Learning Time-shared Hidden Heterogeneity for Counterfactual Outcome Forecast

Anonymous authors
Paper under double-blind review

Abstract

Forecasting counterfactual outcome in the longitudinal setting can be critical for many time-related applications. To solve this problem, the previous works propose to apply different sequence models including long short-term memory (LSTM) networks and transformers to model the relationship between the observed histories, treatments and outcomes, and apply various approaches to remove treatment selection bias. However, these methods neglect the hidden heterogeneity of outcome generation among samples induced by hidden factors which can bring hurdles to counterfactual outcome forecast. To alleviate this problem, we capture the hidden heterogeneity by recovering the hidden factors and incorporate it into the outcome prediction process. Specifically, we propose a Time-shared Heterogeneity Learning from Time Series (THLTS) method which infers the shared part of hidden factors characterizing the heterogeneity across time steps with the architecture of variational encoders (VAE). This method can be a flexible component and combined with arbitrary counterfactual outcome forecast method. Experimental results on (semi-)synthetic datasets demonstrate that combined with our method, the mainstream models can improve their performance.

1 Introduction

Decision-making problems are widely prevalent in many applications, such as healthcare (Huang & Ning, 2012) and marketing (Bottou et al., 2013). Therefore, it is of paramount importance to forecast the counterfactual outcome for different choice of treatments to assist decision. The gold standard for estimating the outcome of different treatments is conducting randomized controlled trials (RCTs) (Booth & Tannock, 2014), which randomly assign treatments to the samples. However, the high expense and time cost of RCTs (Kohavi & Longbotham, 2011) induce the people to instead learning from large amounts of observational data to fulfill this purpose. A lot of previous literature investigate the problem of counterfactual outcome forecast based on the observational dataset with different approaches to address the treatment selection bias (Hassanpour & Greiner, 2019; Assaad et al., 2021), such as treatment invariant representation learning (Johansson et al., 2016; Shalit et al., 2017; Tanimoto et al., 2021; Schwab et al., 2020; Yao et al., 2018; Zeng et al., 2020), sample re-weighting for adjusting distributions (Assaad et al., 2021; Hassanpour & Greiner, 2019; 2020; Johansson et al., 2018; Zou et al., 2020) and data imputation (Bica et al., 2020c; Yoon et al., 2018; Qian et al., 2021).

In many scenarios, the decision-making problems can be more complex and may span a long period of time. It is thus required to forecast the counterfactual outcome at different time steps instead of a single time. Since the size of historical covariate information is varied among the time steps, the methods developed under the static setting can not be directly applied to this setting. To bridge this gap induced by the longitudinal property of the task, some methods apply the sequence models, such as LSTM networks (Hochreiter & Schmidhuber, 1997) and transformers (Vaswani et al., 2017), to characterize the time-dependency between the histories of varying length and outcomes. Generally, a representation is extracted from the histories and play a substitute role of the raw confounder vectors in the static setting. The outcome prediction module forecasts the counterfactual outcome with the learned representation and the counterfactual treatment. Based on this design, the technologies for removing treatment selection bias, can be integrated to achieve more accurate outcome forecast in the setting of time-series.

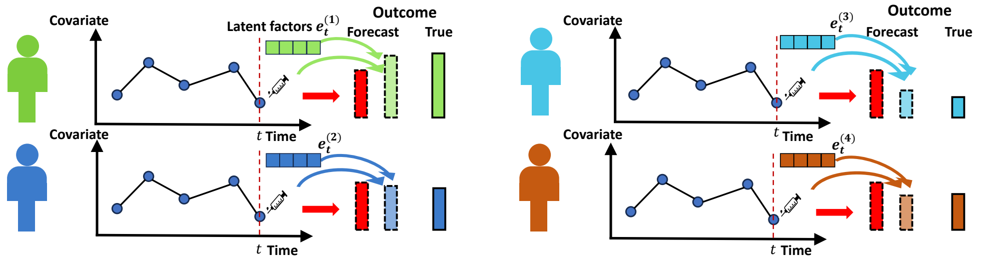

Figure 1: The diagram of counterfactual outcome forecast with hidden heterogeneity. Although the observed histories of the four individuals are same, the true counterfactual outcomes of them are in significantly distinct due to the latent factors. Predicting based on solely the same history results in the same prediction which brings hurdle to the performance. By uncovering latent factors capturing the hidden heterogeneity, we can achieve more precious prediction result.

The paradigm above attributes the heterogeneity of outcome generation among samples to the observed variables of histories in dataset. This hypothesizes that the outcome-related factors are all recorded in the histories. However, in many scenarios, this prerequisite does not hold. There may exist extra factors unrecorded by the observed histories, and can also affect outcome. Thereby, forecasting heterogeneous outcome conditional on solely the observed histories may neglect the outcome variation among samples (i.e. heterogeneity of outcome generation) of the same observed variables. Consequently, the neglect of this outcome variation can lead the prediction to be a coarse approximation of individual outcome and deteriorate the forecast performance (Zou et al., 2023). We call this problem as hidden heterogeneity and visually demonstrate it in Figure 1.

To resolve the problem, we try to address the hidden heterogeneity problem by uncovering the hidden factors and capturing the extra outcome variation. Due to the limited supervision information (i.e. outcome variable is of few-dimension) in this problem, the solution by aggressively learning latent factors for all samples and time steps is excessively flexible and undergoes sub-optimal performance, which is empirically presented in the experiments. To mitigate this circumstance, we propose a novel Time-shared Heterogeneity Learning from Time Series (THLTS) method, which instead learn the shared part of latent factors across time steps for each sample. While sacrificing the flexibility in modeling time-varying dynamics of latent factors, this design is targeting the pursuit of forecast performance like a regularizer.

For practical implementations, we resort to variational autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014) to model the joint distribution of the outcome and the time-shared latent factors given the observed history and treatments. Due to the longitudinal property of the problem, we extend the model architecture to support the inference of latent factors with the varying prior, which are updated according to the successive observation over time. With the encoder component, we can easily infer the latent factors. By incorporating the learned factors into outcome prediction module, we can forecast the individual outcome more preciously compared to the same model-backbone ignoring the hidden heterogeneity. Theoretical analysis validate the rationality of our proposed strategy to learn time-shared latent factors. We conduct extensive experiments on the synthetic datasets and semi-synthetic datasets, where the results reveal the effectiveness of our method.

The main contribution of our paper can be summarized as following:

- This paper investigate counterfactual outcome forecast with the existence of hidden heterogeneity. To the best of our knowledge, this is the pioneer work tailored to improving the off-the-shelf counterfactual forecast model by addressing the hidden heterogeneity problem.
- We expose the insightful idea that learning the shared part of latent factors over time steps and propose a novel Time-shared Heterogeneity Learning from Time Series (THLTS) method.
- Extensive experimental results demonstrate that our method acts like a flexible component and is beneficial for the mainstream counterfactual outcome forecast models when be integrated into them.

2 Related Works

We respectively review the related literature under both static setting and longitudinal setting.

2.1 Counterfactual Outcome Prediction in the Static Setting

There have been a large amount of works devoted to counterfactual outcome forecast in the static setting. The main challenge is the treatment selection bias manifested as the dependency of treatment assignment on the observed confounders. To overcome this challenge, some papers (Johansson et al., 2016; Shalit et al., 2017; Tanimoto et al., 2021; Schwab et al., 2020; Yao et al., 2018; Zeng et al., 2020) borrow the idea of domain adaptation (Tzeng et al., 2014; Ganin & Lempitsky, 2015) to learn the treatment invariant representation of confounders and predict counterfactual outcome by taking the representation as input. Since over-forcing the independency may bring hurdles to prediction performance (Assaad et al., 2021), some other methods (Assaad et al., 2021; Hassanpour & Greiner, 2019; 2020; Johansson et al., 2018; Zou et al., 2020) re-weight samples to adjust the joint distribution of confounders and treatments, and train the countefactual predictive model on the re-weighted dataset. Moreover, there are also some works perform data augmentation to imputes the counterfactual outcome for the observational samples (Bica et al., 2020c; Yoon et al., 2018; Qian et al., 2021). When faced with unobserved confounders, the prediction result may suffer from severe confounding bias. The previous literature resort to extra tools, such as instrumental variables (IVs) and negative controls (Hartford et al., 2017; Heckman, 1997; Wu et al., 2022) to overcome it. Since the assumptions on these tools are restricted, other methods (Louizos et al., 2017; Wang & Blei, 2019; Zou et al., 2023) try to recover the information of unobserved confounders according to the knowledge of data generation process.

2.2 Counterfactual Outcome Forecast in Time Series

In many applications, it is required to forecast counterfactual outcome for a period of time. Under these scenarios, the outcome of a specific time is determined by not only the observations at the present time but also the whole histories. To handle the information of long history, Robins et al. (2000) conduct linear/logistic regression on the truncated history to predict the outcome. Due to the potential complex dependency of outcome on covariates and treatments, previous works (Lim et al., 2018; Bica et al., 2020b; Li et al., 2020; Melnychuk et al., 2022; Hatt & Feuerriegel, 2021; Bica et al., 2020a; Kuzmanovic et al., 2021) propose to utilize sequence models (Chung et al., 2014; Vaswani et al., 2017; Hochreiter & Schmidhuber, 1997) to transform the histories of varying length to representations of fixed size and forecast outcome based on it. Lim et al. (2018) utilize long short-term memory (LSTM) networks to encode the histories and re-weights the samples by the estimated propensity scores for removing selection bias. Alternatively, Bica et al. (2020b) resort to impose treatment invariant representation regularizer on the represention learned by LSTM networks to remove treatment selection bias. Since the ability of LSTM networks to model complex and long dependencies in time is limited, Melnychuk et al. (2022) build the model based on transformers (Vaswani et al., 2017) and propose a novel counterfactual domain confusion (CDC) loss for training the models. Bouchattaoui et al. (2023) is a similar work to ours. However, this work designed a fixed model architecture which can only deal with the problem setting of binary treatment. By contrast, our proposed THLTS method can be viewed as a flexible component which is able to be combined with various model backbones of counterfactual outcome forecast and deal with more complex treatment scenarios. Moreover, this paper also consider the setting of time-varying latent factors and delivers the insightful strategy of learning shared part of latent factors across time steps to deal with it.

3 Problem Formulations and Statements

We define $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^{d_x}$ as the observed covariate vector which records the individual information of each sample, $\mathbf{A} \in \mathcal{A}$ as the treatment and $\mathbf{Y} \in \mathbb{R}$ as outcome. Therefore, the observational datasets is denoted as $\mathcal{D} = \{\{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)}\}_{t=1}^{T^{(i)}}\}_{i=1}^{n}$, where the superscripts and subscripts refer to the sample indexs and time indexs respectively. For the sake of description, we abbreviate the history of individuals by $\mathbf{X}_{t:t+\tau}^{(i)} = (\mathbf{x}_t^{(i)}, \mathbf{x}_{t+1}^{(i)}, \mathbf{x}_{t+2}^{(i)}, ..., \mathbf{x}_{t+\tau}^{(i)})$, $\mathbf{A}_{t:t+\tau}^{(i)} = (\mathbf{a}_t^{(i)}, \mathbf{a}_{t+1}^{(i)}, \mathbf{a}_{t+2}^{(i)}, ..., \mathbf{a}_{t+\tau}^{(i)})$, $\mathbf{Y}_{t:t+\tau}^{(i)} = (\mathbf{y}_t^{(i)}, \mathbf{y}_{t+1}^{(i)}, \mathbf{y}_{t+2}^{(i)}, ..., \mathbf{y}_{t+\tau}^{(i)})$ and $\mathbf{H}_t^{(i)} = \{\mathbf{X}_{1:t}^{(i)}, \mathbf{A}_{1:t-1}^{(i)}, \mathbf{Y}_{1:t-1}^{(i)}\}$.

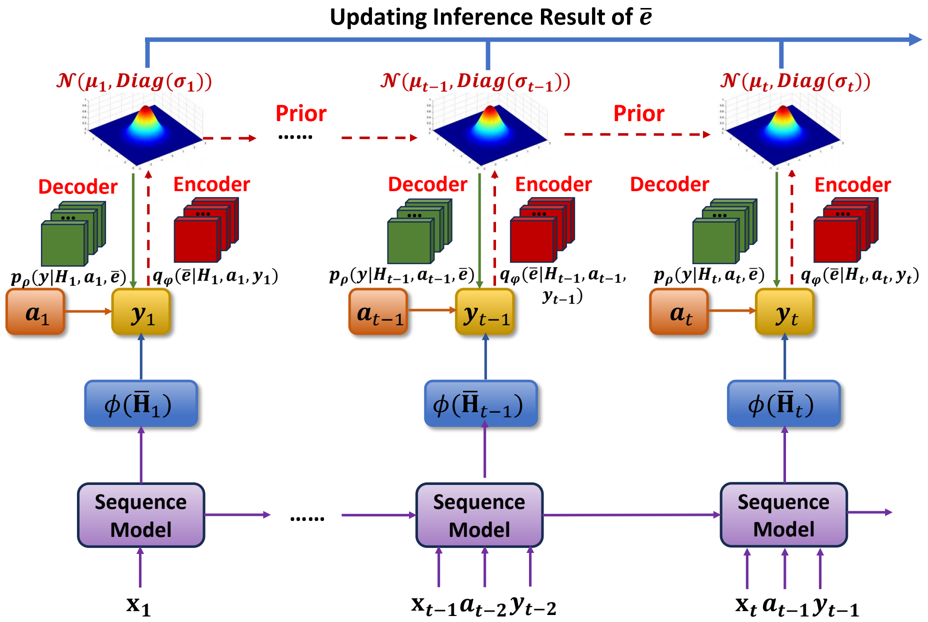
Updating Inference Result of $\bar{e}$
$\mathcal{N}(\mu_1, Diag(\sigma_1))$
$\mathcal{N}(\mu_{t-1}, Diag(\sigma_{t-1}))$
$\mathcal{N}(\mu_t, Diag(\sigma_t))$
Prior
Prior
Decoder
Encoder
Decoder
Encoder
Decoder
Encoder
$p_\rho(y|H_1, a_1, \bar{e})$
$q_\varphi(\bar{e}|H_1, a_1, y_1)$
$p_\rho(y|H_{t-1}, a_{t-1}, \bar{e})$
$q_\varphi(\bar{e}|H_{t-1}, a_{t-1}, y_{t-1})$
$p_\rho(y|H_t, a_t, \bar{e})$
$q_\varphi(\bar{e}|H_t, a_t, y_t)$
a_1
y_1
a_{t-1}
y_{t-1}
a_t
y_t
$\phi(\bar{\mathbf{H}}_1)$
$\phi(\bar{\mathbf{H}}_{t-1})$
$\phi(\bar{\mathbf{H}}_t)$
Sequence Model
Sequence Model
Sequence Model
$\mathbf{x}_1$
$\mathbf{x}_{t-1} a_{t-2} y_{t-2}$
$\mathbf{x}_t a_{t-1} y_{t-1}$

Figure 2: The diagram of time-shared heterogeneity learning framework.

The ultimate target is the outcome function at individual-level $\mathbf{Y}^{(i)}_{t+\tau}(\mathbf{A}^c_{t:t+\tau})$, which refers to the counterfactual outcome of the i^{th} sample at the $(t+\tau)^{th}$ time step under the counterfactual treatments $\mathbf{A}^c_{t:t+\tau}$. Many methods have been proposed in the previous literature to estimate the expected outcome given counterfactual treatments and histories, that is $\mathbb{E}[\mathbf{Y}_{t+\tau}|do(\mathbf{A}_{t:t+\tau} = \mathbf{A}^c_{t:t+\tau}), \mathbf{H}^{(i)}_t]$. This estimator attributes the heterogeneity of outcome generation to the observed historical information. However, in many scenarios, some factors that can also affect the outcomes may be not recorded, which we denote by $\mathbf{e}^{(i)}_t \in \mathcal{E} \subset \mathbb{R}^{d_e}$. This potentially results in extra hidden heterogeneity of outcome among different samples and time steps (Zou et al., 2023), which cause that the expected outcome conditional on histories is not equal to the true individual outcome. Formally, that is

$$\mathbb{E}[\mathbf{Y}^{(i)}_{t+\tau}(\mathbf{A}^c_{t:t+\tau})] \neq \mathbb{E}[\mathbf{y}_{t+\tau}|do(\mathbf{A}_{t:t+\tau} = \mathbf{A}^c_{t:t+\tau}), \mathbf{H}^{(i)}_t].$$

To bridge this gap, we propose to uncover the hidden factors related to outcomes and augment the forecast module with the learned factors. By capturing the hidden heterogeneity, we can forecast the counterfactual outcome more preciously for the individuals. In this paper, we assume the latent factors do not affect treatment assignment, and therefore the standard assumptions (Rosenbaum & Rubin, 1983) in causal inference hold. The identification condition of counterfactual outcome are satisfied. We mainly consider the 1-step forecast problem (i.e. $\tau = 1$). The problem of multi-step forecast is left to future work.

4 THLTS: The Proposed Method

In this section, we present the details of our proposed Time-shared Heterogeneity Learning from Time Series (THLTS) algorithm. It is a flexible component that can be combined with the different models and subsequently improve their performance.

4.1 Preliminary for Capturing Hidden Heterogeneity

The mainstream models predict counterfactual outcome based on the heterogeneity informed by the representation encoding the histories. Formally, the prediction can be expressed by $h(\phi(\mathbf{H}^{(i)}_t), \mathbf{a}^c_t)$, where $\phi(\cdot)$ is the representation function built upon sequence models, and h is the predictive model.

However, hidden heterogeneity of outcome generation beyond the histories can still exist among different samples and time steps, which is caused by hidden outcome-related factors $\mathbf{e}^{(i)}_t$. It has been verified in previous literature (Zou et al., 2023) that by augmenting the prediction model input with the extra hidden factors, that is $g(\phi(\mathbf{H}_t^{(i)}), \mathbf{e}_t^{(i)}, \mathbf{a}_t^c)$, we can approximate the true individual outcome more closer. There have been large amount of works in reinforcement learning investigating the problem of partially observed markov decision process (POMDP) and trying to recover latent states(Lee et al., 2020; Lei et al., 2022; Igl et al., 2018) across time steps. The supervision information for time-varying latent variable inference (i.e. the high-dimensional observation) is usually sufficient in POMDP. Conversely, the supervision information (i.e. outcome variable) is much limited in our problem, since it is only one-dimensional. Therefore, the models built for capturing the time-varying latent factors is excessively flexible and may result in sub-optimal performance. This will be presented thoroughly in the section of experiments. To mitigate this circumstance, we give an subtle method to learn the shared part of time-varying latent factors. This is designed for computational efficiency while trading off the flexibility of models.

4.2 Learning Time-shared Latent factors

Inspired by the analysis above, we attempt to build up a framework that infer the shared part $\bar{\mathbf{e}}$ of time-varying latent factors $\{\mathbf{e}_t\}_{t=1}^T$, and learn the forecast model g taking the histories, time-shared latent factors and treatments as input. Through theoretical analysis below, we determine to learn the mean value of factor factors across time steps as the time-shared latent factors.

Proposition 4.1. *Assuming the function of prediction model g is β-Lipschitz on $\mathbf{e}$, formally $|g(\phi(\mathbf{H}), \mathbf{e}, \mathbf{a}) - g(\phi(\mathbf{H}), \mathbf{e}', \mathbf{a})| \le \beta \cdot ||\mathbf{e} - \mathbf{e}'||_2$, then the total increased error across time induced by substitute time-varying $\mathbf{e}_t$ with constant $\bar{\mathbf{e}}$ can be characterized as following:*

$$\sum_{t=1}^{T}(g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 \le 2\sum_{t=1}^{T}(g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 + 2\beta^2 \cdot \sum_{t=1}^{T} ||\mathbf{e}_t - \bar{\mathbf{e}}||_2^2 \quad (1)$$

We observe that when the latent factors is substituted as the mean value across time steps, the upper bound in the r.h.s of Equation 1 reaches the minimal value. Formally, the substituted latent factors is

$$\bar{\mathbf{e}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{e}_t \quad (2)$$

We can observe from Equation 2 that when the latent factors are generated from a distribution with constant expectation, our pursuit becomes to infer the expectation for each sample. This is a first-order statistic of latent factors and can be learned with variational inference technology (Kingma & Welling, 2014; Rezende et al., 2014).

Proposition 4.2. *We assume the following conditions are satisfied:*

1. *The latent factors are generated by $\mathbf{e}_t = \bar{\mathbf{e}} + \eta_t$, where $\eta_t \sim p(\eta)$ is a noise term with zero mean.*
2. *The outcome distribution $p(\mathbf{y}_t|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}) = \int_\eta p(\mathbf{y}_t|\mathbf{H}_t, \mathbf{a}, \mathbf{e} = \bar{\mathbf{e}} + \eta) \cdot p(\eta)d\eta$ is in the function family of decoder $p_\rho(\mathbf{y}_t|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}})$.*
3. *The posterior distribution $p(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$ is in the function family of encoder $q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$.*

Then there is an optimal solution for maximizing the evidence lower bound (ELBO) of variational autoencoders, which characterizes the underlying data generation process:

$$q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t), \quad p_\rho(\mathbf{y}|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}) = p(\mathbf{y}|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}), \quad p_\rho(\bar{\mathbf{e}}|\mathbf{H}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t)$$

where the ELBO is defined as:

$$\sum_{i=1}^{n}\sum_{t=1}^{T^{(i)}}\left(\mathbb{E}_{\bar{\mathbf{e}}\sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)})}\left[\log p_\rho(\mathbf{y}_t^{(i)}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \bar{\mathbf{e}})\right] + D_{KL}(p_\rho(\bar{\mathbf{e}}|\mathbf{H}_t^{(i)})|q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)}))\right)$$

Therefore, the mean of latent factors can be learned with the architecture of variational autoencoders (VAEs) by assuming the learned latent factors keeps constant across time and maximizing the ELBO. The detailed background information of VAEs can be found in the section F of Appendix.

Although the learned time-shared latent factors can not exactly capture the time-varying process of latent factors, it still can facilitate more precious counterfactual outcome forecast with trade-off between model flexibility and computation efficiency. This will be empirically validated in the section of experimental results. The detailed proof of the propositions above can be found in Section C of Appendix.

4.3 Implementations

In this paper, we choose the architecture of variational autoencoders (VAEs) to infer time-shared latent factors $\bar{\mathbf{e}}^{(i)}$ for each sample. The instantiated model architecture consists of three parts, which are inference result memory, encoders and forecast models respectively. The overall framework is demonstrated in the Figure 2. We successively introduce the components in this subsection.

Inference Result Memory Since the stochastic nature of the data generation process, the time-shared latent factors can not be inferred with a deterministic manner. Hence, our inference result is expressed by a distributional estimation rather than point-wise estimation. Inspired by the common practice in variational inference, the estimated distribution is characterized by a gaussian distribution with the mean vector $\mu_t^{(i)}$ and variance vector $\sigma_t^{(i)}$. When there is no observation of treatment outcome at the initial time step, we set the initial distributional inference of latent factors as standard gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_e)$. Formally, the vector of mean and variance is set as $\mu_0^{(i)} = \mathbf{0}_e, \sigma_0^{(i)} = \mathbf{1}_e$. We keep record of the inferred distribution information for each sample. When new observations of treatment outcome arrive, we can obtain the new inference result (i.e. the mean vector μ_i and variance vector σ_i) and update the distribution record for the corresponding sample.

Latent Factor Encoders When the outcome $\mathbf{y}_{t+1}^{(i)}$ of new treatment $\mathbf{a}_{t+1}^{(i)}$ is observed, we can update the inference result of latent factors based on the new observation. According to Bayes' theorem (Davies, 1988), the posterior is determined by the likelihood and prior distribution. Therefore, the input of our encoders includes not only the information of observations (i.e. new outcome $\mathbf{y}_{t+1}^{(i)}$, new treatments $\mathbf{a}_{t+1}^{(i)}$ and representation of histories $\phi(\mathbf{H}_{t+1}^{(i)})$), but also the mean μ^{pr} and variance σ^{pr} of prior distribution $p_\rho(\bar{\mathbf{e}}|\mathbf{H}_{t+1})$. Specifically, the encoder $q_\varphi(\cdot)$ represents a gaussian distribution characterized by two deep neural networks $f_\varphi^\mu(\cdot)$ and $f_\varphi^\sigma(\cdot)$ as following:

$$\begin{aligned} q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_{t+1}^{(i)}, \mathbf{a}_{t+1}^{(i)}, \mathbf{y}_{t+1}^{(i)}) = \mathcal{N}(f_\varphi^\mu(\mathbf{y}_{t+1}^{(i)}, \mathbf{a}_{t+1}^{(i)}, \phi(\mathbf{H}_{t+1}^{(i)}), \mu^{pr}, \sigma^{pr}), \\ \mathrm{Diag}(f_\varphi^\sigma(\mathbf{y}_{t+1}^{(i)}, \mathbf{a}_{t+1}^{(i)}, \phi(\mathbf{H}_{t+1}^{(i)}), \mu^{pr}, \sigma^{pr})^2)), \end{aligned} \tag{3}$$

where φ is the parameters of deep neural networks and $\phi(\cdot)$ is implemented by sequence model. The specification of μ^{pr} and σ^{pr} will be demonstrated in the part of training process.

Forecast Model The forecast model $g_\rho(\cdot)$ output the counterfactual outcome based on the counterfactual treatments $\mathbf{a}^c$, the representation of histories $\phi(\mathbf{H}_{t+1}^{(i)})$ and inferred latent factors $\bar{\mathbf{e}}_i$. By exploiting the hidden heterogeneity encoded in $\bar{\mathbf{e}}_i$, we can forecast counterfactual outcome more accurately. The model $g_\rho(\cdot)$ can act as the key component of the decoder in VAEs and therefore be trained with the encoder together by the technology of variational inference.

Training Process We train the model components above by decomposing the histories into several time steps. According to the Bayes' Theorem, the obtained posterior distribution can be viewed as the prior distribution of the next time step. Therefore, the obtained $\mu_{t-1}^{(i)}$ and $\sigma_{t-1}^{(i)}$ can substitute the role of μ^{pr} and σ^{pr} for the t^{th} time step. Specifically, the objective function of the i^{th} sample for training at the t^{th} time step is:

$$\begin{aligned} \mathcal{L}_t^{(i)} = \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)})} \Big[\log p_\rho(\mathbf{y}_t^{(i)}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \bar{\mathbf{e}}) - \\ & D_{KL}\left(q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)}) \Big| \mathcal{N}\left(\mu_{t-1}^{(i)}, \mathrm{Diag}((\sigma_{t-1}^{(i)})^2) \right) \right) \Big]. \end{aligned}$$

The decoded outcome distribution $p_\rho(\mathbf{y}_t^{(i)}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \bar{\mathbf{e}})$ is determined by the forecast model, formally $p_\rho(\mathbf{y}_t^{(i)}|\mathbf{H}_t^{(i)}, \mathbf{a}_t^{(i)}, \bar{\mathbf{e}}) = \mathcal{N}(g_\rho(\mathbf{a}_t^{(i)}, \phi(\mathbf{H}_t^{(i)}), \bar{\mathbf{e}}), (\sigma_y)^2)$, where σ_y is set as hyper-parameter.

We model the posterior distribution of time-shared latent factors as the gaussian distribution $\mathcal{N}(\mu_t^{(i)}, \text{Diag}((\sigma_t^{(i)})^2))$ obtained from the encoder. To be specific, the vectors of mean and variance are learned as following:

$$\begin{aligned}\mu_t^{(i)} &= f_\varphi^\mu(\mathbf{y}_t^{(i)}, \mathbf{a}_t^{(i)}, \phi(\mathbf{H}_t^{(i)}), \mu_{t-1}^{(i)}, \sigma_{t-1}^{(i)}),\\ \sigma_t^{(i)} &= f_\varphi^\sigma(\mathbf{y}_t^{(i)}, \mathbf{a}_t^{(i)}, \phi(\mathbf{H}_t^{(i)}), \mu_{t-1}^{(i)}, \sigma_{t-1}^{(i)}).\end{aligned} \tag{4}$$

For the initial time step, the prior distribution is set as $\mu_0^{(i)} = \mathbf{0}_e, \sigma_0^{(i)} = \mathbf{1}_e$. Finally, inspired by the analysis in Proposition 4.2, we define the objective function for training the model as the sum of $\mathcal{L}_t^{(i)}$ among the samples and time steps:

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{t=1}^{T^{(i)}} \mathcal{L}_t^{(i)}.$$

Forecast Process When the model has been trained, we can forecast counterfactual outcome from the model $g_\rho(\cdot)$ based on simultaneously histories, treatments and the inferred latent factors. For the individual with observed histories $\mathbf{H}_{t+1}$, counterfactual treatment $\mathbf{a}_{t+1}^c$, the counterfactual outcome is estimated with the following two steps.

At the first step, the posterior distribution of time-shared latent factors $\bar{\mathbf{e}}$ is obtained by successively feeding the elements of histories $\{(\mathbf{y}_j, \mathbf{a}_j, \phi(\mathbf{H}_j))\}_{j=1}^t$ into the encoder $q_\varphi(\cdot)$ and updating the parameters of inferred distribution μ_j and σ_j until $j = t$ according to Equation 4. The resulting distribution of latent factors is $\mathcal{N}(\mu_t, \text{Diag}((\sigma_t)^2))$.

Secondly, we repeatedly sample latent factors for m times and empirically estimate the expectation $\mathbb{E}[g_\rho(\mathbf{a}_{t+1}^c, \phi(\mathbf{H}_{t+1}), \bar{\mathbf{e}})]$ as the forecast result. Formally, the estimated result is as following:

$$\begin{aligned}&\hat{\mathbf{y}}_{t+1}(\mathbf{a}_{t+1}^c) = \frac{1}{m}\sum_{j=1}^{m} g_\rho(\mathbf{a}_{t+1}^c, \phi(\mathbf{H}_{t+1}), \bar{\mathbf{e}}^j),\\ &\bar{\mathbf{e}}^j \sim \mathcal{N}(\mu_t, \text{Diag}((\sigma_t)^2)), \quad 1 \le j \le m\end{aligned} \tag{5}$$

The pseudo-code of whole algorithm can be found in Algorithm 1 of Appendix. It is noteworthy that our model and forecast process is agnostic to the architecture of sequence model. Therefore, our method can be viewed as a flexible model component. When plugged into any off-the-shell counterfactual forecast model, it can significantly enhance the forecast performance. This claim will be substantiated in the section of experiments.

5 Experiments

In this section, we first evaluate the effectiveness of THLTS by conducting various experiments using synthetic dataset. To further verify the effectiveness of THLTS in real scenarios, we leverage a real-world medical dataset MIMIC-III (Johnson et al., 2016; Wang et al., 2020) to construct a semi-synthetic dataset.

5.1 Experimental Setup

We validate the effectiveness of our THLTS method by combining it with the representative instances of the advanced counterfactual forecast models in time series, that are long-short term memory (LSTM) architecture with IPS-Reweighting (RMSN) (Lim et al., 2018), treatment invariant representation (CRN) (Bica et al., 2020b) and Transformer architecture with treatment invariant representation (Causal Transformer) (Melnychuk et al., 2022) respectively. We also compare the forecast performance with several other methods, that are G-net (Li et al., 2020) and MSM (Robins et al., 2000). Furthermore, to justify the rationality of learning shared part of latent factors, we introduce a new

Table 1: Experimental results for synthetic dataset under the setting of varying strength of latent factors, the average RMSE ± standard deviation is recorded for 10 repeated experiments.

Fix the sequence length d = 20, varying the latent factor strength σ_e					
σ_e	$\sigma_e = 0.5$	$\sigma_e = 1.0$	$\sigma_e = 1.5$	$\sigma_e = 2.0$	$\sigma_e = 2.5$
MSM	0.657 ± 0.000	0.831 ± 0.000	1.134 ± 0.000	1.375 ± 0.000	1.626 ± 0.000
G-Net	0.542 ± 0.030	0.601 ± 0.021	0.719 ± 0.026	0.851 ± 0.050	0.958 ± 0.041
RMSN	0.379 ± 0.013	0.525 ± 0.019	0.658 ± 0.019	0.797 ± 0.042	0.924 ± 0.032
RMSN-THLTS$^{(v)}$	0.338 ± 0.010	0.462 ± 0.009	0.573 ± 0.017	0.689 ± 0.017	0.799 ± 0.017
RMSN-THLTS	0.328 ± 0.007	0.441 ± 0.009	0.557 ± 0.011	0.673 ± 0.021	0.783 ± 0.020
CRN	0.302 ± 0.008	0.437 ± 0.009	0.551 ± 0.017	0.665 ± 0.020	0.777 ± 0.019
CRN-THLTS$^{(v)}$	0.278 ± 0.004	0.392 ± 0.007	0.495 ± 0.014	0.604 ± 0.016	0.722 ± 0.016
CRN-THLTS	**0.266 ± 0.005**	**0.364 ± 0.007**	**0.463 ± 0.006**	**0.567 ± 0.003**	**0.677 ± 0.004**
CT	0.366 ± 0.012	0.428 ± 0.005	0.521 ± 0.005	0.620 ± 0.006	0.840 ± 0.007
CT-THLTS$^{(v)}$	0.471 ± 0.011	0.609 ± 0.002	0.738 ± 0.014	0.903 ± 0.020	1.096 ± 0.039
CT-THLTS	0.296 ± 0.009	0.396 ± 0.010	0.489 ± 0.005	0.599 ± 0.009	0.712 ± 0.014

baseline method for comparison, which is denoted as THLTS$^{(v)}$. It leverages a linear layer $\psi(\cdot)$ to replace the priors of latent factors μ^{pr} and σ^{pr} by $\psi(\mu_{t-1}^{(i)})$ and $\psi(\sigma_{t-1}^{(i)})$ respectively to encourage the model to capture the temporal variability.

5.2 Synthetic Experiment

Dataset. We simulated various synthetic dataset under different setting to evaluate the effectiveness of THLTS. For each sample, we sample the covariates $\mathbf{x}_i$ with dimension d_x at the time step $t = 0$ independently from Gaussian Distribution:

$$\mathbf{x}_{t=0}^{(i)} \sim \mathcal{N}(0, \mathbf{I}_{d_x}). \tag{6}$$

We set the dimension of context d_x to be 10 across all synthetic settings. The covariates of each sample at the t^{th} time step depends on the previous covariates and treatment:

$$\mathbf{x}_t^{(i)} = \frac{1}{ws} \sum_{j=t-ws}^{t-1} \mathbf{x}_j^{(i)} + \mathbf{a}_{t-1}^{(i)} \cdot A + \mathcal{N}(0, 0.3^2 I), \tag{7}$$

where ws denotes the window size that determines the influence of previous covariates, and $A \in R^{p \times 1}$ is the constant vector sampled from $\mathcal{N}(0, \mathbf{I}_p)$. The treatment assignment is confounded by the current covariates:

$$a_i^t \sim Bernoulli(\beta^T x_t^{(i)}), \tag{8}$$

where $\beta \in \mathbb{R}^{p \times 1}$ is a parameter vector sampled from Gaussian Distribution $\mathcal{N}(0, \mathbf{I}_p)$. The outcome of each sample is generated as following:

$$\mathbf{y}_t^{(i)} = B \cdot [\mathbf{x}_{t-ws+1}^{(i)}, \mathbf{x}_{t-ws+2}^{(i)}, ..., \mathbf{x}_t^{(i)}] \cdot C + CE(t) \cdot \mathbf{e}_t^{(i)} + \mathcal{N}(0, 0.5^2), 1 \leq t \leq T^{(i)} \tag{9}$$

where $B \in \mathbb{R}^{ws \times 1}, C \in \mathbb{R}^{p \times 1}$ are the weight vectors with each element sampled from $\mathcal{N}(0, 1)$, $CE(t) = \frac{CE(t-1)}{2} + \mathbf{a}_t^{(i)}$ is the time-decaying treatment effect at the t^{th} time step. In the synthetic dataset, the time horizon of each sample is set to be constant $T^{(i)} = d, 1 \leq i \leq n$. The variable $\mathbf{e}_t^{(i)}$ is unrecorded in the histories $\mathbf{H}_t$ and encodes the hidden heterogeneity. To verify the necessity of dealing with hidden heterogeneity, we firstly consider the setting of static latent factors across time steps. The latent factors are generated with the following two steps:

$$\bar{\mathbf{e}}^{(i)} \sim \mathcal{N}(0, \sigma_e^2), \quad \mathbf{e}_t^{(i)} = \bar{\mathbf{e}}^{(i)}, \quad 1 \leq t \leq d \tag{10}$$

where σ^e is a constant that controls the strength of latent factors.

Results. For each experimental setting, we have conducted repeated experiments for 10 times, and we record the average RMSE and standard deviation for each method.

Table 2: Synthetic experimental results under different trajectory horizon d, the average RMSE ± standard deviation is recorded for 10 repeated experiments.

Fix the hidden factor strength σ_e = 1.5, varying the sequence length d					
d	$d = 10$	$d = 15$	$d = 20$	$d = 25$	$d = 30$
MSM	1.178 ± 0.000	1.123 ± 0.000	1.134 ± 0.000	1.086 ± 0.000	1.120 ± 0.000
G-Net	0.758 ± 0.009	0.696 ± 0.017	0.719 ± 0.026	0.731 ± 0.034	0.796 ± 0.022
RMSN	0.729 ± 0.012	0.650 ± 0.028	0.658 ± 0.019	0.660 ± 0.043	0.686 ± 0.037
RMSN-THLTS$^{(v)}$	0.695 ± 0.008	0.585 ± 0.009	0.573 ± 0.017	0.543 ± 0.012	0.561 ± 0.016
RMSN-THLTS	0.694 ± 0.007	0.578 ± 0.007	0.557 ± 0.011	0.522 ± 0.018	0.538 ± 0.017
CRN	0.621 ± 0.005	0.529 ± 0.009	0.551 ± 0.017	0.528 ± 0.017	0.586 ± 0.019
CRN-THLTS$^{(v)}$	0.636 ± 0.011	0.515 ± 0.007	0.495 ± 0.014	0.457 ± 0.015	0.471 ± 0.012
CRN-THLTS	**0.617 ± 0.005**	**0.500 ± 0.010**	**0.463 ± 0.006**	**0.423 ± 0.005**	**0.430 ± 0.010**
CT	0.668 ± 0.005	0.548 ± 0.005	0.521 ± 0.005	0.480 ± 0.005	0.480 ± 0.005
CT-THLTS$^{(v)}$	0.885 ± 0.005	0.758 ± 0.040	0.738 ± 0.014	0.699 ± 0.028	0.707 ± 0.029
CT-THLTS	0.642 ± 0.008	0.520 ± 0.007	0.489 ± 0.005	0.453 ± 0.007	0.459 ± 0.007

As demonstrated in Table 1, MSM underperforms the other methods because it struggle with the complex variable relationship in the data generation. The models harnessing the power of deep learning (i.e. G-Net, RMSN, CRN, CT) achieve more precious forecast. When our proposed THLTS is applied in conjunction with the counterfactual outcome forecast models, their performance has been significantly enhanced. We find that larger strength of latent factors makes the forecast performance worse for all the models because the hidden heterogeneity problem is more severe. Under these scenarios, our method can leads to more remarkable performance enhancement, which shows the rationality of our method. Though THLTS$^{(v)}$ also offer advantages to these models, the performance improvement provided by THLTS$^{(v)}$ is inferior to that of THLTS.

We also conduct the experiments with different trajectory horizon d. Table 2 illustrates the experimental results. The advantages of THLTS over baseline methods become progressively larger as the trajectory horizon d increases. This is because longer histories can facilitate more precise recovery of latent factors in our method.

We conduct experiments under the setting of time-varying latent factors $\mathbf{e}_t^{(i)}$ to justify our strategy of learning time-shared latent factors. We established the underlying latent factor to evolve around a given centroid. Formally, $\mathbf{e}_t^{(i)} = \bar{\mathbf{e}}^{(i)} + \mathcal{N}(0, \sigma_{vary})$, where σ_{vary} controls the variation degree of latent factors across time steps. The results are presented in Figure 3, suggest that the performance enhancement provided by THLTS is overall more pronounced than that of THLTF$^{(v)}$. This shows the trade-off between model flexibility and forecast performance. When σ_{vary} becomes larger, the time-varying part of latent factors surpasses the time-shared part, the margin becomes weaker. We also conduct experiments of time-varying latent factors without given centroid. The results also validate the effectiveness of our method. Additional experimental results can be found in Section A of Appendix.

5.3 Semi-synthetic Experiment

Dataset. We used a semi-synthetic pipeline constructed by Melnychuk et al. (2022) using MIMIC-III (Johnson et al., 2016) to further verify the effectiveness of THLTS. Specifically, we extracted 1,000 patient trajectories with 25 vital covariates (including heart rate, sodium and red blood cell count) and 3 static covariates (gender, age and ethnicity) similarly to (Melnychuk et al., 2022). Partial outcome is generated by time-varying latent factors through an Gaussian process function, and the strength is controlled by a constant parameter α_g, the details of semi-synthetic experiment is included in Section B of Appendix.

Results. To note that when the strength α_g becomes larger, the influence of latent factors on outcome generation across time steps also increases. As results shown in Table 3, after integration with our THLTS component, the counterfactual outcome forecast models achieve better forecast performance

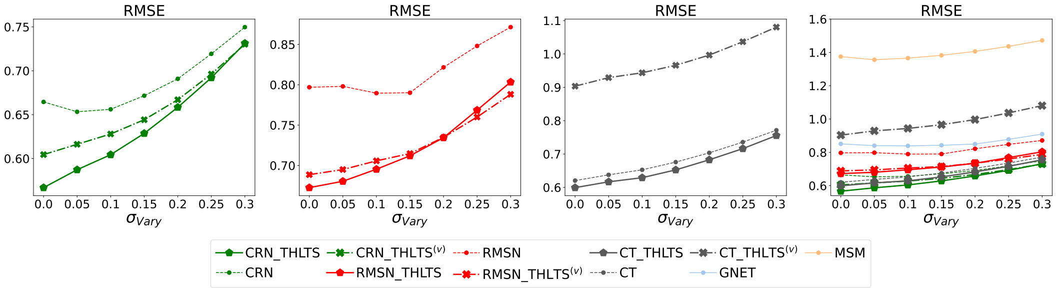

Figure 3: Performance of methods under the setting that the latent factors evolve around a given centroid across time steps. The x-axis represents different variation degree of latent factors (i.e. σ_{vary}). The rightest sub-figure shows the performance of all the methods. The leftest three sub-figures respectively shows the effect of THLTS on CRN, RMSN and CT.

Table 3: Results for semi-synthetic dataset, the average RMSE ± standard deviation is recorded for 10 repeated experiments.

Vary the heterogeneity strength α_g					
α_g	$\alpha_g = 0.5$	$\alpha_g = 1.0$	$\alpha_g = 1.5$	$\alpha_g = 2.0$	$\alpha_g = 2.5$
MSM	1.127 ± 0.000	1.027 ± 0.000	1.118 ± 0.000	1.033 ± 0.000	0.762 ± 0.000
G-Net	0.410 ± 0.050	0.392 ± 0.048	0.397 ± 0.046	0.391 ± 0.035	0.396 ± 0.037
RMSN	0.256 ± 0.006	0.270 ± 0.004	0.268 ± 0.006	0.266 ± 0.005	0.271 ± 0.004
RMSN-THLTS$^{(v)}$	0.252 ± 0.008	0.267 ± 0.005	0.265 ± 0.006	0.263 ± 0.004	0.269 ± 0.004
RMSN-THLTS	0.246 ± 0.008	0.252 ± 0.004	0.253 ± 0.005	0.255 ± 0.005	0.261 ± 0.004
CRN	0.254 ± 0.018	0.268 ± 0.018	0.287 ± 0.015	0.294 ± 0.013	0.300 ± 0.013
CRN-THLTS$^{(v)}$	0.256 ± 0.024	0.265 ± 0.020	0.284 ± 0.020	0.291 ± 0.020	0.296 ± 0.021
CRN-THLTS	**0.221** ± **0.004**	0.267 ± 0.005	0.264 ± 0.006	0.276 ± 0.006	0.282 ± 0.006
CT	0.263 ± 0.008	0.267 ± 0.011	0.268 ± 0.009	0.264 ± 0.007	0.263 ± 0.007
CT-THLTS$^{(v)}$	0.259 ± 0.010	0.261 ± 0.007	0.261 ± 0.005	0.259 ± 0.005	0.257 ± 0.005
CT-THLTS	0.239 ± 0.006	**0.240** ± **0.006**	**0.246** ± **0.005**	**0.247** ± **0.003**	**0.247** ± **0.003**

than the original model. Compared to the version of capturing the time varying dynamics of latent factors THLTS$^{(v)}$, our proposed THLTS aims to capture the time-shared part, which acts as a regularizer constraining the learning of latent factors, and further improve the forecast performance. In summary, the semi-synthetic experiments on the MIMIC-III datasets confirm the effectiveness of our method.

6 Conclusion

In this paper, we studied the hidden heterogeneity problem of counterfactual outcome forecast in longitudinal setting. Neglecting the latent factors unobserved in the histories can degrade the prediction performance of counterfactual outcome forecast model. To tackle this problem, we propose Time-shared Heterogeneity Learning from Time Series (THLTS) method to uncover the time-shared part of latent factors and augment the forecast model with the learned latent representation. Owing to the flexibility of our method, it can be combined with arbitrary model backbones. Extensive experimental results indicate the effectiveness of our method on improving the performance of mainstream counterfactual outcome forecast models.

References

Serge Assaad, Shuxi Zeng, Chenyang Tao, Shounak Datta, Nikhil Mehta, Ricardo Henao, Fan Li, and Lawrence Carin. Counterfactual representation learning with balancing weights. In *International*

Conference on Artificial Intelligence and Statistics, pp. 1972–1980. PMLR, 2021.

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

Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. *arXiv preprint arXiv:2002.04083*, 2020b.

Ioana Bica, James Jordon, and Mihaela van der Schaar. Estimating the effects of continuous-valued interventions using generative adversarial networks. *Advances in neural information processing systems (NeurIPS)*, 2020c.

CM Booth and IF Tannock. Randomised controlled trials and population-based observational research: partners in the evolution of medical evidence. *British journal of cancer*, 110(3):551–555, 2014.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14 (11), 2013.

Mouad El Bouchattaoui, Myriam Tami, Benoit Lepetit, and Paul-Henry Cournède. Causal dynamic variational autoencoder for counterfactual regression in longitudinal data. *arXiv preprint arXiv:2310.10559*, 2023.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

P Davies. Kendall's advanced theory of statistics. volume 1. distribution theory, 1988.

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

Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Deep iv: A flexible approach for counterfactual prediction. In *International Conference on Machine Learning*, pp. 1414–1423. PMLR, 2017.

Negar Hassanpour and Russell Greiner. Counterfactual regression with importance sampling weights. In *IJCAI*, pp. 5880–5887, 2019.

Negar Hassanpour and Russell Greiner. Learning disentangled representations for counterfactual regression. In *International Conference on Learning Representations*, 2020.

Tobias Hatt and Stefan Feuerriegel. Sequential deconfounding for causal inference with unobserved confounders. *arXiv preprint arXiv:2104.09323*, 2021.

James Heckman. Instrumental variables: A study of implicit behavioral assumptions used in making program evaluations. *Journal of human resources*, pp. 441–462, 1997.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Xuelin Huang and Jing Ning. Analysis of multi-stage treatments for recurrent diseases. *Statistics in medicine*, 31(24):2805–2821, 2012.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for POMDPs. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2117–2126. PMLR, 10–15 Jul 2018.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 3020–3029, New York, New York, USA, 2016. PMLR.

Fredrik D Johansson, Nathan Kallus, Uri Shalit, and David Sontag. Learning weighted representations for generalization across designs. *arXiv preprint arXiv:1802.08598*, 2018.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Ron Kohavi and Roger Longbotham. Unexpected results in online controlled experiments. *Acm Sigkdd Explorations Newsletter*, 12(2):31–35, 2011.

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

Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Variational causal dynamics: Discovering modular world models from interventions. *CoRR*, abs/2206.11131, 2022.

Rui Li, Zach Shahn, Jun Li, Mingyu Lu, Prithwish Chakraborty, Daby Sow, Mohamed Ghalwash, and Li-wei H Lehman. G-net: a deep learning approach to g-computation for counterfactual outcome prediction under dynamic treatment regimes. *arXiv preprint arXiv:2003.10551*, 2020.

Bryan Lim, Alaa Ahmed, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. *Advances in neural information processing systems*, 31, 2018.

Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard S. Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, 2017.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15293–15329. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/melnychuk22a.html.

Zhaozhi Qian, Alicia Curth, and Mihaela van der Schaar. Estimating multi-cause treatment effects via single-cause perturbation. *Advances in Neural Information Processing Systems*, 34, 2021.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

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

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

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

Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. Learning Counterfactual Representations for Estimating Individual Dose-Response Curves. In *AAAI Conference on Artificial Intelligence*, 2020.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3076–3085, 2017.

Akira Tanimoto, Tomoya Sakai, Takashi Takenouchi, and Hisashi Kashima. Regret minimization for causal inference on large treatment space. In *International Conference on Artificial Intelligence and Statistics*, pp. 946–954. PMLR, 2021.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

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

Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pp. 222–235, 2020.

Yixin Wang and David M Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019.

Anpeng Wu, Kun Kuang, Bo Li, and Fei Wu. Instrumental variable regression with confounder balancing. In *International Conference on Machine Learning*, pp. 24056–24075. PMLR, 2022.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a50abba8132a77191791390c3eb19fe7-Paper.pdf.

Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

Shuxi Zeng, Serge Assaad, Chenyang Tao, Shounak Datta, Lawrence Carin, and Fan Li. Double robust representation learning for counterfactual prediction. *arXiv preprint arXiv:2010.07866*, 2020.

Hao Zou, Peng Cui, Bo Li, Zheyan Shen, Jianxin Ma, Hongxia Yang, and Yue He. Counterfactual prediction for bundle treatment. *Advances in Neural Information Processing Systems*, 33, 2020.

Hao Zou, Haotian Wang, Renzhe Xu, Bo Li, Jian Pei, Ye Jun Jian, and Peng Cui. Factual observation based heterogeneity learning for counterfactual prediction. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing (eds.), *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pp. 350–370. PMLR, 11–14 Apr 2023. URL https://proceedings.mlr.press/v213/zou23a.html.

A ADDITIONAL EXPERIMENTS

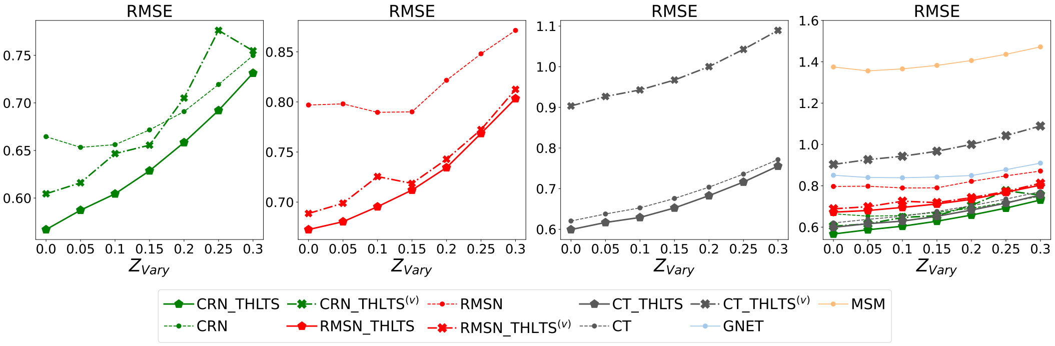

Figure 4: Performance of methods under the setting that the latent factors evolve without a given centroid. The x-axis represents different variation degree of latent factors (i.e. σ_{vary}). The rightest sub-figure shows the performance of all the methods. The leftest three sub-figures respectively shows the effect of THLTS on CRN, RMSN and CT.

The performance of THLTS when latent factor evolves without a given centroid: We consider a different scenario that latent factors evolves based on the value of the previous time step, formally $\mathbf{e}_t^{(i)} = \mathbf{e}_{t-1}^{(i)} + \mathcal{N}(0, \sigma_{vary})$, where σ_{vary} is a constant that controls the variation degree of latent factor across time steps. We set $d = 20$ and $m = 1.5$. The results are shown in Figure 4, the trends are similar to the results in main paper.

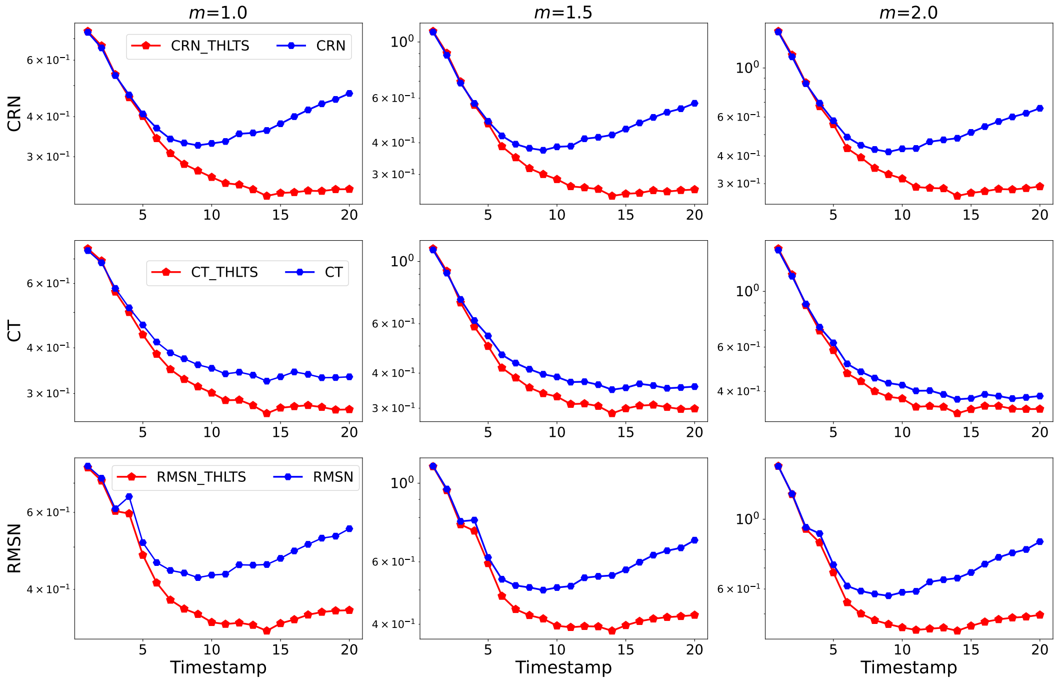

Figure 5: Counterfactual outcome forecast performance at each timestamp t.

The performance of THLTS for each timestamp: We further illustrate the performance of THLTS to estimate the counterfactual outcome at each time step t. Figure 5 depicts the results, the advantages of THLTS becomes progressively more prominent when the index of time step $t \geq 5$, by the virtue of more precise recovery of latent factors with longer histories data.

B Experimental Details

B.1 Semi-synthetic Experiments

We used the semi-synthetic dataset built from MIMIC-IIIJohnson et al. (2016) followed the pipeline introduced in Melnychuk et al. (2022); Schulam & Saria (2017). Specifically, we extracted 1,000 patient trajectories with time horizon $T^{(i)} = 20$. The covariates is composed of 25 time-varying patient signs (including heart rate, sodium and red blood cell count) and 3 static patients' information (gender, age and ethnicity).

Based on the extracted patient trajectories, there are three binary treatment and two continuous outcome simulated at each time step. The treatment assignment probability $p_{\mathbf{A}_t^l}$ is confounded by the patient covariates and previous outcome as follows:

$$p_{\mathbf{A}_t^l} = \sigma(\gamma_{\mathbf{A}}^l \bar{\mathbf{A}}_{T_l}(\bar{\mathbf{Y}}_{t-1} + \gamma_{\mathbf{X}}^l f_Y^l(\mathbf{X}_t) + b_l)), \tag{11}$$

where γ_X^l, γ_A^l are constant parameters that control the confounding strength of treatment $\mathbf{A}_l$, $\sigma(\cdot)$ is a sigmoid function, b_l is the constant bias for each treatment, then each binary treatment A_t is sample from Bernoulli Distribution with parameter $\mathbf{A}_t^l$. The untreated outcome is then generated by the combination of endogenous and exogenous parts:

$$\mathbf{Y}_{untreated,t}^{(i),j} = \underbrace{\alpha_S^j \text{B-Spline}(t) + \alpha_g^j g^{j,(i)}(t)}_{\text{endogenous}} + \underbrace{\alpha_f^j f_Z^j(\mathbf{X}_\mathbf{t}^{(\mathbf{i})})}_{\text{exogenous}} + \epsilon_t, \tag{12}$$

where $\alpha_S^j, \alpha_g^j, \alpha_f^j$ are constant parameter and ϵ_t is sampled from $\mathcal{N}(0, 0.005^2)$. Here we focus on $g^{j,(i)}(t)$, which is generated by independently for each patient from Gaussian process with Matérn Kernel, which is equivalent to the time-varying latent factors discussed in this paper. α_g^j is a constant parameter controls the contribution of it.

We then generate treated outcome as following:

$$\mathbf{Y}_{treated,t}^{(i),j} = \mathbf{Y}_{untreated,t}^{(i),j} + \sum_{i=t-ws}^{t} \frac{min_{l=0,1,2} \mathbb{1}[\mathbf{A}_i^l = 1] p_{\mathbf{A}_i^l} \beta_{l,j}}{(w^l - i)^{0.5}} \cdot \alpha_g^j g^{j,(i)}, \tag{13}$$

where the individual latent factors also impact the patients' response of a given treatment.

B.2 Computation Resource

We conduct our experiments on a Linux server, where the operation system is 18.04.1-Ubuntu. There are 8 NVIDIA GeForce RTX 3090 GPUS on this server. However, we only use one GPU. The internal memory is 504GB. Each run of our experiments cost around 10 minutes.

C Proof

Proposition C.1. ***(Restated)*** *Assuming the function of prediction model g is β-Lipschitz on $\mathbf{e}$, formally $|g(\phi(\mathbf{H}), \mathbf{e}, \mathbf{a}) - g(\phi(\mathbf{H}), \mathbf{e}', \mathbf{a})| \leq \beta \cdot ||\mathbf{e} - \mathbf{e}'||_2$, then the total increased error across time induced by substitute time-varying $\mathbf{e}_t$ with constant $\bar{\mathbf{e}}$ can be characterized as following:*

$$\sum_{t=1}^{T^{(i)}} (g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 \leq 2 \sum_{t=1}^{T^{(i)}} (g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 + 2\beta^2 \cdot \sum_{t=1}^{T^{(i)}} ||\mathbf{e}_t - \bar{\mathbf{e}}||_2^2 \tag{14}$$

Proof. For arbitrary time step $t \in \{1, 2, 3, ..., T^{(i)}\}$, we have

$$\begin{aligned}
(g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 &= (g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c) + g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 \\
&\leq 2 \cdot (g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c))^2 \\
&+ 2 \cdot (g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 \\
&= 2 \cdot (g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c))^2 + 2 \cdot (\beta||\mathbf{e}_t - \bar{\mathbf{e}}||_2)^2
\end{aligned}$$

By taking the sum of inequality for $t \in \{1, 2, 3, ..., T^{(i)}\}$, we obtain

$$\sum_{t=1}^{T^{(i)}} (g(\phi(\mathbf{H}_t), \bar{\mathbf{e}}, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 \leq 2 \sum_{t=1}^{T^{(i)}} (g(\phi(\mathbf{H}_t), \mathbf{e}_t, \mathbf{a}_t^c) - \mathbf{Y}_t(\mathbf{a}_t^c))^2 + 2\beta^2 \cdot \sum_{t=1}^{T^{(i)}} ||\mathbf{e}_t - \bar{\mathbf{e}}||_2^2$$

□

Proposition C.2. ***(Restated)*** *We assume the following conditions are satisfied:*

1. *The latent factors are generated by $\mathbf{e}_t = \bar{\mathbf{e}} + \eta_t$, where $\eta_t \sim p(\eta)$ is a noise term with zero mean.*
2. *The outcome distribution $p(\mathbf{y}_t|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}) = \int_\eta p(\mathbf{y}_t|\mathbf{H}_t, \mathbf{a}, \mathbf{e} = \bar{\mathbf{e}} + \eta) \cdot p(\eta) d\eta$ is in the function family of decoder $p_\rho(\mathbf{y}_t|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}})$.*
3. *The posterior distribution $p(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$ is in the function family of encoder $q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$.*

Then there is an optimal solution for maximizing the evidence lower bound (ELBO) of variational autoencoders, which characterizes the underlying data generation process:

$$q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t), \quad p_\rho(\mathbf{y}|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}) = p(\mathbf{y}|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}), \quad p_\rho(\bar{\mathbf{e}}|\mathbf{H}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t),$$

where the ELBO is defined as:

$$\sum_{i=1}^{n} \sum_{j=1}^{T^{(i)}} \left(\mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_j^{(i)}, \mathbf{a}_j^{(i)}, \mathbf{y}_j^{(i)})} \left[\log p_\rho(\mathbf{y}_j^{(i)}|\mathbf{H}_j^{(i)}, \mathbf{a}_j^{(i)}, \bar{\mathbf{e}}) \right] + D_{KL}(p_\rho(\bar{\mathbf{e}}|\mathbf{H}_j^{(i)})|q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_j^{(i)}, \mathbf{a}_j^{(i)}, \mathbf{y}_j^{(i)})) \right)$$

Proof. We can decompose the joint distribution of outcome sequence as

$$\log p_\rho(\mathbf{Y}_{1:t}|\mathbf{X}_{1:t}, \mathbf{A}_{1:t}) = \sum_{i=1}^{t} \log p_\rho(\mathbf{y}_i|\mathbf{X}_{1:i}, \mathbf{A}_{1:i}, \mathbf{Y}_{1:i-1}) = \sum_{i=1}^{t} \log p_\rho(\mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i)$$

Specifically,

$$\begin{align}
& \log p_\rho(\mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i) \nonumber \\
= \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} [\log p_\rho(\mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_{1:i})] \nonumber \\
= \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \left[\log \frac{p_\rho(\bar{\mathbf{e}}, \mathbf{Y}_{1:t}|\mathbf{X}_{1:t}, \mathbf{A}_{1:t})}{p_\rho(\bar{\mathbf{e}}|\mathbf{X}_{1:t}, \mathbf{A}_{1:t}, \mathbf{Y}_{1:t})} \right] \tag{15} \\
= \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \left[\log \frac{p_\rho(\bar{\mathbf{e}}, \mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i)}{q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \cdot \frac{q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)}{p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \right] \tag{16} \\
= \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \left[\log \frac{p_\rho(\bar{\mathbf{e}}, \mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i)}{q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \right] \tag{17} \\
+ \quad & D_{KL}(q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)|p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)) \tag{18} \\
= \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} \left[\log \frac{p_\rho(\mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i, \bar{\mathbf{e}})}{q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} + \log p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i) \right] \tag{19} \\
+ \quad & D_{KL}(q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)|p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)). \tag{20}
\end{align}$$

Since the time-shared latent factors $\bar{\mathbf{e}}$ is independently of $\mathbf{a}$ given the histories $\mathbf{H}$, we have $p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i) = p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i)$. Then we have

$$\begin{align}
& \log p_\rho(\mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i) \nonumber \\
= \quad & \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)} [\log p_\rho(\mathbf{y}_i|\mathbf{H}_i, \mathbf{a}_i, \bar{\mathbf{e}})] \nonumber \\
+ \quad & D_{KL}(p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i)|q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)) \tag{21} \\
+ \quad & D_{KL}(q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)|p_\rho(\bar{\mathbf{e}}|\mathbf{H}_i, \mathbf{a}_i, \mathbf{y}_i)) \nonumber
\end{align}$$

We take the sum of equations above for all the training samples and time steps, and obtain

$$\begin{aligned}
& \sum_{i=1}^{n} \log p_\rho(\mathbf{Y}^{(i)}_{1:T^{(i)}} | \mathbf{X}^{(i)}_{1:T^{(i)}}, \mathbf{A}^{(i)}_{1:T^{(i)}}) \\
- & \sum_{i=1}^{n} \sum_{j=1}^{T^{(i)}} D_{KL}(q_\varphi(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \mathbf{y}^{(i)}_j) | p_\rho(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \mathbf{y}^{(i)}_j)) \\
= & \sum_{i=1}^{n} \sum_{j=1}^{T^{(i)}} \mathbb{E}_{\bar{\mathbf{e}} \sim q_\varphi(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \mathbf{y}^{(i)}_j)} \left[\log p_\rho(\mathbf{y}^{(i)}_j | \mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \bar{\mathbf{e}}) \right] \\
+ & \sum_{i=1}^{n} \sum_{j=1}^{T^{(i)}} D_{KL}(p_\rho(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j)) | q_\varphi(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \mathbf{y}^{(i)}_j)) \qquad (22)
\end{aligned}$$

From the equations above, we can observe that when the following conditions are satisfied,

$$q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t), \quad p_\rho(\mathbf{y}|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}) = p(\mathbf{y}|\mathbf{H}_t, \mathbf{a}, \bar{\mathbf{e}}), \quad p_\rho(\bar{\mathbf{e}}|\mathbf{H}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t).$$

We have

$$p_\rho(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t) = p(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t) = q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{T^{(i)}} D_{KL}(q_\varphi(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \mathbf{y}^{(i)}_j) | p_\rho(\bar{\mathbf{e}}|\mathbf{H}^{(i)}_j, \mathbf{a}^{(i)}_j, \mathbf{y}^{(i)}_j)) = 0$$

Therefore, the r.h.s of Equation 22, which is the ELBO of our variational autoencoders, becomes $\sum_{i=1}^{n} \log p(\mathbf{Y}^{(i)}_{1:T^{(i)}} | \mathbf{X}^{(i)}_{1:T^{(i)}}, \mathbf{A}^{(i)}_{1:T^{(i)}})$ and reaches the maximal value. □

D LIMITATIONS

In this paper, we only theoretically analyze the setting where the latent factors are generated by a specific process $\mathbf{e}_t = \bar{\mathbf{e}} + \eta_t$, though the data generation process violating this assumption is also examined in the experiments. We leave the analysis of more complex scenarios to future works.

Due to the intrinsic challenges in evaluating counterfactual prediction, we only conduct experiments on the synthetic datasets and semi-synthetic datasets. Evaluation with expensive real-world experiments should also be considered.

E SYMBOL SUMMARY

We summarize the symbols and corresponding definitions in Table 4.

F BACKGROUND INFORMATION OF VAE

Generally, variational autoencoder is a framework that assume the observation $\mathbf{o}$ is generated from latent factors $\mathbf{z}$ and conditions $\mathbf{c}$ (if exists) and learn a variational approximation $q_\varphi(\mathbf{z}|\mathbf{o}, \mathbf{c})$ to substitute the true posterior distribution $p(\mathbf{z}|\mathbf{o}, \mathbf{c})$. The encoder outputting this approximated posterior is trained with decoder $p_\rho(\mathbf{o}|\mathbf{z}, \mathbf{c})$ and conditional prior component $p_\varphi(\mathbf{z}|\mathbf{c})$ to maximize a lower bound of the log-likelihood of the observed data $p(\mathbf{o}|\mathbf{c})$. Formally, the evidence lower bound (ELBO) is defined as:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\log p(\mathbf{o}|\mathbf{c})] \geq \mathcal{L}(\varphi, \rho) &= \mathbb{E}_{\mathcal{D}}[\mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{o},\mathbf{c})}[\log p_\rho(\mathbf{o}|\mathbf{z}, \mathbf{c}) + \log p_\varphi(\mathbf{z}|\mathbf{c}) - \log q_\varphi(\mathbf{z}|\mathbf{o}, \mathbf{c})]] \\
&= \mathbb{E}_{\mathcal{D}}[\mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{o},\mathbf{c})}[\log p_\rho(\mathbf{o}|\mathbf{z}, \mathbf{c})] - D_{KL}(\log q_\varphi(\mathbf{z}|\mathbf{o}, \mathbf{c})| \log p_\varphi(\mathbf{z}|\mathbf{c}))]
\end{aligned}$$

Usually, the outputted distributions of these components are defined as Gaussian distribution with parameterized expectation and variance. After training the models, we can use reparameterization

Table 4: The symbols used in the paper and corresponding definitions

Symbol	Definition
$\mathbf{x}_t^{(i)} \in \mathcal{X} \subset \mathbb{R}^{d_x}$	Covariate of the i^{th} sample at the t^{th} time step
$\mathbf{a}_t^{(i)} \in \mathcal{A}$	Treatment of the i^{th} sample at the t^{th} time step
$\mathbf{y}_t^{(i)} \in \mathbb{R}$	Outcome of the i^{th} sample at the t^{th} time step
$T_t^{(i)} \in \mathbb{N}$	The observed history length of the i^{th} sample
$\mathbf{X}_{t:t+\tau}^{(i)}$	The observed covariate history of the i^{th} sample
$\mathbf{A}_{t:t+\tau}^{(i)}$	The observed treatment history of the i^{th} sample
$\mathbf{Y}_{t:t+\tau}^{(i)}$	The observed outcome history of the i^{th} sample
$\mathbf{H}_t^{(i)}$	The observed history of the i^{th} sample
$\phi(\cdot)$	The sequence model learning the representation of history
$\mathbf{e}_t^{(i)}$	The latent factors of the i^{th} sample at the t^{th} time step
$\bar{\mathbf{e}}^{(i)}$	The time-shared latent factors of the i^{th} sample
$q_\varphi(\cdot)$	The encoder outputting the posterior of latent factors
$p_\rho(\cdot)$	The decoder outputting the distribution of outcomes
$g_\rho(\cdot)$	The predictive model (also a component in the decoder)
$m \in \mathbb{N}$	The repeated sampling times for estimating counterfactual outcome

trick to sample latent factors from $q_\varphi(\mathbf{z}|\mathbf{o}, \mathbf{c})$. Given the sampled latent factors, we can change the conditions to the target value and decode the desired new observations. Specifically, in this paper, the observed histories $\mathbf{H}_t$ and current treatment $\mathbf{a}_t$ play the role of conditions $\mathbf{c}$, observed outcome $\mathbf{y}_t$ act as the observation $\mathbf{o}$, and the time-shared latent factors substitute the latent factors $\mathbf{z}$. Therefore, the encoder outputs the variational approximation of posterior $q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$. We can sample time-shared latent factors given the histories, observed treatments and outcomes from $q_\varphi(\bar{\mathbf{e}}|\mathbf{H}_t, \mathbf{a}_t, \mathbf{y}_t)$, change the treatment (i.e. part of conditions) to be the counterfactual $\mathbf{a}_t^c$ (i.e. target value of conditions), and finally obtained the counterfactual outcome from the decoder $p_\rho(\mathbf{y}|\mathbf{H}_t, \mathbf{a}^c, \bar{\mathbf{e}})$.

G Pseudo-Code of Our Method

The working process of our algorithm can be found in Algorithm 1.

Algorithm 1 Time-shared Heterogeneity Learning from Time Series (THLTS)

```
Input: Observational data $\{\{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_t^{(i)}\}_{t=1}^{T^{(i)}}\}_{i=1}^{n}$, the histories of evaluated sample
    $\{\mathbf{x}_t, \mathbf{a}_t, \mathbf{y}_t\}_{t=1}^{T}$ and the counterfactual treatment $\mathbf{a}_{T+1}^{c}$ at the $(T+1)^{th}$ time step.
Output: Counterfactual outcome forecast $\hat{\mathbf{y}}$.
Train the backbone model for learning representation of histories $\phi$ and latent factor model,
    including encoder $q_{\varphi}(\cdot)$ and forecast model $g_{\rho}(\cdot)$.
Set $\mu_0 \leftarrow \mathbf{0}_e, \sigma_0 \leftarrow \mathbf{1}_e$.
for $k = 1, 2, ..., T$ do
    Update $\mu_k \leftarrow f_{\varphi}^{\mu}(\mathbf{y}_{k-1}, \mathbf{a}_{k-1}, \phi(\mathbf{H}_{k-1}), \mu^{k-1}, \sigma^{k-1})$
    Update $\sigma_k \leftarrow f_{\varphi}^{\sigma}(\mathbf{y}_{k-1}, \mathbf{a}_{k-1}, \phi(\mathbf{H}_{k-1}), \mu^{k-1}, \sigma^{k-1})$.
end for
Set $\hat{y} \leftarrow 0$. // Under out-of-sample setting.
for $k = 1, 2, ..., m$ do
    Sample $r \sim \mathcal{N}(0, \mathbf{I}_e)$.
    Compute $\bar{\mathbf{e}} \leftarrow \mu_T + r \odot \sigma_T$
    Update $\hat{y} \leftarrow \hat{y} + \frac{1}{m} \cdot g^{\rho}(\mathbf{a}_{T+1}^{c}, \phi(\mathbf{H}_{T+1}), \bar{\mathbf{e}})$.
end for
return Forecasted outcome $\hat{y}$.
```