et al. [22] proposed a semi-parametric Bayesian framework using Gaussian Processes (GPs) to model treatment effects in multivariate longitudinal data and it unifies response modeling for both discrete and continuously-administered treatments. More recently, approaches for counterfactual predictions have been proposed to predict treatment response over time under target counterfactual treatment strategies of interest [23], [24], [25], [26], [27]. Notably, G-Net [26] used LSTM for counterfactual prediction of time-varying treatment outcomes under alternative dynamic treatment strategies. [27] proposed a Causal Transformer architecture for counterfactual outcomes prediction; different from our method, they combined 3 subnetworks and time-varying covariates into the self-attention structure. These methods rely on strict causal assumptions with the primary goal of making unbiased estimates of treatment effect under target counterfactual treatment strategies. Furthermore, these works do not focus on modeling sequential treatments from observational data; instead, the focus of these prior works is to model the outcomes under target *counterfactual* treatment policies of interest. In contrast, our work simultaneously predicts the treatments under observational dynamic treatment policies and time-varying outcomes in alternating steps.

Deep learning methods: are being increasingly studied in clinical healthcare applications and show better performance than traditional machine learning models [1], [28], [29]. [1], [30] presented the benchmark results for several clinical prediction tasks such as mortality prediction, length of stay prediction, and ICD-9 code group prediction using deep learning models. [28] proposed a representation of patients' entire raw EHR records based on the Fast Healthcare Interoperability Resources (FHIR) format and demonstrated that deep learning methods are capable of accurately predicting multiple medical events. RNNs are generally utilized [1], [31] to model the time-varying variables on three clinically-relevant prediction tasks. [29] proposed a novel model by adding a new gate in RNN to learn the joint representation of heterogeneous temporal events. Attention models [4], [32] are also leveraged for clinical time-series modeling, thereby dispensing recurrence entirely. [4] developed the SAnD (Simply Attend and Diagnose) architecture, which employs a masked, self-attention mechanism, and uses positional encoding and dense interpolation strategies for incorporating temporal order. [33], [34] utilized the contrastive learning in the clinical time series.

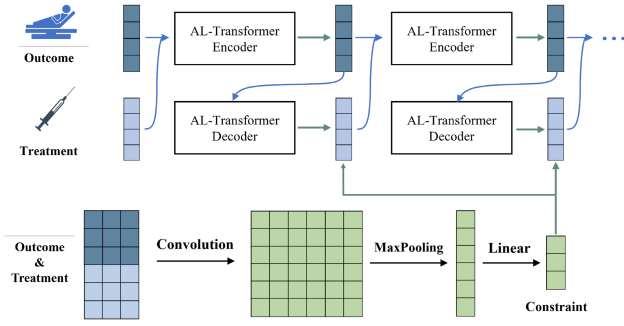


Fig. 1. Illustration of the alternating model. We use the AL-Transformer encoder to encode the outcomes and treatments. After obtaining the representations, we predict the outcomes for the next time step using the decoder. The model utilizes the predicted outcomes and treatments as inputs to predict the outcomes in the subsequent time steps. To control the sparsity of the treatment variables, we also train a CNN network to constrain the treatment variable.

III. OUR MODEL

Notation: We denote our dataset as $D = R, A$, an integrated set of discrete or continuous-valued treatments and outcomes. Given an ICUs visit indexed by i , each patient in our dataset has the clinical data R_i and A_i . The patient response R_i consists of outcome variables $R_{i,t,1}, R_{i,t,2}, \dots, R_{i,t,n}$. Each outcome variable $R_{i,t,n}$ represents vital signs or clinical lab measurements, e.g. heart rate, blood pressure, glucose, and urine output, where i indicates the patient ID, t is the current time step and n is the dimension of the outcome variables. Similarly, the treatment A_i consists of treatment variables $A_{i,t,1}, A_{i,t,2}, \dots, A_{i,t,m}$. Each treatment variable $A_{i,t,m}$ represents a continuous-valued medication or injection dosage, e.g. the amount of vasopressor administered in the time step t . m is the number or dimension of the treatment variables.

For each patient, our goal is to predict the next k outcomes and treatments $(R_{t+1}, A_{t+1}), (R_{t+2}, A_{t+2}), \dots, (R_{t+k}, A_{t+k})$ based on the variables $(R_1, A_1), (R_2, A_2), \dots, (R_t, A_t)$ in previous time steps. To this end, we propose an alternating sequential model AL-Transformer, and Fig. 1 shows the overview of its model structure. In our proposed AL-Transformer, we propose a sparsity constraint to constrain the sparse treatment output and use the causal convolutional self-attention to enhance the temporal modeling. This section describes the framework of AL-Transformer at first, and then introduces the causal convolutional self-attention and the sparsity constraint in detail. Finally, the loss functions are presented.

A. Framework of AL-Transformer

We utilize the encoder and decoder modules of the AL-Transformer to model patients' time series of outcomes and treatments, and alternate between them for prediction. As illustrated in Fig. 2, the encoder module models the patient's previous outcomes and treatments while predicting the patient's outcomes at the current time step, and outputs the encoding of previous outcomes and treatments. The decoder module takes the outcome and treatment sequences at the current time step and the encoded representations from the encoder module as

inputs, and predicts the treatment sequence at the current time step. This process is then repeated by the encoder and decoder modules for alternate prediction of outcomes and treatments.

The encoder predicts the sequence of patient i at time t based on previous outcome and treatment of patient i . Its input can be defined as:

$$X_{en} = [R_{i,1} \oplus A_{i,1}; \dots; R_{i,t-1} \oplus A_{i,t-1}]W_{en}^{in} + PE \quad (1)$$

where \oplus indicates concatenation between vectors; W_{en}^{in} indicates the weight matrix of input linear transformation; PE represents the position encoding. We employ the convolutional self-attention operation on the encoder input:

$$X_{en}^{temp} = LN(X_{en} + CA(X_{en}, X_{en}, X_{en})) \quad (2)$$

where LN indicates the layer normalization operation; CA indicates the causal convolutional self-attention operation.

The output of the convolutional self-attention is added to the input using residual connection, and then layer normalization is applied to obtain a temporary representation X_{en}^{temp} . Then, the encoder performs a linear projection on X_{en}^{temp} to obtain the representation of previous outcome and treatment:

$$E_{R,T} = LN(FFN(X_{en}^{temp}) + X_{en}^{temp}) \quad (3)$$

where FFN indicates the linear feed-forward network. X_{en}^{temp} is residually connected with the linearly transformed result, and layer-normalized to get the final Encoding representations of previous outcome and treatment sequence $E_{R,A}$.

In AL-Transformer, the encoder is used to encode the patient's previous outcome and treatment and predict the patient's outcome sequence. The calculation process can be defined as:

$$R_{i,t} = E_{R,A}W_{en}^O \quad (4)$$

where $R_{i,t}$ indicates the outcome sequence of patient i at time t predicted by the encoder; $E_{R,A}$ indicates the encoding of previous outcome and treatment sequences; W_{en}^O indicates the weight matrix of linear layers.

Unlike most time series forecasting methods that use previous sequences of the same time series as input to predict future sequences, AL-Transformer uses outcome results generated by encoder with representation of previous outcome and treatment to predict treatment sequences. Specifically, the decoder in AL-Transformer uses the encoder's predicted outcome $R_{i,t}$ as input, and combines the encoding $E_{R,A}$ to predict the treatment sequence of time t . The specific input of the decoder can be defined as:

$$X_{de} = [R_{i,1}; \dots; R_{i,t}]W_{de}^{in} + PE \quad (5)$$

where W_{de}^{in} indicates the weight matrix of input linear transformation.

The calculation process of the decoder can be defined as:

$$X_{de}^{temp1} = LN(X_{de} + CA(X_{de}, X_{de}, X_{de})) \quad (6)$$

$$X_{de}^{temp2} = LN(X_{de}^{temp1} + CA(X_{de}^{temp1}, E_{R,T}, E_{R,T})) \quad (7)$$

$$E_R = LN(FFN(X_{de}^{temp2}) + X_{de}^{temp2}) \quad (8)$$

$$A_{i,t} = E_RW_{de}^O \quad (9)$$

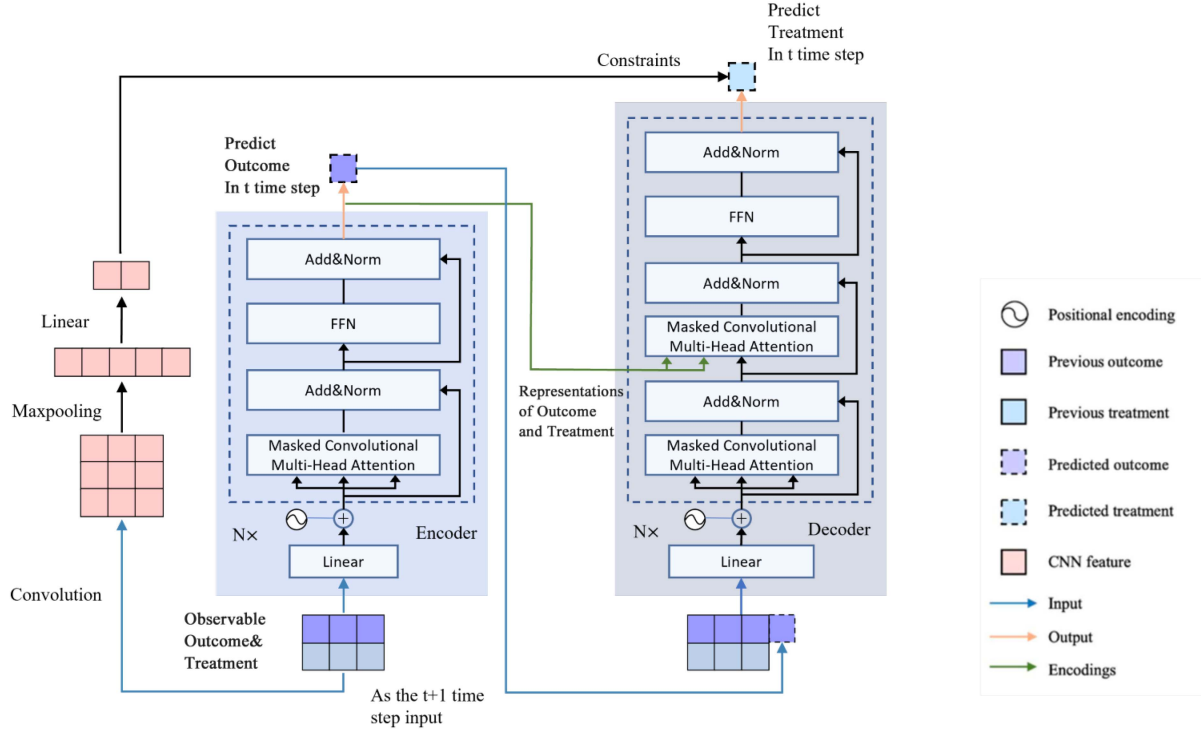


Fig. 2. Overview of the AL-Transformer model. We visualize the information flow in the AL-Transformer model. In the middle, we depict the encoder part of the model that uses observed variables to predict the outcome. In the right side, we describe the decoder that uses the representations obtained from the encoder and new overall variables to predict the treatment outcome. In the left side, we show the Sparsity constraints.

where E_R indicates the output encoding of the decoder through convolutional self-attention mechanism; W_{de}^O indicates the weight matrix of the output layer. $A_{i,t}$ is the treatment sequence predicted by the model at time t .

When the encoder and decoder in the AL-Transformer predict the patient's outcome sequence and treatment sequence at current time respectively, the predicted sequence can be used as a new input to the encoder for prediction, so that the process can be repeated continuously.

B. Causal Convolutional Self-Attention

In the AL-Transformer architecture, the self-attention mechanism used in the original Transformer model struggles with handling long sequences of data. This is because in time series, variables can evolve over time with equal or similar values due to various underlying events. To effectively distinguish between identical or similar points in a sequence, it is often necessary to consider the local context of sequence points. However, the scaled dot product self-attention mechanism used in the original Transformer model calculates the similarity between the query vector and key vector between sequence points, without considering the local context of each sequence point. As a result, this self-attention mechanism is not effective in distinguishing between identical or similar sequence points. To address this issue, we improve the causal convolutional self-attention mechanism proposed by Li et al. [35] and use it to replace the self-attention mechanism in the Transformer. The causal convolutional self-attention mechanism incorporates local information

from previous moments at each time point and can better handle sequence points with similar semantics.

The causal convolutional self-attention mechanism integrates local context information from the previous $k - 1$ time steps into the linear transformation result generated by the sequence point, enhancing the model's ability to capture the local shape of sequences when computing similarity. In contrast, the traditional self-attention mechanism performs linear projection on the input sequence S to obtain the query matrix Q , the key matrix K , and the value matrix V , which is equivalent to using a convolution operation with a kernel size of 1 on S . Since the causal convolution operation integrates only the sequence information before the sequence point, it does not destroy the temporal order relationship of the sequence. To calculate the causal convolutional self-attention, an appropriate padding length is first selected according to the size of the convolution kernel used in the subsequent convolution operation to pad the length of the input sequence. Then, the corresponding one-dimensional convolution operation is performed to obtain Q , K , and V integrated with local spatial information, followed by the scaled dot product self-attention calculation:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{\text{Conv}_k(QK^T)}{\sqrt{d_d}} \right) \text{Conv}_1(V) \quad (10)$$

C. Sparsity Constraints

In clinical scenarios, the treatment sequence is sparser compared to the outcome sequence, which makes it challenging to

model both outcome and treatment simultaneously. To address this issue and ensure that the treatment sequence predicted by AL-Transformer maintains sparsity, a classification model is proposed to constrain the output of the decoder. This model serves as a sparsity constraint, which generates a probability distribution for each treatment variable at the current time based on the patient's previous outcome and treatment. This allows the model to determine whether the patient requires treatment at the current time.

To construct the variable matrix $H(t-1) \in \mathbb{R}^{(n+m) \times w}$ in the sparsity constraint, we first stack the patient's outcome and treatment sequences and then concatenate them along the variable dimension. Here, n represents the number of variables in the outcome sequence, m represents the number of variables in the treatment sequence, and w represents the number of time steps used in the prediction. Specifically, the outcome and treatment data from the w time steps preceding time t are used to predict the probability distribution of each treatment variable. The convolution operation on the variable matrix $H(t-1)$ can be defined as:

$$H(t-1) = [R_{i,t-w-1} \oplus A_{i,t-w-1}; R_{i,t-1} \oplus A_{i,t-1}] \quad (11)$$

$$\text{Conv}_k^i(t-1) = H(t-1)_{:,k:k+l-1} \circ \text{Kenl} + b \quad (12)$$

$$f_k^i(t-1) = \text{BN}(\text{Conv}_k^i(t-1)) \quad (13)$$

$$S_{i,\text{Kenl}}(t-1) = \max_k f_k^i(t-1) \quad (14)$$

where $\text{Conv}_k^i(t-1)$ indicates the k -th element in the i -th feature map; l indicates the kernel size of the convolutional layer; $\text{Kenl} \in \mathbb{R}^{(n+m) \times l}$ indicates the size of filter; b indicates the bias; BN indicates the batch normalization operation.

After obtaining the final feature $S(t-1)$ through concatenating the feature maps obtained from convolution operations with different kernel widths and applying a linear transformation, the final treatment probability distribution $P(t)$ is computed. Since the treatment sequence at each time point consists of multiple variables, we apply a threshold thr to each element $P_{i,t,c}$ in the probability distribution $P(t)$ for each treatment variable c . If $P_{i,t,c}$ is greater than thr , it indicates that patient i needs treatment at time t for the corresponding treatment variable c . We have a discussion about how to determine the thresholds in the Section V. To penalize unnecessary treatments, we calculate the penalty coefficient I_t^c for treatment variable c as follows:

$$I_{i,t,c} = \begin{cases} 1 & \text{if } P_{i,t,c} \geq thr \\ 0 & \text{if } P_{i,t,c} < thr \end{cases} \quad (15)$$

Finally, the final penalty coefficient can be defined as:

$$I_{i,t} = I_{i,t,1} \oplus \dots \oplus I_{i,t,m} \quad (16)$$

where m indicates the number of variables in the treatment sequence.

Once the punishment coefficient of the treatment sequence has been obtained at time t , it can be used to constrain the predicted treatment data outputted by the decoder. The resulting treatment data at time t is defined as follows:

$$A'_{i,t} = I_{i,t} \circ A_{i,t} \quad (17)$$

where I_t indicates the penalty coefficients matrix; A_t indicates the treatment sequence predicted by the decoder in AL-Transformer at time t .

D. Loss Function

We train the alternating prediction model using two loss functions. 1) A Binary Cross Entropy (BCE) loss function that computes the loss between the probability distribution P_t predicted by the sparsity constraints model and the label of each treatment variable. 2) The Mean Squared Error (MSE) loss between the predicted value and the true value.

The BCE loss can be defined as:

$$L_{BCE} = \sum_{\sigma \in (i,t,c)} -[l_\sigma \log P_\sigma + (1-l_\sigma) \log(1-P_\sigma)] \quad (18)$$

where $P_{i,t,c}$ indicates the probability distribution of the treatment variable c of patient i at time t predicted by the sparsity constraints; $l_{i,t,c}$ indicates the real situation of the treatment variable c of patient i at time t .

The MSE loss of the outcome and the treatment sequence can be defined as:

$$L_{MSE}^R = \sum_i \sum_t \sum_m (R_{i,t,m} - y_{i,t,m}^R)^2 \quad (19)$$

$$L_{MSE}^A = \sum_i \sum_t \sum_m (A_{i,t,n} - y_{i,t,n}^R)^2 \quad (20)$$

where $R_{i,t,m}$ indicates the value of the variable m predicted by AL-Transformer at time t ; $y_{i,t,m}^R$ indicates the real value of the variable m in the outcome sequence at time t ; $A_{i,t,n}$ indicates the value of variable n predicted by AL-Transformer at time t ; $y_{i,t,n}^T$ indicates the true value of variable s at time t in the treatment sequence.

And the total loss L_t can be defined as:

$$L_t = L_{MSE}^R + L_{MSE}^A + L_{BCE} \quad (21)$$

It is worth noting that in computing the MSE loss value for the treatment sequence, we used the treatment sequence output by the AL-Transformer in training without the sparsity constraint.

IV. EXPERIMENT

A. Data

The Medical Information Mart for Intensive Care (MIMIC) database [14] is a large, freely-available database comprising deidentified data for patients admitted to intensive care units at the Beth Israel Deaconess Medical Center (BIDMC). MIMIC-III contains data associated with over forty thousand ICU patients between 2001 and 2012. It includes information such as demographics, vital sign measurements, laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality. For the task of treatment-outcome prediction, we extract two datasets for patients with sepsis and respiratory failure respectively from the MIMIC database. At each hour, if there are multiple measured physiological or clinical variables, measurements within that hour are averaged. Sepsis dataset contains data from 13,418 patients with sepsis according to the sepsis 3

criteria [36] and each patient has 47 outcome variables and 2 treatment variables corresponding to the hourly vasopressor and fluids amount. Data during the first 72-hours of each patient's ICU stay are extracted. Respiratory failure dataset consists of 5,783 patients who have been on mechanical ventilator for at least 24 hours in the ICUs. Each patient has 22 outcome variables and 3 treatment variables corresponding to ventilator settings. Data for each patient up to 48 hours are extracted. We divide the patients into 8:1:1, as training set, validation set and test set. In the comparison experiment, we adopt two prediction strategies, using the first 12 hours of data to predict the next 12 hours, and using the first 24 hours of data to predict the next 24 hours. The details of the variables information in two sub-dataset are shown in the Appendix.

For each individual variable of the patients, due to the different ranges of the data values, training the model on raw data would increase the difficulty of convergence. Therefore, we use normalization as a data preprocessing technique:

$$X = \frac{X - X_{mean}}{X_{std}} \quad (22)$$

B. Implementation

In the AL-Transformer, the feature dimension is set to 256, the number of self-attention network blocks of the encoder is set to 3; The number of self-attention blocks for the decoder is set to 1. In the causal convolutional self-attention used by AL-Transformer, the convolution kernel sizes w_q , w_k , w_v are respectively set as 3, 3, 1. Dropout rate is set to 0.1. The window of sparsity constraints w is set to 12, and the convolution kernel sizes of the three CNN blocks in its classification model are set to 3, 5, and 7 respectively.

All experiments presented in this section are implemented using the PyTorch deep learning framework. During the training process, the number of epochs for model training is set to 50, and early stopping is used to avoid overfitting of the model. The model was trained using the Adam optimizer with a learning rate of 0.0002 and a weight decay of 0.0005. The batch size for model training is set to 64. The code related to this paper can be found at Github.¹

C. Compared Methods

The compared methods are summarized as follows. 1) RNN: A recurrent neural network proposed by [37] 2) GRU: The recurrent gating unit proposed by [38] 3) LSTM: The long short-term memory network proposed by [39]. 4) AL-LSTM: A dual LSTM network designed in this subsection for performance comparison with alternating prediction models. The two LSTM networks model and predict the outcome and treatment sequence separately, but there is no interaction between these two LSTMs. 5) Transformer: A neural network based on a self-attention mechanism proposed by [40] 6) Informer: An improved sequence prediction model based on Transformer proposed by [41].

D. Results

The evaluation metrics for treatment and outcome prediction are Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Overall Geometric Mean. Geometric Mean indicates the central tendency of a set of numbers, so we utilize it as the overall metric for each method. Table I shows the detailed performance comparisons with the compared methods on the sepsis dataset.

The proposed AL-Transformer outperforms other approaches in most performance indices. In the 12-hour prediction task, compared with the baseline model AL-LSTM, the overall prediction error of AL-Transformer is significantly reduced by 29.95%. In the 24-hour prediction task, compared with the baseline model AL-LSTM, which is also the best baseline model, the overall prediction error of AL-Transformer is significantly reduced by 24.35%. It indicates that our model has better performance in longer time sequence than baselines. In addition, it can be seen that among all the models which can predict the outcome and treatment sequence simultaneously, the models based on the self-attention mechanism such as Transformer and Informer are better than recurrent neural networks such as RNN, GRU, and LSTM. Specifically, Transformer and Informer outperform recurrent neural networks on outcome prediction metrics and longer time sequence performance, but do not perform as well in treatment prediction metrics. This is due to the limited effectiveness of the self-attention mechanism in dealing with sparse variables. At the same time, it can be observed from Table I that AL-LSTM achieves the best overall performance among all the compared methods in 12 hours prediction. This suggests that alternate prediction mechanism can achieve better performance in short sequence modeling, while the self-attentive mechanism can achieve better results in longer sequences.

Table II presents the comparison results on the respiratory failure dataset. AL-Transformer exhibits the best results on most performance indices, with only a slight difference from Transformer in the MSE of treatment variables. Compared with Transformer, the best method on this metric, AL-Transformer reduces the error by 7.58%. Moreover, our model significantly outperforms the baseline model LSTM, reducing the overall prediction error by 26.11%. The results indicate that self-attention-based models have better effects on sequences with fewer outcome variables, and the models based on self-attention mechanism perform better than recurrent neural networks on most metrics. For 24-hour prediction, AL-Transformer shows a remarkable improvement over LSTM with a significant reduction in overall prediction error by 37.93%. Compared with Transformer, the best baseline method on overall error, our model reduced the error by 6.42%. Specifically, AL-Transformer outperforms LSTM by 22.56% in MAE and 29.38% in MSE for outcome sequence, and by 32.08% in MAE and 60.05% in MSE for treatment. Additionally, compared with the best baseline method for each metric, AL-Transformer reduces the MAE and MSE of outcome by 4.14% and 4.89%, respectively, and the MAE and MSE of treatment by 11.91% and 7.51%, respectively, compared to Transformer. Our experiments also demonstrate that self-attention-based models, including Transformer, Informer, and AL-Transformer, outperform traditional recurrent neural

¹[Online]. Available: <https://github.com/meiyoufeng116/AL-Transformer>

TABLE I
COMPARISON RESULTS ON THE SEPSIS DATASET

Methods	12 Hours Prediction					24 Hours Prediction				
	Outcome		Treatment		Overall	Outcome		Treatment		Overall
	MAE	MSE	MAE	MSE	Geometric Mean	MAE	MSE	MAE	MSE	Geometric Mean
RNN	0.5229	0.6313	0.1027	0.0349	0.1854	0.5600	0.6872	0.0893	0.0343	0.1852
GRU	0.4424	0.4810	0.1058	0.0411	0.1744	0.4490	0.5015	0.1208	0.0434	0.1854
LSTM	0.5274	0.6358	0.0733	0.0346	0.1708	0.5121	0.5635	0.0787	0.0339	0.1666
AL-LSTM	0.4527	0.5250	0.0649	0.0329	0.1496	0.4371	0.4897	0.0519	0.0287	0.1337
Transformer	<u>0.3906</u>	<u>0.4336</u>	0.0994	0.0408	0.1619	<u>0.3620</u>	<u>0.3787</u>	0.0846	0.0413	0.1480
Informer	0.3995	0.4391	0.1028	0.0366	0.1603	0.3701	<u>0.3761</u>	0.0820	0.0351	0.1415
Our	0.3132	0.3388	0.0424	0.0268	0.1048	0.3116	0.3148	0.0372	0.0285	0.1009
vs. AL-LSTM	30.82% ↓	35.47% ↓	34.67% ↓	18.54% ↓	29.95% ↓	28.71% ↓	35.72% ↓	28.32% ↓	0.70% ↓	24.35% ↓
vs. Transformer	19.82% ↓	21.86% ↓	57.34% ↓	34.31% ↓	35.27% ↓	13.92% ↓	16.87% ↓	56.03% ↓	30.99% ↓	31.82% ↓
vs. Best Baseline	19.82% ↓	21.86% ↓	34.67% ↓	18.54% ↓	29.95% ↓	13.92% ↓	16.30% ↓	28.32% ↓	0.70% ↓	24.35% ↓

Best performing result in bold, and the second best is underscored. The down arrow indicates the performance gain obtained by our method compared to the baseline method.

TABLE II
COMPARISON RESULTS ON RESPIRATORY FAILURE DATASET

Methods	12 Hours Prediction					24 Hours Prediction				
	Outcome		Treatment		Overall	Outcome		Treatment		Overall
	MAE	MSE	MAE	MSE	Geometric Mean	MAE	MSE	MAE	MSE	Geometric Mean
RNN	0.4651	0.5217	0.3283	1.1420	0.5492	0.4903	0.5492	0.3678	0.7212	0.5170
GRU	0.4223	0.4617	<u>0.2807</u>	1.0443	0.4890	0.4339	0.4665	0.3324	0.6773	0.4620
LSTM	0.4739	0.5432	0.3937	1.1511	0.5844	0.4873	0.5685	0.4116	0.6965	0.5309
AL-LSTM	0.4324	0.4701	0.3862	1.1679	0.5503	0.4592	0.5166	0.4570	0.8583	0.5523
Transformer	0.3880	0.4119	0.3242	0.9351	<u>0.4692</u>	<u>0.3937</u>	<u>0.4212</u>	<u>0.3129</u>	<u>0.3008</u>	<u>0.3534</u>
Informer	0.4036	<u>0.4118</u>	0.3549	0.9962	0.4924	0.4042	0.4419	0.3363	0.3221	0.3729
Our	0.3767	0.3919	0.2501	0.9421	0.4318	0.3774	0.4015	0.2796	0.2782	0.3295
vs. AL-LSTM	12.89% ↓	16.64% ↓	35.25% ↓	19.33% ↓	21.53% ↓	17.81% ↓	22.28% ↓	38.82% ↓	67.58% ↓	40.34% ↓
vs. Transformer	2.91% ↓	4.86% ↓	22.86% ↓	0.75% ↑	7.97% ↓	4.14% ↓	4.68% ↓	10.64% ↓	7.51% ↓	6.76% ↓
vs. Best Baseline	2.91% ↓	4.83% ↓	7.93% ↓	0.75% ↑	7.58% ↓	4.14% ↓	4.89% ↓	11.91% ↓	7.51% ↓	6.42% ↓

Best performing result in bold, and the second best is underscored. The down arrow means the performance gain obtained by our method compared to the baseline method.

TABLE III
QUANTITATIVE RESULTS OF ABLATION STUDY ON SEPSIS DATASET

Component	12 Hours								24Hours					
	Convolutional		Sparsity		Outcome		Treatment		Overall	Outcome		Treatment		Overall
	self-attention	constraints	MAE	MSE	MAE	MSE	MAE	MSE	Geometric Mean	MAE	MSE	MAE	MSE	Geometric Mean
✓			0.3240	0.3516	0.0883	0.0250	0.1234	0.1234	0.3588	0.3587	0.0524	0.0276	0.1168	
✓	✓		0.3399	0.3638	0.0637	0.0250	0.1184	0.1184	0.3519	0.3551	0.0622	0.0299	0.1234	
✓		✓	0.3226	0.3492	0.0436	0.0243	0.1043	0.1043	0.3598	0.3608	0.0381	0.0283	0.1088	
✓	✓	✓	0.3081	0.3344	0.0427	0.0251	0.1024	0.1024	0.3233	0.3276	0.0355	0.0239	0.0973	

networks such as RNN, GRU, and LSTM, in non-sparse data. Furthermore, our model shows better performance than Transformer and Informer for 24-hour prediction, indicating that the Causal Convolution Module has a stronger impact over longer time sequences.

V. DISCUSSION

A. Ablation Study.

The proposed alternating prediction model in this section consists of three main components: AL-Transformer, causal convolutional self-attention, and sparsity constraints. To evaluate the impact of each component on model performance, we

conducted an ablation analysis on AL-Transformer with different components. The results of the analysis on the sepsis dataset are shown in Table III, where the geometric mean represents the overall prediction error of the model.

Table III clearly shows that the more components are incorporated into AL-Transformer, the better its overall performance becomes. With regard to the causal convolutional self-attention mechanism, the comparison reveals that integrating it into AL-Transformer leads to a slight increase in MAEs of outcomes by 4.9% (12 hours) and 1.91% (24 hours) compared to using only AL-Transformer for prediction. Meanwhile, MAEs of treatment are reduced by 27.8% (12 hours) and increased by 18.77% (24 hours), and the geometric means are reduced by 4.07%

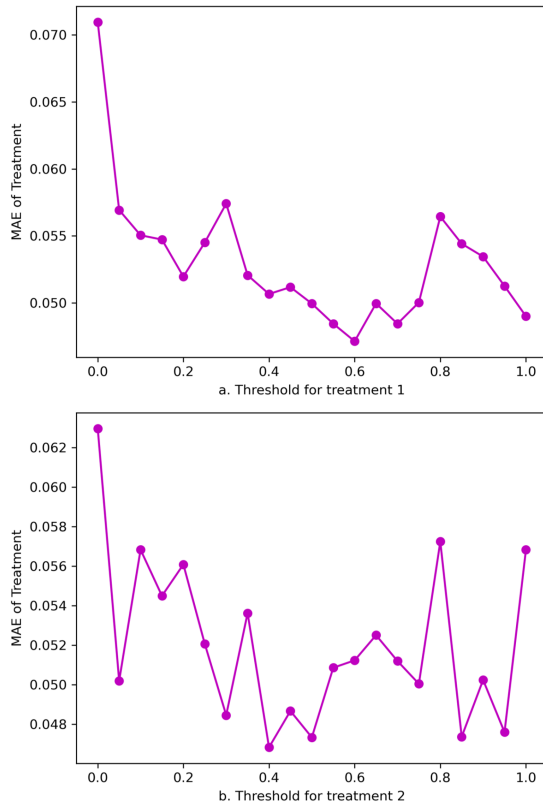


Fig. 3. Discussing the impact of different treatment thresholds on the 24-hour prediction in the sepsis dataset. In (a) The threshold of treatment 2 is set to 0.4. In (b) The threshold of treatment 1 is set to 0.6.

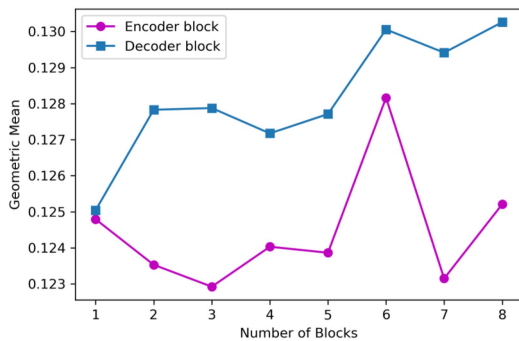


Fig. 4. Impact of the number of self-attention blocks on the performance of the AL-Transformer encoder and decoder in the 24-hour prediction of the sepsis dataset.

(12 hours) and increased by 5.63% (24 hours). This suggests that incorporating sequential local contextual information into the self-attention mechanism can somewhat improve the model's predictive ability for treatment. Furthermore, the performance of the model is significantly enhanced when the sparsity constraint is included. This may be due to the sparsity constraint suppressing the errors present in the convolutional self-attention mechanism when making long-range predictions.

As for the effectiveness of the sparsity constraint, it can be found by comparison that when the AL-Transformer is matched

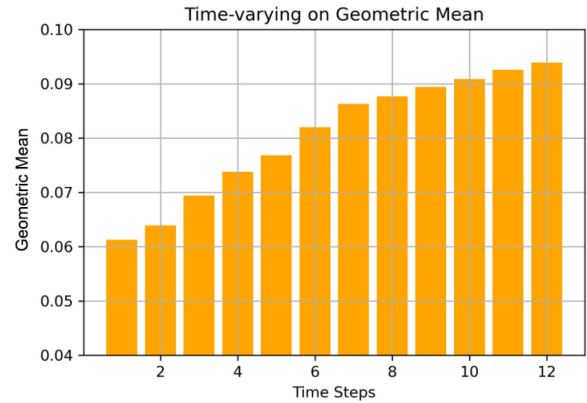


Fig. 5. Discussion on the performance of the model for different time steps in the first 12 hours of the 24-hour prediction on the sepsis dataset.

with the sparsity constraint, compared with only using the AL-Transformer for prediction, the MAE of outcome shows slightly changed, and the MAE of treatment is significantly reduced by 50.66% (12 hours) and 21.17% (24 hours), the geometric mean is reduced by 15.5% (12 hours) and 6.9% (24 hours). These two comparative results demonstrate the effectiveness of the sparsity constraint in our model. Besides, it can be observed that after using the sparsity constraint, the MAE of outcome has slightly changed, because the sparsity constraint is only proposed to ensure the sparsity of the treatment data.

B. Threshold of Sparsity Constraints

In most classification tasks, a threshold of 0.5 is commonly used. However, in some clinical tasks, with the granularity of hours, patients may not require treatment most of the time, leading to a highly imbalanced distribution of positive and negative treatment samples. To explore the impact of different thresholds on the model's performance, we evaluated the effect of varying the classification threshold for treatment variables on sequence prediction. Fig. 3 illustrates the effect of different thresholds for treatment variables 1 and 2 on the MAE for the treatment sequence in sepsis dataset's validation set. The test set remained unseen during this step, and we determined the optimal thresholds based on the MAE on the validation set. We observed that the MAE is highest when the threshold values for both variables are set to 1 or 0, and it decreases as the threshold values decrease to about 0.5. The optimal combination of thresholds is found to be [0.6, 0.4]. Importantly, when both variables have a threshold of 1, the sparsity constraint has no effect on the model. Therefore, our proposed sparsity constraint can improve the model's performance in predicting treatment variables.

C. Number of encoder/decoder Blocks

To discussion the influence of the number of encoder and decoder blocks, we set the number of decoder blocks to 1 to investigate the impact of different numbers of encoder blocks on the results. Similarly, we set the number of encoder blocks to

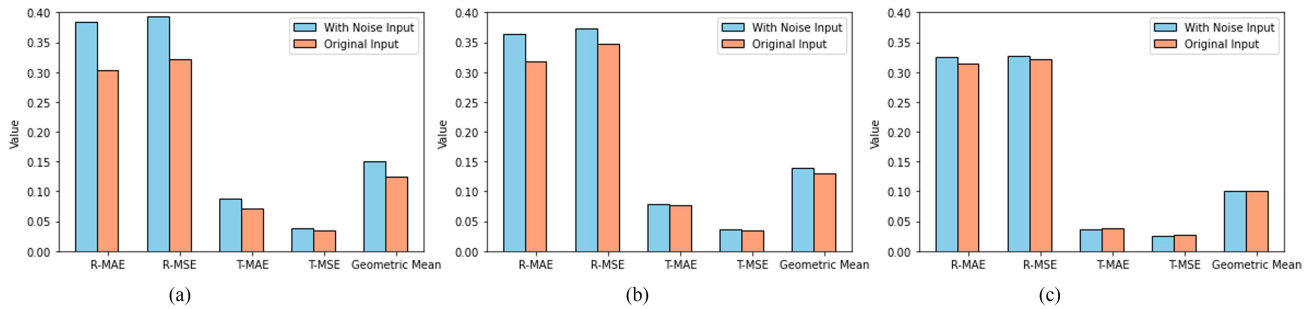


Fig. 6. Results of noise experiment in Transformer, Informer and AL-Transformer, using normal distribution in input data with a mean of 0 and a standard deviation of 0.5 as the noise. (a) Transformer, (b) Informer, and (c) AL-Transformer.

1 to investigate the effect of different numbers of decoder blocks on the results. As can be seen from Fig. 4, with the increasing number of decoder blocks, the geometric mean exhibits an increasing trend. Conversely, the increase in the number of encoder blocks has a smaller effect on the geometric mean, which means that the model is not sensitive to the number of encoder blocks compared to the number of decoder blocks. This is because more blocks would increase the number of parameters, leading to a higher risk of overfitting. Additionally, more blocks may cause the model to capture the noise in the data, thereby reducing its performance.

D. Prediction Performance on Different Time Steps

We conducted an analysis of the geometric mean values of outcome and treatment sequences for 12 future time steps in Spesis dataset 24 hours prediction. The results are shown in Fig. 5. We observed that the geometric mean value initially increased with the increase of the time step to be predicted, and eventually tended to stabilize. Specifically, when predicting the value of the first time step, the geometric mean value was 0.0612. However, when predicting the value of the 12th time step, the geometric mean value increased by 53.43% to 0.0939. This analysis provides useful insights for doctors in real scenarios, allowing them to select more reliable prediction results to assist with the diagnosis and treatment of patients.

E. Performance With Noisy Data

To explore the performance of the model with noisy data, we add Gaussian Noise to the test dataset and perform a large number of simulations. Specifically, we add normal distributed noise to the input data and perform one hundred experiments. The results of each experiment are compared with the ground truth, to calculate the MAE, MSE, and geometric mean of treatment and outcome. Fig. 6 shows the result of Transformer, Informer and AL-Transformer performance with Gaussian noise input in 100 simulation. As we can see, in general, in outcome metrics, the input with noise has significant impacts on Transformer and Informer. The R-MAE and R-MSE of Transformer are decreased by 20.88% and 18.44%, respectively. And R-MAE and R-MSE of Informer are decreased by 12.68% and 6.96%, respectively. On the other hand, AL-Transformer is slightly influenced by the noise data, only reducing 3.3% and 1.9% in R-MAE and R-MSE. In treatment metrics, Transformer has weak performances in

the noise data situation, the T-MSE and T-MAE drop by 19% and 10.14%. On the contrary, the noise data does not have a significant impact on Informer and AL-Transformer. It is noted that the performance of treatment MSE increases by 8.77% in the noise input case, which may be due to the noise data suppressing the number of outlier points generated by the model. It also proves that the alternating sequence model suppresses the accumulation of errors in the sequence to a certain extent. In conclusion, compared to the other self-attention based model, the alternating sequence modeling brings better robustness for AL-Transformer within the noisy data case.

VI. CONCLUSION AND FUTURE WORK

This paper presents a self-attention based approach to model patient treatments and outcomes under time-varying and sequential dynamic treatment strategies. We use AL-Transformer to simultaneously model outcomes and treatments to capture dependencies between them, and to alternately predict outcome and treatment sequences. Causal convolutional self-attention is used in the AL-Transformer to enhance the temporal information of sequential data. Additionally, we propose sparsity constraints to constrain the sparse treatment output. Extensive experiments demonstrate the effectiveness of our model, and an ablation study verifies the contribution of sparsity constraints and convolutional self-attention to model performance.

While our current study is focusing on modeling the patients' trajectories and treatment jointly, there still has some limitations in our pipeline. For example, We did not individually consider some time-invariant covariates such as age, gender, comorbidities, etc and treat them as a time-varying variable. While predicting these covariates may not be meaningful, we overlooked their significant role in treatment prediction. Additionally, our model did not quantify the uncertainty in the prediction process, which could aid medical professionals in making informed decisions. For future work, we plan to investigate approaches for quantifying model uncertainty as well as modeling of missing data in clinical time-series measurements for more robust treatment and outcome prediction. We also plan to design a separate encoding module to extract information present in the covariates and integrate it into our encoder. By doing so, we aim to utilize the previously overlooked information and enhance the model's ability to effectively model patient data.

REFERENCES

- [1] S. Purushotham et al., "Benchmarking deep learning models on large healthcare datasets," *J. Biomed. Inform.*, vol. 83, pp. 112–134, 2018.
- [2] H. Harutyunyan et al., "Multitask learning and benchmarking with clinical time series data," *Sci. Data*, vol. 6, no. 1, 2019, Art. no. 96.
- [3] Z. Che et al., "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 6085.
- [4] H. Song et al., "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4091–4098.
- [5] J. Oh, J. Wang, and J. Wiens, "Learning to exploit invariances in clinical time-series data using sequence transformer networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2018, pp. 332–347.
- [6] A. Radford et al., "Improving language understanding by generative pre-training," 2018.
- [7] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Ch. Assoc. Comput. Linguist.: Human Lang. Technol.*, 2019, pp. 4171–4186.
- [8] J. Futoma et al., "Predicting disease progression with a model for multivariate longitudinal clinical data," in *Proc. 1st Mach. Learn. Healthcare Conf.*, 2016, pp. 42–54.
- [9] L. w. H. Lehman et al., "A physiological time series dynamics-based approach to patient monitoring and outcome prediction," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 3, pp. 1068–1076, May 2015.
- [10] O. Ren et al., "Predicting and understanding unexpected respiratory decompensation in critical care using sparse and heterogeneous clinical data," in *Proc. IEEE Int. Conf. Healthcare Inform.*, 2018, pp. 144–151.
- [11] A. Hüyük et al., "Explaining by imitating: Understanding decisions by interpretable policy learning," in *Proc. Int. Conf. Learn. Representations*, 2020.
- [12] A. Pace, A. Chan, and M. van der Schaar, "POETREE: Interpretable policy learning with adaptive decision trees," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [13] A. Dejl et al., "Recurrent sum-product networks for sequential treatment regimes," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop Time Ser. Health*, 2022.
- [14] A. E. Johnson et al., "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, 2016, Art. no. 160035.
- [15] X. Wang, D. Sontag, and F. Wang, "Unsupervised learning of disease progression models," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 85–94.
- [16] J. Futoma et al., "An improved multi-output gaussian process RNN with real-time validation for early sepsis detection," in *Proc. Mach. Learn. Healthcare Conf.*, 2017, pp. 243–254.
- [17] J. R. Gardner et al., "GPYtorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7587–7597.
- [18] X. Zhang et al., "INPREM: An interpretable and trustworthy predictive model for healthcare," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 450–460, doi: [10.1145/3394486.3403087](https://doi.org/10.1145/3394486.3403087).
- [19] L. Rasmy et al., "Med-bert: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction," *NPJ Digit. Med.*, vol. 4, no. 1, 2021, Art. no. 86.
- [20] S. Tipirneni and C. K. Reddy, "Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 6, pp. 1–17, Jul. 2022, doi: [10.1145/3516367](https://doi.org/10.1145/3516367).
- [21] Y. Xu, Y. Xu, and S. Saria, "A Bayesian nonparametric approach for estimating individualized treatment-response curves," in *Proc. Mach. Learn. Healthcare Conf.*, 2016, pp. 282–300.
- [22] H. Soleimani, A. Subbaswamy, and S. Saria, "Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions," in *Proc. Uncertainty Artif. Intell.*, 2017.
- [23] B. Lim, A. Alaa, and M. Van der Schaar, "Forecasting treatment responses over time using recurrent marginal structural networks," in *Proc. Neural Inf. Process. Syst.*, 2018.
- [24] I. Bica et al., "Estimating counterfactual treatment outcomes over time through adversarially balanced representations," in *Proc. Int. Conf. Learn. Representations*, 2020.
- [25] I. Bica, A. M. Alaa, and M. van der Schaar, "Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 884–895.
- [26] R. Li et al., "G-net: A recurrent network approach to G-computation for counterfactual prediction under a dynamic treatment regime," in *Proc. Mach. Learn. Health*, 2021, pp. 282–299.
- [27] V. Melnychuk, D. Frauen, and S. Feuerriegel, "Causal transformer for estimating counterfactual outcomes," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 15293–15329.
- [28] A. Rajkomar et al., "Scalable and accurate deep learning with electronic health records," *NPJ Digit. Med.*, vol. 1, no. 1, 2018, Art. no. 18.
- [29] L. Liu et al., "Learning the joint representation of heterogeneous temporal events for clinical endpoint prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 109–116.
- [30] M. Komorowski et al., "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," *Nature Med.*, vol. 24, no. 11, 2018, Art. no. 1716.
- [31] J. Oh, J. Wang, S. Tang, M. W. Sjoding, and J. Wiens, "Relaxed parameter sharing: Effectively modeling time-varying relationships in clinical time-series," in *Proc. 4th Mach. Learn. Healthcare Conf.*, 2019, pp. 27–52. [Online]. Available: <http://proceedings.mlr.press/v106/oh19a.html>
- [32] Y. Xu et al., "Raim: Recurrent attentive and intensive model of multimodal patient monitoring data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2565–2573.
- [33] Y. Zhou et al., "A contrastive learning approach for ICU false arrhythmia alarm reduction," *Sci. Rep.*, vol. 12, no. 1, 2022, Art. no. 4689.
- [34] F. Wu et al., "A diffusion model with contrastive learning for ICU false arrhythmia alarm reduction," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, 2023, pp. 4912–4920.
- [35] S. Li et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019.
- [36] M. Singer et al., "The third international consensus definitions for sepsis and septic shock (sepsis-3)," *Jama*, vol. 315, no. 8, pp. 801–810, 2016.
- [37] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.
- [38] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
- [39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [40] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.
- [41] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.