Self-Interpretable Time Series Prediction with Counterfactual Explanations

Jingquan Yan [1] **Hao Wang** [1]

Abstract

Interpretable time series prediction is crucial for safety-critical areas such as healthcare and autonomous driving. Most existing methods focus on interpreting predictions by assigning important scores to segments of time series. In this paper, we take a different and more challenging route and aim at developing a self-interpretable model, dubbed Counterfactual Time Series (CounTS), which generates counterfactual and actionable explanations for time series predictions. Specifically, we formalize the problem of time series counterfactual explanations, establish associated evaluation protocols, and propose a variational Bayesian deep learning model equipped with counterfactual inference capability of time series abduction, action, and prediction. Compared with state-of-the-art baselines, our self-interpretable model can generate better counterfactual explanations while maintaining comparable prediction accuracy. Code will be available at https://github.com/Wang-ML-Lab/self-interpretable-time-series.

1. Introduction

Deep learning (DL) has become increasingly prevalent, and there is naturally a growing need for understanding DL predictions in many decision-making area, such as healthcare diagnosis and public policy-making. The high-stake nature of these areas means that these DL predictions are considered trustworthy only when they can be well explained. Meanwhile, time-series data has been frequently used in these areas but it is always challenging to explain a time-series prediction due to the nature of temporal dependency and varying patterns over time. Moreover, time-series data often comes with confounding variables that affect both the input and output, making it even harder to explain predictions from DL models.

On the other hand, many existing explanation methods are based on assigning importance scores for different parts of the input to explain model predictions (Ribeiro et al., 2016; Lundberg & Lee, 2017b; Chen et al., 2018; Wang et al., 2019; Weinberger et al., 2020; Plumb et al., 2020). However, understanding the contribution of different input parts are usually not sufficiently informative for decision making: people often want to know what changes made to the input could have lead to a specific (desirable) prediction (Wachter et al., 2017b; Goyal et al., 2019; Nemirovsky et al., 2022). We call such changed input that could have shifted the prediction to a specific target *actionable counterfactual explanations*. Below we provide an example in the context of time series.

Example 1 (Actionable Counterfactual Explanation). *Suppose there is a model that takes as input a time series of breathing signal* $\mathbf{x} \in \mathbb{R}^T$ *from a subject of age* $u = 60$ *to predict the corresponding sleep stage as* $y^{pred} =$ *'Awake'* $\in \{$*'Awake'*, *'Light Sleep'*, *'Deep Sleep'*$\}$. *Typical methods assign importance scores to each entry of* $\mathbf{x}$ *to explain the prediction. However, they do not provide* actionable *counterfactual explanations on how to modify* $\mathbf{x}$ *to* $\mathbf{x}^{cf}$ *such that the prediction can change to* $y^{cf} =$ *'Deep Sleep'. An ideal method with such capability could provide more information on why the model make specific predictions.*

Actionable counterfactual explanations help people understand how to achieve a counterfactual (target) output by modifying the current model input. However, such explanations may not be sufficiently informative in practice, especially under the causal effect of confounding variables which are often immutable. Specifically, some variables can hardly be changed once its value has been determined, and suggesting changing such variables are both meaningless and infeasible (e.g., a patient age and gender when modeling medical time series). This leads to a stronger requirement: a good explanation should make as few changes as possible on immutable variables; we call such explanations *feasible counterfactual explanations*. Below we provide an example in the context of time series.

Example 2 (Feasible Counterfactual Explanation). *In Example 1, age* u *is a confounder that affects both* $\mathbf{x}$ *and* y *since elderly people (i.e., larger* u*) are more likely to*

[1]Department of Computer Science, Rutgers University. Correspondence to: Jingquan Yan <jy766@rutgers.edu>.

Proceedings of the 40th *International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

have irregular breathing $\mathbf{x}$ *and more 'Awake' time (i.e.,* y = *'Awake') at night. To generate a counterfactual explanation to change* y^{pred} *to 'Deep Sleep', typical methods tend to suggest decreasing the age* u *from* 60 *to* 50*, which is* infeasible *(since age cannot be changed in practice). An ideal method would first infer the age* u *and search for a* feasible *counterfactual explanation* $\mathbf{x}^{cf}$ *that could change* y^{pred} *to 'Deep Sleep' while keeping* u *unchanged.*

In this paper, we propose a self-interpretable time series prediction model, dubbed Counterfactual Time Series (CounTS), which can both (1) perform time series predictions and (2) provide actionable and feasible counterfactual explanations for its predictions. Under common causal structure assumptions, our method is guaranteed to identify the causal effect between the input and output in the presence of exogenous (confounding) variables, thereby improving the generated counterfactual explanations' feasibility. Our contribution is summarized as follows:

- We identify the actionability and feasibility requirements for generating counterfactual explanations for time series models and develop the first general self-interpretable method, dubbed CounTS, that satisfies such requirements.
- We provide theoretical guarantees that CounTS can identify the causal effect between the time series input and output in the presence of exogenous (confounding) variables, thereby improving feasibility in the generated explanations.
- Experiments on both synthetic and real-world datasets show that compared to state-of-the-art methods, CounTS significantly improves performance for generating counterfactual explanations while still maintaining comparable prediction accuracy.

2. Related Work

Interpretation Methods for Neural Networks. Various attribution-based interpretation methods have been proposed in recent years. Some methods focused on local interpretation (Ribeiro et al., 2016; Lundberg & Lee, 2017b; Plumb et al., 2018; Chen et al., 2018; Wang et al., 2019) while others are designed for global interpretation (Ghorbani et al., 2019; Natesan Ramamurthy et al., 2020). The main idea is to assign attribution, or importance scores, to the input features in terms of their impact on the prediction (output). For example, such importance scores can be computed using gradients of the prediction with respect to the input (Selvaraju et al., 2017; Lundberg & Lee, 2017a; Shrikumar et al., 2017; Sundararajan et al., 2017). Some interpretation methods are specialized for time series data; these include perturbation-based (Pan et al., 2021), rule-based (Rajapaksha & Bergmeir, 2022), and attention-based methods (Heo et al., 2018; Lim et al., 2021). One typical method, Feature Importance in Time (FIT), evaluates the importance of the input data based on the temporal distribution shift and unexplained distribution shift (Tonekaboni et al., 2020). However, these methods can only produce importance scores of the input features for the current prediction and therefore cannot generate counterfactual explanations (see Sec. 5 and Appendix D for empirical results).

Counterfactual Explanations for Time Series Models. There also works that generate counterfactual explanations for time series models. (Dhaou et al., 2021) proposed an association-rule algorithm to explain time series prediction by finding the frequent pairs of timestamps and generating counterfactual examples. (Nemirovsky et al., 2022) proposed a general explanation framework that generates counterfactual examples using residual generative adversarial networks (RGAN); it can be adapted for time series models. However, these works either fail to generate realistic counterfactual explanations (due to discretization error) or fail to generate feasible counterfactual explanations for time series models. In contrast, our CounTS as a principled variational causal method can naturally generate realistic and feasible counterfactual explanations. Such advantages are empirically verified in Sec. 5.

Bayesian Deep Learning and Variational Autoencoders. Our work is also related to the broad categories of variational autoencoders (VAEs) (Kingma & Welling, 2013) (which use inference networks to approximate posterior distributions) and Bayesian deep learning (BDL) (Wang & Yeung, 2016; 2020; Wang, 2017; Wang et al., 2015; Ding et al., 2021) models (which uses a deep component to process high-dimensional signals and a task-specific/graphical component to handle conditional/causal dependencies). (Lin et al., 2022) proposed the first VAE-based model for generating causal explanations for graph neural networks. (Louizos et al., 2017; Pawlowski et al., 2020) proposed the first VAE-based models for performing causal inference and estimating treatment effect. However, none of them addressed the problem of counterfactual explanation, which involves solving an inverse problem to obtain the optimal counterfactual input. In contrast, our CounTS is the first VAE-based model to address this challenge, with theoretical guarantees and promising empirical results. From the perspective of BDL (Wang & Yeung, 2016; 2020), CounTS uses deep neural networks to process high-dimension signals (i.e., the deep component in (Wang & Yeung, 2016)) and uses a Bayesian network to handle the conditional/causal dependencies among variables (i.e., the task-specific or graphical component in (Wang & Yeung, 2016)). Therefore, CounTS is also the first BDL model for generating counterfactual explanations.

3. Preliminaries

Causal Model. Following the definition in (Pearl, 2009), a causal model is described by a 3-tupple $M = \langle U, V, F \rangle$. U is a set of exogenous variables $\{u_1, \ldots, u_m\}$ that is not determined by any other variables in this causal model. V is a set of endogenous variables $\{v_1, \ldots, v_n\}$ that are determined by variables in $U \cup V$. We assume the causal model can be factorized according to a directed graph where each node represents one variable. F is a set of functions $\{f_1, \ldots, f_n\}$ describing the generative process of V:

$$v_i = f_i(pa_i, u_i), i = 1, \ldots, n,$$

where pa_i denotes direct parent nodes of v_i.

Counterfactual Inference. Counterfactual inference is interested in questions like "observing that $X = x$ and $Y = y$, what would be probability that $Y = y^{cf}$ if the input X had been x^{cf}?". Formally, given a causal model $\langle U, V, F \rangle$ where $Y, X \in V$, counterfactual inference proceeds in three steps (Pearl, 2009):

1. **Abduction.** Calculate the posterior distribution of u given the observation $X = x$ and $Y = y$, i.e., $P(u|X = x, Y = y)$.
2. **Action.** Perform causal intervention on the variable X, i.e., $do(X = x^{cf})$.
3. **Prediction.** Calculate the counterfactual probability $P(Y_{X=x^{cf}}(u) = y^{cf})$ with respect to the posterior distribution of $P(u|X = x, Y = y)$.

Putting the three steps together, we have

$$\begin{aligned} &P(Y_{X=x^{cf}} = y^{cf}|X = x, Y = y) \\ =&\sum\nolimits_u P(Y_{X=x^{cf}}(u) = y^{cf})P(u|x, y), \end{aligned}$$

where x^{cf} can be an counterfactual explanation describing what would have shifted the outcome Y from y to y^{cf}.

4. Method

In this section, we formalize the problem of counterfactual explanations for time series prediction and describe our proposed method for the problem.

Problem Setting. We focus on generating counterfactual explanations for predictions from time series models. We assume the model takes as input a multivariate time series $\mathbf{x}_i \in \mathbb{R}^{D \times T}$ and predicts the corresponding label $\mathbf{y}_i$, which can be a categorical label, a real value, or a time series $\mathbf{y}_i \in \mathbb{R}^T$. Given a specific input $\mathbf{x}_i$ and the model's prediction $\mathbf{y}_i^{pred}$, our goal is to explain the model by finding a counterfactual time series $\mathbf{x}_i^{cf} \neq \mathbf{x}_i$ that could have lead the model to an alternative (counterfactual) prediction $\mathbf{y}_i^{cf}$.

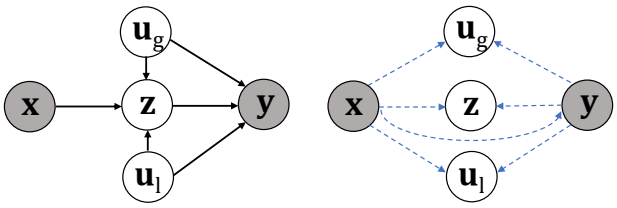

Figure 1. **Left:** The causal graph and generative model for CounTS. **Right:** Inference model for CounTS.

4.1. Learning of CounTS

Causal Graph and Generative Model for CounTS. Our self-interpretable model CounTS is based on the causal graph in Fig. 1(left), where $\mathbf{x} \in \mathbb{R}^{D \times T_{in}}$ is the input time series, $\mathbf{y}$ is the label, $\mathbf{z} \in \mathbb{R}^{H_z \times T_{mid}}$ is the representation of $\mathbf{x}$, $\mathbf{u}_l \in \mathbb{R}^{H_l \times T_{mid}}$ is the local exogenous variable, and $\mathbf{u}_g \in \mathbb{R}^{H_g}$ is the global exogenous variable. Both $\mathbf{u}_l$ and $\mathbf{u}_g$ are confounder variables; $\mathbf{u}_l$ can take different values in different time steps, while $\mathbf{u}_g$ is shared across all time steps.

This causal graph assumes the following factorization of the generative model $p_\theta(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$:

$$p_\theta(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{u}_l, \mathbf{u}_g)p_\theta(\mathbf{z}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{x})p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}), \quad (1)$$

where $\boldsymbol{\theta}$ denotes the collection of parameters for the generative model, and $p_\theta(\mathbf{u}_l, \mathbf{u}_g) = p_\theta(\mathbf{u}_l)p_\theta(\mathbf{u}_g)$. Here $p_\theta(\mathbf{z}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{x})$ and $p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z})$ are the encoder and the predictor, respectively. Specifically, we have

$$\begin{aligned} p_\theta(\mathbf{u}_l) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad p_\theta(\mathbf{u}_g) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ p_\theta(\mathbf{z}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{x}) = \mathcal{N}(\mu_z(\mathbf{u}_l, \mathbf{u}_g, \mathbf{x}; \boldsymbol{\theta}), \sigma_z(\mathbf{u}_l, \mathbf{u}_g, \mathbf{x}; \boldsymbol{\theta})), \\ p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}) = \mathcal{N}(\mu_y(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}; \boldsymbol{\theta}), \sigma_y(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}; \boldsymbol{\theta})), \end{aligned}$$

where $\mu_z(\cdot;\cdot)$, $\sigma_z(\cdot;\cdot)$, $\mu_y(\cdot;\cdot)$, $\sigma_y(\cdot;\cdot)$ are neural networks parameterized by $\boldsymbol{\theta}$. Note that for classification models the predictor $p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z})$ becomes a categorical distribution $Cat(f_y(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}; \boldsymbol{\theta}))$, where $f_y(\cdot;\cdot)$ is a neural network.

Inference Model for CounTS. We use an inference model $q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$ to approximate the posterior distribution of the latent variables, i.e., $p_\theta(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$. As shown in Fig. 1(right), we factorize $q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$ as

$$q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{y}|\mathbf{x})q_\phi(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y})q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}), \quad (2)$$

$$q_\phi(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y}) = q_\phi(\mathbf{u}_l|\mathbf{x}, \mathbf{y})q_\phi(\mathbf{u}_g|\mathbf{x}, \mathbf{y}) \quad (3)$$

where $\boldsymbol{\phi}$ is the collection of the inference model's parameters, and . We parameterized each factor in Eqn. 2 as

$$\begin{aligned} q_\phi(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mu_y(\mathbf{x}; \boldsymbol{\phi}), \sigma_y^2(\mathbf{x}; \boldsymbol{\phi})), \\ q_\phi(\mathbf{u}_l|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mu_{u_l}(\mathbf{x}, \mathbf{y}; \boldsymbol{\phi}), \sigma_{u_l}^2(\mathbf{x}, \mathbf{y}; \boldsymbol{\phi})), \\ q_\phi(\mathbf{u}_g|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mu_{u_g}(\mathbf{x}, \mathbf{y}; \boldsymbol{\phi}), \sigma_{u_g}^2(\mathbf{x}, \mathbf{y}; \boldsymbol{\phi})), \\ q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mu_z(\mathbf{x}, \mathbf{y}; \boldsymbol{\phi}), \sigma_z^2(\mathbf{x}, \mathbf{y}; \boldsymbol{\phi})), \end{aligned}$$

where $\mu_\cdot(\cdot;\cdot)$ and $\sigma_\cdot(\cdot;\cdot)$ denote neural networks with ϕ as their parameters.

Evidence Lower Bound. Our CounTS uses the evidence lower bound (ELBO) $\mathcal{L}_{ELBO}$ of the log likelihood $\log p(\mathbf{y}|\mathbf{x})$ as an objective to learn the generative and inference models. Maximizing the ELBO is equivalent to learning the optimal variational distribution $q_\phi(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x}) = \int q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})d\mathbf{y}$ that best approximates the posterior distribution of the label and latent variables $p_\theta(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$. Specifically, we have

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q_\phi(\mathbf{y},\mathbf{u}_l,\mathbf{u}_g,\mathbf{z}|\mathbf{x})}[p_\theta(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})] - \mathbb{E}_{q_\phi(\mathbf{y},\mathbf{u}_l,\mathbf{u}_g,\mathbf{z}|\mathbf{x})}[q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})]. \quad (4)$$

Note that different from typical ELBOs, we explicitly involves $\mathbf{y}$ and use $q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$ rather than $q_\phi(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})$ (see Appendix A for details); this is to expose the factor $q_\phi(\mathbf{y}|\mathbf{x})$ in Eqn. 2 to allow for additional supervision on $\mathbf{y}$ (more details below). With the factorization in Eqn. 1 and Eqn. 2, we can decompose Eqn. 4 as:

$$\begin{aligned}
\mathcal{L}_{ELBO} = \ & \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{u}_l,\mathbf{u}_g,\mathbf{z}|\mathbf{x},\mathbf{y})}[p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z})] && (5)\\
& - \mathbb{E}_{q_\phi(\mathbf{y},\mathbf{u}_l,\mathbf{u}_g|\mathbf{x})}\left[KL[q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{z}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{x})]\right] && (6)\\
& - \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})}\left[KL[q_\phi(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{u}_l, \mathbf{u}_g)]\right] && (7)\\
& - \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})}[q_\phi(\mathbf{y}|\mathbf{x})], && (8)
\end{aligned}$$

where each term is computed with our neural network parameterization; Fig. 2(left) shows the network structure. Below we briefly discuss the intuition of each term.

(1) **Eqn. 5 predicts** the label $\mathbf{y}$ using $\mathbf{u}_l$, $\mathbf{u}_g$, and $\mathbf{z}$ inferred from $\mathbf{x}$ ($\mathbf{y}$ is marginalized out).
(2) **Eqn. 6 regularizes** the inference model $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ to get closer to the generative model $p_\theta(\mathbf{z}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{x})$.
(3) **Eqn. 7 regularizes** $q_\phi(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y})$ using the prior distribution $p(\mathbf{u}_l, \mathbf{u}_g)$.
(4) **Eqn. 8 regularizes** $q_\phi(\mathbf{y}|\mathbf{x})$ by maximizing its entropy, preventing it from collapsing to deterministic solutions.

Final Objective Function. Inspired by (Louizos et al., 2017), we include an additional term (N is the training set size)

$$\mathcal{L}_y = \sum\nolimits_{i=1}^{N} \log q(\mathbf{y}_i|\mathbf{x}_i) \quad (9)$$

to supervise $q(\mathbf{y}|\mathbf{x})$ using label $\mathbf{y}$ during training; this helps produce more accurate estimation for $\mathbf{u}_l$, $\mathbf{u}_g$, $\mathbf{z}$ using $q_\phi(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y})$ and $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$. The final objective function then becomes

$$\mathcal{L}_{CounTS} = \mathcal{L}_{ELBO} + \lambda\mathcal{L}_y, \quad (10)$$

where λ is a hyperparameter balancing both terms.

4.2. Inference Using CounTS

After learning both the generative model (Eqn. 1) and the inference model (Eqn. 2) by maximizing $\mathcal{L}_{CounTS}$ in Eqn. 13, our **self-interpretable** CounTS can perform inference to (1) make **predictions** on $\mathbf{y}$ using the yellow branch in Fig. 2(left), and (2) generate **counterfactual explanation** $\mathbf{x}^{cf}$ for any target label $\mathbf{y}^{cf}$ using the blue branch in Fig. 2(right).

4.2.1. Prediction

We use the yellow branch in Fig. 2(left) to *predict* $\mathbf{y}$. Specifically, given an input time series $\mathbf{x}$, the encoder will first infer the *initial label* $\mathbf{y}$ using $q_\phi(\mathbf{y}|\mathbf{x})$ and then infer $(\mathbf{u}_l, \mathbf{u}_g)$ and $\mathbf{z}$ from $\mathbf{x}$ and $\mathbf{y}$ using $q_\phi(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x}, \mathbf{y})$. $(\mathbf{u}_l, \mathbf{u}_g)$ and $\mathbf{z}$ are then fed into the predictor $p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z})$ to infer the *final* label $\mathbf{y}$. Formally, CounTS predict the label as

$$\mathbf{y}^{pred} = \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{u}_l,\mathbf{u}_g,\mathbf{z}|\mathbf{x},\mathbf{y})}[p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z})]. \quad (11)$$

Empirically, we find that directly using the means of $q_\phi(\mathbf{y}|\mathbf{x})$ and $q_\phi(\mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x}, \mathbf{y})$ as input to $p_\theta(\mathbf{y}|\mathbf{u}_l, \mathbf{u}_g, \mathbf{z})$ already achieves satisfactory accuracy.

4.2.2. Generating Counterfactual Explanation

We use the blue branch in Fig. 2(right) to generate counterfactual explanations via counterfactual inference. Our goal is to find the optimal counterfactual explanation $\mathbf{x}^{cf}$ defined below.

Definition 4.1 (Optimal Counterfactual Explanation). *Given a factual observation $\mathbf{x}$ and prediction $\mathbf{y}^{pred}$, the optimal counterfactual explanation $\mathbf{x}^{cf}$ for the counterfactual outcome for $\mathbf{y}^{cf}$ is*

$$\mathbf{x}^{cf} = \text{argmax}_{\mathbf{x}'}\ p(Y_{\mathbf{x}=\mathbf{x}'}(\mathbf{u}) = \mathbf{y}^{cf}),$$

where $\mathbf{u} = (\mathbf{u}_l, \mathbf{u}_g)$ and the counterfactual likelihood is defined as

$$\begin{aligned}
& p(Y_{\mathbf{x}=\mathbf{x}'}(\mathbf{u}) = \mathbf{y}^{cf}) && (12)\\
& = \sum_{\mathbf{u}} p\left(\mathbf{y} = \mathbf{y}^{cf}|do\left(\mathbf{x} = \mathbf{x}'\right), \mathbf{u}\right) p(\mathbf{u}|\mathbf{x} = \mathbf{x}, \mathbf{y} = \mathbf{y}^{pred}).
\end{aligned}$$

In words, we search for the optimal $\mathbf{x}^{cf}$ that would have shifted the model prediction from $\mathbf{y}^{pred}$ to $\mathbf{y}^{cf}$ while keeping $(\mathbf{u}_l, \mathbf{u}_g)$ unchanged. Since the definition of counterfactual explanations in Definition 4.1 involves causal inference with the intervention on $\mathbf{x}$, we need to first *identify* the causal probability $p(\mathbf{y} = \mathbf{y}^{cf}|do(\mathbf{x} = \mathbf{x}'), \mathbf{u})$ using observational probability, i.e., removing 'do' in the equation. The theorem below shows that this is achievable.

Theorem 4.1 (Identifiability). *Given the posterior distribution of exogenous variable $p(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y})$, the effect of action $p(\mathbf{y} = \mathbf{y}^{cf}|do(\mathbf{x} = \mathbf{x}'), \mathbf{u}_l, \mathbf{u}_g)$ can be identified using $\mathbb{E}_{p(\mathbf{z}|\mathbf{x}',\mathbf{u}_l,\mathbf{u}_g)}[p\left(\mathbf{y}^{cf}|\mathbf{z}, \mathbf{u}_l, \mathbf{u}_g\right)]$.*

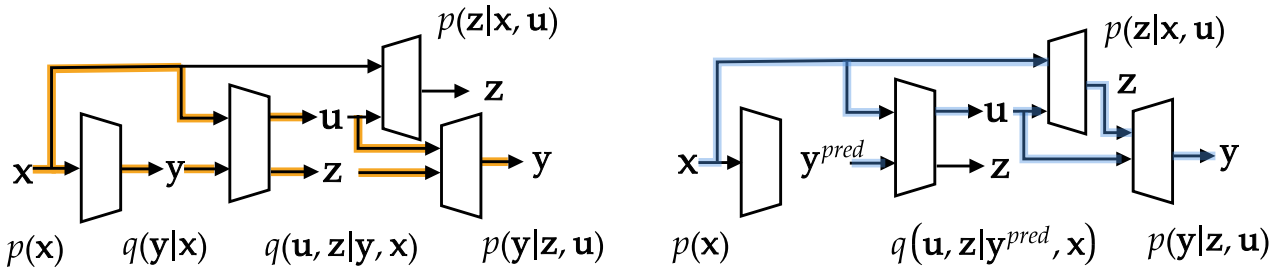

Figure 2. Network structure. We omit subscripts of p_θ and q_ϕ for clarity. **Left:** CounTS makes *predictions* on $\mathbf{y}$ using the yellow branch. **Right:** Given the current $\mathbf{x}$ and $\mathbf{y}^{pred}$, CounTS generates *counterfactual explanation* $\mathbf{x}^{cf}$ for any target label $\mathbf{y}^{cf}$ using the blue branch.

See Appendix B.1 for the proof. With Theorem 4.1, we can rewrite Eqn. 12 as

$$\mathcal{L}_{cf} = \mathbb{E}_{p(\mathbf{u}|\mathbf{x}=\mathbf{x},\mathbf{y}=\mathbf{y}^{pred})}\mathbb{E}_{p(\mathbf{z}|\mathbf{x}',\mathbf{u})}[p(\mathbf{y}^{cf}|\mathbf{z},\mathbf{u})], \quad (13)$$

where $\mathbf{u} = (\mathbf{u}_l, \mathbf{u}_g)$ and $p(\mathbf{u}|\mathbf{x} = \mathbf{x}, \mathbf{y} = \mathbf{y}^{pred})$ is approximated by $q_\phi(\mathbf{u}|\mathbf{x} = \mathbf{x}, \mathbf{y} = \mathbf{y}^{pred})$. We use Monte Carlo estimates to compute the expectation in Eqn. 12 and Eqn. 13, iteratively compute the gradient $\frac{\partial \mathcal{L}_{cf}}{\partial \mathbf{x}'}$ (via back-propagation) to search for the optimal $\mathbf{x}'$, and use it as $\mathbf{x}^{cf}$ (see the complete algorithm in Appendix B.2).

5. Experiments

In this section, we evaluate our CounTS and existing methods on two *synthetic* and three *real-world datasets*. For each dataset, we evaluate different methods in terms of three metrics: (1) **prediction accuracy**, (2) **counterfactual accuracy**, and (3) **counterfactual change ratio**, with the last one as the most important metric. These metrics take different forms for different datasets (see details in Sec. 5.2-5.4).

5.1. Baselines and Implementations

We compare our CounTS with state-of-the-art methods for generating explanations for deep learning models, including Regularized Gradient Descent (**RGD**) (Wachter et al., 2017a), Gradient-weighted Class Activation Mapping (**GradCAM**) (Selvaraju et al., 2017), Gradient SHapley Additive exPlanations (**GradSHAP**) (Lundberg & Lee, 2017a), Local Interpretable Model-agnostic Explanations (**LIME**) (Ribeiro et al., 2016), Feature Importance in Time (**FIT**) (Tonekaboni et al., 2020), Case-crossover APriori (**CAP**) (Dhaou et al., 2021), and Counterfactual Residual Generative Adversarial Network (**CounteRGAN**) (Nemirovsky et al., 2022) (see Appendix C.2 for more details). Note that among these baselines, only RGD and CounteRGAN can generate actionable explanations. Other baselines, including FIT (which is designed for time series models), only provide importance scores as explanations; therefore some evaluation metrics may not be applicable for them (shown as '-' in tables).

All methods above are implemented with PyTorch (Paszke et al., 2019). For fair comparison, the prediction model in all the baseline explanation methods has the same neural network architecture as the inference module in our CounTS. See the Appendix for more details on the architecture, training, and inference.

Table 1. **Results on the toy dataset.** We mark the best result with **bold face** and the second best results with underline.

	CounteRGAN	RGD	GradCAM	GradSHAP	LIME	FIT	CAP	CounTS (Ours)
Pred. Accuracy (%) ↑	**85.19**							<u>83.26</u>
CCR ↑	<u>1.25</u>	1.21	1.09	1.13	1.2	1.15	0.97	**1.33**
Counterf. Accuracy (%) ↑	**78.75**	<u>77.96</u>	-	-	-	45.62	-	70.28

5.2. Toy Dataset

Dataset Description. We designed a toy dataset where the label is affected by only part of the input. A good counterfactual explanation should only modify this part of the input while keeping the other part unchanged. Following the causal graph in Fig. 1(left), we have input $\mathbf{x} \in \mathbb{R}^{12}$ with each entry independently sampled from $\mathcal{N}(\mu_x, \sigma_x^2)$ and an exogenous variable $u \in \mathbb{R}$ (as the confounder) sampled from $\mathcal{N}(\mu_u, \sigma_u^2)$. To indicate which part of $\mathbf{x}$ affects the label, we introduce a mask vector $\mathbf{m} \in \mathbb{R}^{12}$ with first 6 entries $\mathbf{m}_{1:6}$ set to 1 and the last 6 entries $\mathbf{m}_{7:12}$ set to 0. We then generate $\mathbf{z} \in \mathbb{R}^{12}$ as $\mathbf{z} = \mathbf{u} \cdot (\mathbf{m} \odot \mathbf{x})$ and the label $y \sim Bern(\sigma(\mathbf{z}^\top \mathbf{1} + u))$ where σ is the sigmoid function. Here $\mathbf{x}$'s first 6 entries $\mathbf{x}_{1:6}$ is label-related and the last 6 entries $\mathbf{x}_{7:12}$ is label-agnostic.

Evaluation Metrics. We use three evaluation metrics:

- **Prediction Accuracy.** This is the percentage of time series correctly predicted ($y^{pred} = y$) in the test set.
- **Counterfactual Accuracy.** For a prediction model f, the generated counterfactual explanation $\mathbf{x}^{cf}$, and the target label y^{cf}, counterfactual accuracy is the percentage of time series where $\mathbf{x}^{cf}$ successfully change the model's prediction to y^{cf} (i.e., $f(\mathbf{x}^{cf}) = y^{cf}$).
- **Counterfactual Change Ratio (CCR).** This measures how well the counterfactual explanation $\mathbf{x}^{cf}$ changes the label-related input $\mathbf{x}_{1:6}$ while keeping the label-agnostic input $\mathbf{x}_{7:12}$ unchanged. Formally we use the average ratio of $\frac{\|\mathbf{x}_{1:6}^{cf} - \mathbf{x}_{1:6}\|_1}{\|\mathbf{x}_{7:12}^{cf} - \mathbf{x}_{7:12}\|_1}$ across the test set.

Note that CCR is the most important metric among the three

Table 2. **Results on the *Spike* dataset.** We mark the best result with **bold face** and the second best results with underline.

		CounteRGAN	RGD	GradCAM	GradSHAP	LIME	FIT	CAP	CounTS (Ours)
Pred. MSE ↓		0.128							**0.117**
CCR ↑	2 Active	2.206	2.217	2.043	1.989	1.872	1.863	1.614	**2.322**
	1 Active	0.705	0.683	0.654	0.615	0.473	0.55	0.497	**0.730**
Counterf. MSE ↓		**0.074**	**0.074**	-	-	-	0.394	-	0.103

since our main focus is to generate actionable and feasible counterfactual explanations.

Quantitative Results. Table 1 compares our CounTS with the baselines in terms of the three metrics. CounTS outperforms all the baselines in terms of CCR with minimal prediction accuracy loss. This shows that our CounTS successfully identifies and fixes the exogenous variables u and $\mathbf{m}$ to generate actionable and feasible counterfactual explanations while still achieving prediction accuracy comparable to baselines. Note that actionable methods (i.e., CounTS, RGD, and CounteRGAN) outperforms importance score methods (i.e., FIT, LIME, and GradSHAP). This is expected because importance-score-based explanations can only explain the original prediction and therefore fail to infer what change on $\mathbf{x}$ will shift the prediction from y^{pred} to y^{cf}.

Our counterfactual accuracy is lower than CounteRGAN and RGD methods. This is reasonable since these baselines do not infer and fix the posterior distribution $p(\mathbf{u}|\mathbf{x}, \mathbf{y})$ and are therefore more flexible for the generator (or back-propagation) to modify their input to push y^{pred} closer to the target y^{cf}. However, such flexibility comes at the cost of low feasibility, reflected in their poor CCR performance.

5.3. Spike Dataset

Dataset Description. Inspired by (Tonekaboni et al., 2020), we construct the *Spike* synthetic dataset. In the dataset, each time series contains 3 channel, i.e., $\mathbf{x} \in \mathbb{R}^{3\times T}$. Each channel is a independent non–linear auto-regressive moving average (NARMA) sequence with randomly distributed spikes. The label sequence $\mathbf{y} \in \mathbb{R}^T$ starts at 0; it *may* switch to 1 as soon as there is a spike observed in any of the 3 channels and remain 1 until the last timestamp. A 3-dimensional exogenous variable $\mathbf{u} \in \{0, 1\}^3$ determines whether the spike in the channel can affect the final output; if so, we say the channel is *active*. Each of the three entries in $\mathbf{u}$, i.e., is sampled independently from three different Bernoulli distributions with parameters 0.8, 0.4, and 0, respectively (see more details on the dataset in Appendix C.1).

Evaluation Metrics. We use three evaluation metrics:

- **Prediction MSE.** We use the mean square error (MSE) $\frac{1}{N}\sum_i^N \|\mathbf{y}_i^{pred} - \mathbf{y}_i\|_2^2$ to measure the prediction error in the test set with N time series.
- **Counterfactual MSE.** Similar to Sec. 5.2, for a prediction model f, the generated counterfactual explanation $\mathbf{x}^{cf}$, and the target label $\mathbf{y}^{cf}$, counterfactual MSE is defined as $\frac{1}{N}\sum_i^N \|f(\mathbf{x}_i^{cf})) - \mathbf{y}_i^{cf}\|_2^2$; it measures how successfully $\mathbf{x}^{cf}$ changes the model's prediction to y^{cf}.
- **CCR.** For a given input $\mathbf{x}_i$, we set the target counterfactual label $\mathbf{y}_i^{cf}$ by shifting $\mathbf{y}^{pred}$ by 20 timestamps to the right. If at timestamp t, there is a spike in an active channel in $\mathbf{x}_i$ triggering the output $\mathbf{y}_i^{pred}$ to switch from 0 to 1, an ideal counterfactual explanation $\mathbf{x}_i^{cf}$ should (1) suppress all spikes between $[t, t+20)$, (2) create a new spike at $t+20$ timestamp in all active channels of the original input $\mathbf{x}_i$, and (3) keep all inactive channels unchanged. Therefore the counterfactual change ratio can be defined as (with N time series): $CCR = \frac{1}{N}\sum_{i=1}^N \frac{\|\mathbf{m}_i \odot (\mathbf{x}_i - \mathbf{x}_i^{cf})\|_1}{\|(\mathbf{1}-\mathbf{m}_i)\odot(\mathbf{x}_i - \mathbf{x}_i^{cf})\|_1}$, where $\mathbf{m}_i = [\mathbf{u}, \mathbf{u}, \ldots, \mathbf{u}] \in \mathbb{R}^{3\times T}$ repeats $\mathbf{u}$ in each time step to mask inactive channels.

Note that the scale of CCR will depend on the number of active channels, we therefore report our results for time series with 1 and 2 active channels. Higher CCR indicates better performance.

Quantitative Results. Table 2 shows the results for different methods in the *Spike*. Similar to the toy dataset, our CounTS outperforms all baselines in terms of CCR, and actionable methods (i.e., CounTS, RGD, and CounteRGAN) outperforms importance score methods (i.e., FIT, LIME, and GradSHAP) thanks to the former's capability of modifying the input to shift the prediction towards the counterfactual target label.

Interestingly, besides promising performance in terms of CCR, our CounTS can also improve prediction performance, achieving lower prediction MSE. This is potentially due to CounTS's ability to model the exogenous variable $\mathbf{u}$ that decides whether a spike in a specific channel affects the label $\mathbf{y}$. Similar to the toy datset, we notice both CounteRGAN and RGD methods have lower counterfactual MSE (both at 0.074) than CounTS because they do not need to infer and fix the exogenous variable $\mathbf{u}$ (i.e., the mask) and therefore have more flexibility to modify the input $\mathbf{x}$. Since both CounteRGAN and RGD method are unaware of the mask, they suffer from lower CCR and produce more unwanted modification in inactive channels (more details below).

Qualitative Results. Fig. 3 shows an example time series as a case study. In this example, only channel 1 (blue) is active and thus the spike in channel 1 at timestamp $t = 16$ will flip the output $\mathbf{y}$ from 0 to 1. Both CounTS's and RGD's predictions $\mathbf{y}^{pred}$ (blue in Column 1) are very close to the ground truth.

Following the evaluation protocol above, we set our counterfactual (target) label $\mathbf{y}^{cf}$ by shifting $\mathbf{y}^{pred}$ by 20 timestamps to the right and padding 0s on the left (yellow). We should expect an ideal counterfactual explanation to have no spike in $t = [16, 36)$ and a new spike at $t = 36$ in channel 1. Since channel 2 and 3 are inactive, there should not

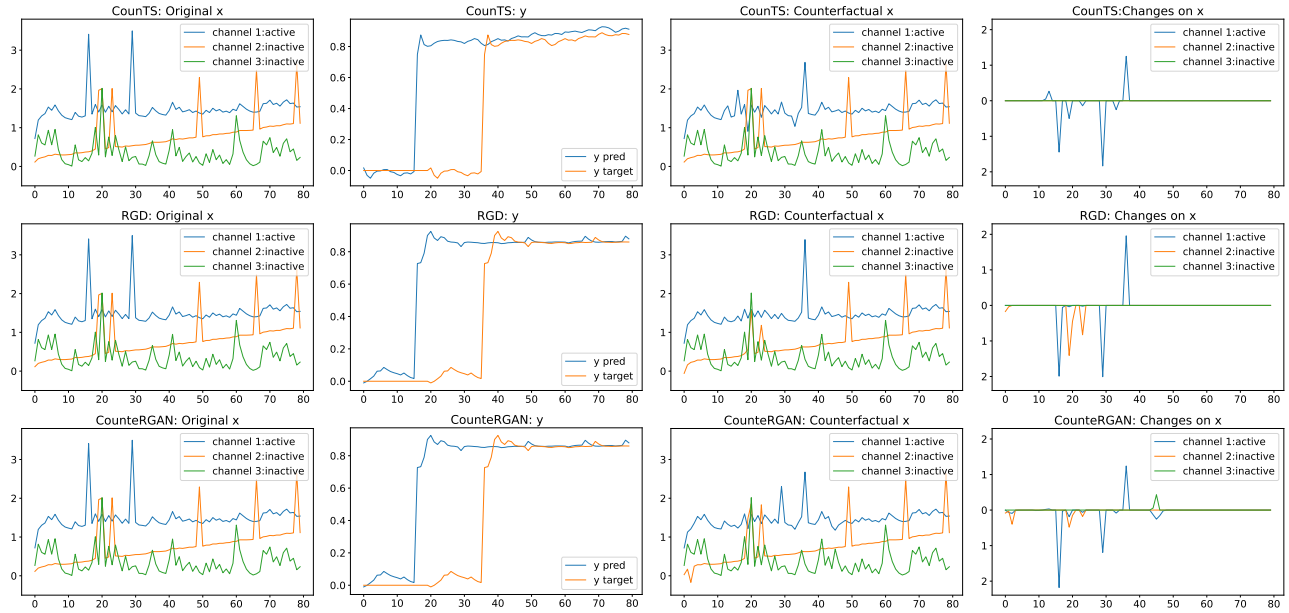

Figure 3. Qualitative results on the *Spike* dataset. **Column 1** shows the original input $\mathbf{x}$ for CounTS, RGD, and CounteRGAN, respectively. **Column 2** shows the predictions $\mathbf{y}^{pred}$ and counterfactual (target) labels $\mathbf{y}^{cf}$. The prediction for RGD and CounteRGAN are identical since they use the same prediction model. **Column 3** shows the counterfactual input $\mathbf{x}^{cf}$. **Column 4** shows the changes on input, i.e., $\mathbf{x}^{cf} - \mathbf{x}$. FIT *cannot* provide actionable explanations (see the importance score generated by FIT in Appendix D).

Table 3. **Average CCR for both intra-dataset and cross-dataset settings.** The average CCR is calculated over 6 counterfactual action settings (A→D, L→D, R→D, R→A, D→A, and L→A) for every model and dataset setting. The *last row* are the average CCR over all 6 dataset settings for every method.

	CounteRGAN	RGD	GradCAM	GradSHAP	LIME	FIT	CAP	CounTS (Ours)
→SOF	1.153	1.135	1.126	0.908	0.932	1.072	1.034	**1.313**
→SHHS	1.231	1.241	1.056	0.975	1.020	1.193	0.929	**1.390**
→MESA	1.285	1.188	1.031	1.025	1.007	1.142	1.031	**1.377**
SOF	1.126	1.088	0.887	0.996	1.078	1.053	0.963	**1.281**
SHHS	1.133	1.111	0.890	1.068	1.059	1.067	1.000	**1.333**
MESA	1.163	1.166	0.883	1.042	0.987	1.089	0.962	**1.329**
Average	1.182	1.155	0.979	1.002	1.014	1.103	0.987	**1.337**

be any changes. We can see all actionable methods (i.e., CounTS, CounteRGAN, and RGD) try to reduce the spikes in $t = [16, 36)$. However, CounteRGAN fails to remove the spike at $t = 29$; both RGD and CounteRGAN makes undesirable changes in inactive channels 2 and 3. Compared to baselines, our CounTS shows a significant advantage at inferring the active channel and suppressing the changes in inactive channels. This demonstrates that CounTS is more actionable and feasible than the existing time-series explanation methods. Note that FIT is *not* an actionable explanation method and therefore can only provide importance scores to explain the current prediction $\mathbf{y}^{pred}$ (see Appendix D Fig. 5 for details).

5.4. Real-World Datasets

Dataset Description. We evaluate our model on three real-world medical datasets: Sleep Heart Health Study (*SHHS*) (Quan et al., 1997), Multi-ethnic Study of Atherosclerosis (*MESA*) (Zhang et al., 2018a), and Study of Osteoporotic Fractures (*SOF*) (Cummings et al., 1990), each containing subjects' full-night breathing breathing signals. The breathing signals are divided into 30-second segments; each segment has one of the sleep stages ('Awake', 'Light Sleep', 'Deep Sleep', 'Rapid Eye Movement (REM)') as its label. In total, there are 2,651, 2,055 and 453 patients in SHHS, MESA, and SOF, respectively, with an average of 1,043 breathing signal segments (approximately 8.7 hours).

Table 4. **CCR for each counterfactual action setting in *SHHS*.**

	A→D	L→D	R→D	R→A	D→A	L→A	Average
CounteRGAN	1.119	1.144	1.055	1.298	1.125	1.059	1.133
RGD	1.006	1.130	0.985	1.180	1.155	1.201	1.111
GradCAM	0.882	0.790	0.859	0.954	0.882	0.977	0.890
GradSHAP	1.100	1.105	0.903	1.171	1.100	1.027	1.068
LIME	0.855	**1.178**	1.067	1.125	0.855	1.273	1.059
FIT	0.948	1.012	1.215	1.405	0.948	0.877	1.067
CAP	0.885	0.829	1.236	0.967	0.885	1.199	1.000
CounTS (Ours)	**1.244**	1.159	**1.349**	**1.515**	**1.324**	**1.407**	**1.333**

Corresponding to our causal graph in Fig. 1(left), $\mathbf{x}$ is the original breathing signal, $\mathbf{z}$ is signal patterns (representation) learned from the encoder, $\mathbf{u}$ can be 'gender', 'age', and even 'dataset index' ('dataset index' will be used during cross-dataset prediction; more details later), and $\mathbf{y}$ is the sleep stage label. Intuitively, 'gender', 'age', and 'dataset index' (due to different experiment instruments and signal measurement) can have causal effect on both the breathing signal patterns (subjects of different ages can have different breathing frequencies and magnitudes) and sleep stages (elderly subjects have less 'Deep Sleep' at night).

Evaluation Metrics. For brevity, we use abbreviations 'A',

'R', 'L', and 'D' to represent the four sleep stages 'Awake', 'Rapid Eye Movement (REM)', 'Light Sleep', 'Deep Sleep', respectively. We use three evaluation metrics:

- **Prediction Metric** and **Counterfactual Metric** are similar to those in the toy classification dataset.
- **CCR.** Since ground-truth confounders are not available in real-world datasets, we propose to compute CCR on two consecutive 30-second segments with different sleep stages. For example, given the two segments $[\mathbf{x}_{i,1}, \mathbf{x}_{i,2}]$ with the corresponding two predicted sleep stages $[\mathbf{y}_{i,1}^{pred}, \mathbf{y}_{i,2}^{pred}] = [A, D]$, we set the counterfactual labels $[\mathbf{y}_{i,1}^{cf}, \mathbf{y}_{i,2}^{cf}] = [A, A]$; we refer to this setting as D→A. An ideal counterfactual explanation $[\mathbf{x}_{i,1}^{cf}, \mathbf{x}_{i,2}^{cf}]$ should shift the model prediction from $[A, D]$ to $[A, A]$ while keeping the first segment unchanged (i.e., smaller $\|\mathbf{x}_{i,1} - \mathbf{x}_{i,1}^{cf}\|_1$). CCR can therefore be computed as (with N segment pairs in the test set): $CCR = \frac{1}{N}\sum_{i=1}^{N} \frac{\|\mathbf{x}_{i,2} - \mathbf{x}_{i,2}^{cf}\|_1}{\|\mathbf{x}_{i,1} - \mathbf{x}_{i,1}^{cf}\|_1}$.

In the experiments, we focus on cases where the target label $\mathbf{y}_{i,2}^{cf}$ is either 'Awake' or 'Deep Sleep' (two extremes of sleep stages), and evaluated 6 *counterfactual action settings*, namely A→D, L→D, R→D, R→A, D→A, and L→A.

Cross-Dataset and Intra-Dataset Settings. Since the three datasets are collected with different devices and procedures, we consider the dataset index as an additional exogenous variable in $\mathbf{u}$, which has causal effect on both the learned representation $\mathbf{z}$ and the sleep stage $\mathbf{y}$. This leads to two different settings: the intra-dataset setting (without the dataset index confounder) and the cross-dataset setting (with the dataset index confounder).

- **Intra-Dataset Setting.** Training and test sets are from same dataset, and we do not involve the dataset index as part of the exogenous variable (confounder) $\mathbf{u}$.
- **Cross-Dataset Setting.** We choose two of the datasets as the source datasets, with the remaining one as the target dataset (e.g., *MESA+SHHS→SOF*). We use all source datasets (e.g., *SHHS* and *MESA*) and 10% of the target dataset (e.g., *SOF*) as the training set and use the remaining 90% of the target dataset as the test set. We treat the dataset index as part of $\mathbf{u}$, which is predicted during both training and inference.

Quantitative Results. Table 3 shows the average CCR of different methods for both intra-dataset and cross-dataset settings. Results show that our CounTS outperforms the baselines in all the dataset settings in terms of the average CCR. This demonstrates CounTS's capability of generating feasible and actionable explanations in complex real-world datasets. Table 4 shows the detailed CCR for each counterfactual action setting in *SHHS*, where CounTS is leading in almost all counterfactual action settings. Detailed results for other datasets can be found in Table 7∼11 of Appendix D.

Table 5. **Prediction accuracy for both intra-dataset and cross-dataset settings.** All baselines (e.g., RGD, CounteRGAN, and FIT) explain the same prediction model and therefore share the same prediction accuracy. '→*SHHS*' means '*MESA+SOF→SHHS*' and similarly for '→*MESA*' and '→*SOF*'.

Method	→*SOF*	→*SHHS*	→*MESA*	*SOF*	*SHHS*	*MESA*
All Baselines	0.702	0.747	**0.768**	0.729	**0.801**	**0.813**
CounTS (Ours)	**0.719**	**0.753**	0.741	**0.732**	0.789	0.799

Table 6. **Average counterfactual accuracy for both intra-dataset and cross-dataset settings.** The average is calculated over all 6 counterfactual action settings.

	→SOF	→SHHS	→MESA	SOF	SHHS	MESA	Average
RGD	0.913	**0.887**	**0.907**	**0.862**	**0.907**	**0.892**	**0.895**
CounteRGAN	0.897	0.868	0.880	0.844	0.888	0.868	0.874
FIT	0.331	0.327	0.346	0.248	0.322	0.294	0.311
CounTS (Ours)	**0.916**	**0.887**	0.889	0.852	0.903	0.887	0.889

Table 5 shows the prediction accuracy for both intra-dataset and cross-dataset settings. Similar to Table 1 and Table 2, all baselines (e.g., RGD, CounteRGAN, and FIT) explain the same prediction model and therefore share the same prediction accuracy. Results show that CounTS achieves prediction accuracy comparable to the original prediction model. Table 6 shows the average counterfactual accuracy for different dataset settings over 6 counterfactual action settings. As in the toy and *Spike* datasets, RGD achieves higher counterfactual accuracy because it does not need to infer and fix the exogenous variable $\mathbf{u}$ and hence enjoys more flexibility when modifying the input $\mathbf{x}$; this often leads to undesirable changes (e.g., breathing frequency and amplitude which are related to the subject's age), less feasible counterfactual explanations, and therefore worse CCR performance. In contrast, breathing frequency and amplitude are potentially captured by $\mathbf{u}_l$ and $\mathbf{u}_g$ in CounTS and kept unchanged (see qualitative results below). Detailed counterfactual accuracy for every counterfactual action setting and every dataset setting can be found in Appendix D.

Qualitative Analysis: Problem Setting. As some background on sleep staging, note that a patient's breathing signal will be much more periodic in 'Deep Sleep' than in 'Awake'. Fig. 4 shows an example time series from the MESA dataset as a case study. It contains two consecutive segments of breathing signals (60 seconds in total). The first segment ($0 \sim 30$ seconds) is regular and periodic, and therefore both CounTS and baselines correctly predict its label as 'Deep Sleep' ('D'); the second segment ($30 \sim 60$ seconds) is irregular and therefore both CounTS and baselines correctly predict its label as 'Awake' ('A'). The goal is to generate a counterfactual explanation such that the model predicts the *second* segment as 'Deep Sleep'.

Qualitative Analysis: Ideal Counterfactual Explanation. Since a 'Deep Sleep' segment should be more periodic, an

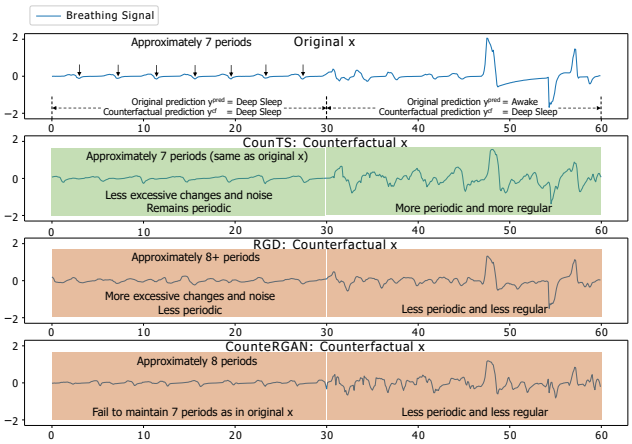

Figure 4. Example on real-world dataset from *MESA* intra-dataset experiment. For this example, $\mathbf{y}^{pred}$ is [D,A] and target $\mathbf{y}^{cf}$ is [D,D] since we assume 'Deep Sleep' and 'Active' have the most different patterns. FIT *cannot* provide actionable explanations (see the importance score generated by FIT in Appendix D).

ideal counterfactual explanation should keep the first segment ($0 \sim 30$ seconds) *unchanged* and maintain its *periodicity*, make the second segment ($30 \sim 60$ seconds) *more periodic* (the pattern for 'Deep Sleep'), and keep the breathing frequency (number of breathing cycles) unchanged throughout both segments (this is related to "feasibility" since a patient usually has the same breathing frequency during 'Deep Sleep').

Qualitative Analysis: Detailed Results. Fig. 4 compares the generated counterfactual explanations from CounTS, RGD, and CounteRGAN, and we can see that our CounTS's explanation is closer to the ideal case. Specifically:

1. **The First 30-Second Segment:** In the original breathing signal, the first 30-second segment ($0 \sim 30$ seconds) is periodic and contains 7 breathing cycles (i.e., 7 periods). Our CounTS managed to keep the first 30-second segment signal mostly unchanged, maintaining its periodicity and 7 breathing cycles. In contrast, RGD changes the first segment to the point that its periodicity is mostly lost. CounteRGAN can maintain the first segment's periodicity; however it increases the number breathing cycles from 7 to 8, making it *infeasible*.
2. **The Second 30-Second Segment:** In the original breathing signal, the second 30-second segment ($30 \sim 60$ seconds) is irregular since the patient is in the 'Awake' sleep stage. CounTS managed to make the 'Awake' (i.e., 'Active') signal much more periodic, successfully steering the prediction to 'Deep Sleep'. In contrast, RGD and CounteRGAN failed to make the second 30-second segment ($30 \sim 60$ seconds) more periodic.
3. **Capturing and Maintaining Global Temporal Patterns:** Meanwhile, we can observe that CounTS's explanation in the second 30-second segment ($30 \sim 60$ seconds) has a much more similar breathing frequency (measured by the number of breathing cycles per minute) with the first 30-second segment ($0 \sim 30$ seconds), compared to RGD and CounteRGAN. This observation shows that CounTS is capable of capturing global temporal pattern (the breathing frequency of the individual) and keeping such exogenous variables unchanged in its explanation thanks to the global exogenous variable $\mathbf{u}_g$ in our model.

6. Conclusion

In this paper, we identified the actionability and feasibility requirements for time series models counterfactual explanations and proposed the first self-interpretable time series prediction model, CounTS, that satisfies both requirements. Our theoretical analysis shows that CounTS is guaranteed to identify the causal effect between time series input and output in the presence of confounding variables, thereby generating counterfactual explanations in the causal sense. Empirical results showed that our method achieves competitive prediction accuracy on time series data and is capable of generating more actionable and feasible counterfactual explanations. Interesting future work includes further reducing the computation complexity during counterfactual inference and extending our method to multimodal data.

Acknowledgement

We would like to thank the reviewers/AC for the constructive comments to improve the paper. JY and HW are partially supported by NSF Grant IIS-2127918. We also thank National Sleep Research Resource (NSRR) data repository and the researchers for sharing SHHS (Zhang et al., 2018b; Quan et al., 1998), MESA (Zhang et al., 2018b; Chen et al., 2014) and SOF (Zhang et al., 2018b; Spira et al., 2008).

References

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pp. 883–892. PMLR, 2018.

Chen, X., Wang, R., Zee, P., Lutsey, P., Javaheri, S., Alcántara, C., Jackson, C., Williams, M., and Redline, S. Racial/ethnic differences in sleep disturbances: The multi-ethnic study of atherosclerosis (mesa). *Sleep*, 38, 11 2014. doi: 10.5665/sleep.4732.

Cummings, S. R., Black, D. M., Nevitt, M. C., Browner, W. S., Cauley, J. A., Genant, H. K., Mascioli, S. R., Scott, J. C., Seeley, D. G., Steiger, P., et al. Appendicular bone density and age predict hip fracture in women. *JAMA*, 263(5):665–668, 1990.

Dhaou, A., Bertoncello, A., Gourvénec, S., Garnier, J., and Le Pennec, E. Causal and interpretable rules for time series analysis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2764–2772, 2021.

Ding, H., Ma, Y., Deoras, A., Wang, Y., and Wang, H. Zero-shot recommender systems. *arXiv preprint arXiv:2105.08318*, 2021.

Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., and Lee, S. Counterfactual visual explanations. In *International Conference on Machine Learning*, pp. 2376–2384. PMLR, 2019.

Heo, J., Lee, H. B., Kim, S., Lee, J., Kim, K. J., Yang, E., and Hwang, S. J. Uncertainty-aware attention for reliable interpretation and prediction. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

Lin, W., Lan, H., Wang, H., and Li, B. Orphicx: A causality-inspired latent variable model for interpreting graph neural networks. In *CVPR*, 2022.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017b.

Natesan Ramamurthy, K., Vinzamuri, B., Zhang, Y., and Dhurandhar, A. Model agnostic multilevel explanations. *Advances in neural information processing systems*, 33: 5968–5979, 2020.

Nemirovsky, D., Thiebaut, N., Xu, Y., and Gupta, A. Countergan: Generating counterfactuals for real-time recourse and interpretability using residual gans. In Cussens, J. and Zhang, K. (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 1488–1497. PMLR, 01–05 Aug 2022.

Pan, Q., Hu, W., and Chen, N. Two birds with one stone: Series saliency for accurate and interpretable multivariate time series forecasting. In *IJCAI*, pp. 2884–2891, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Pawlowski, N., Coelho de Castro, D., and Glocker, B. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020.

Pearl, J. *Causality*. Cambridge University Press, 2 edition, 2009.

Plumb, G., Molitor, D., and Talwalkar, A. S. Model agnostic supervised local explanations. *Advances in neural information processing systems*, 31, 2018.

Plumb, G., Al-Shedivat, M., Cabrera, Á. A., Perer, A., Xing, E., and Talwalkar, A. Regularizing black-box models for improved interpretability. *Advances in Neural Information Processing Systems*, 33:10526–10536, 2020.

Quan, S., Howard, B., Iber, C., Kiley, J., Nieto, F., O'Connor, G., Rapoport, D., Redline, S., Robbins, J., Samet, J., and Wahl, The sleep heart health study: Design, rationale, and methods. *Sleep*, 20:1077–85, 01 1998. doi: 10.1093/sleep/20.12.1077.

Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O'connor, G. T., Rapoport, D. M., Redline, S., Robbins, J., Samet, J. M., et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.

Rajapaksha, D. and Bergmeir, C. Limref: Local interpretable model agnostic rule-based explanations for forecasting, with an application to electricity smart meter data. *arXiv preprint arXiv:2202.07766*, 2022.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Spira, A. P., Blackwell, T., Stone, K. L., Redline, S., Cauley, J. A., Ancoli-Israel, S., and Yaffe, K. Sleep-disordered breathing and cognition in older women. *Journal of the American Geriatrics Society*, 56(1):45–50, 2008.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Tonekaboni, S., Joshi, S., Campbell, K., Duvenaud, D. K., and Goldenberg, A. What went wrong and when? instance-wise feature importance for time-series black-box models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 799–809. Curran Associates, Inc., 2020.

Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017a.

Wachter, S., Mittelstadt, B. D., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Cybersecurity*, 2017b.

Wang, H. *Bayesian Deep Learning for Integrated Intelligence: Bridging the Gap between Perception and Inference*. PhD thesis, Hong Kong University of Science and Technology, 2017.

Wang, H. and Yeung, D.-Y. Towards bayesian deep learning: A framework and some existing methods. *TDKE*, 28(12): 3395–3408, 2016.

Wang, H. and Yeung, D.-Y. A survey on bayesian deep learning. *CSUR*, 53(5):1–37, 2020.

Wang, H., Wang, N., and Yeung, D. Collaborative deep learning for recommender systems. In *KDD*, pp. 1235–1244, 2015.

Wang, S., Zhou, T., and Bilmes, J. Bias also matters: Bias attribution for deep neural network explanation. In *International Conference on Machine Learning*, pp. 6659–6667. PMLR, 2019.

Weinberger, E., Janizek, J., and Lee, S.-I. Learning deep attribution priors based on prior knowledge. *Advances in Neural Information Processing Systems*, 33:14034–14045, 2020.

Zhang, G.-Q., Cui, L., Mueller, R., Tao, S., Kim, M., Rueschman, M., Mariani, S., Mobley, D., and Redline, S. The national sleep research resource: towards a sleep data commons. *JAMA*, 25(10):1351–1358, 2018a.

Zhang, G.-Q., Cui, L., Mueller, R., Tao, S., Kim, M., Rueschman, M., Mariani, S., Mobley, D., and Redline, S. The national sleep research resource: Towards a sleep data commons. *Journal of the American Medical Informatics Association*, pp. 572–572, 08 2018b. doi: 10.1145/3233547.3233725.

Supplementary Material

A. Evidence Lower Bound

We can derive the evidence lower bound (ELBO) in Eqn. 4 as follows:

$$
\begin{aligned}
&\log p(\mathbf{y}|\mathbf{x}) \\
&= \log \int_{\mathbf{z}} \int_{\mathbf{u}} p(\mathbf{y}, \mathbf{z}, \mathbf{u}|\mathbf{x}) d\mathbf{u} d\mathbf{z} \\
&\geq \mathbb{E}_{q_\phi(\mathbf{u},\mathbf{z},\mathbf{y}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{y}, \mathbf{u}, \mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{u}, \mathbf{z}, \mathbf{y}|\mathbf{x})} \right] \\
&= \mathbb{E}_{q_\phi(\mathbf{u},\mathbf{z},\mathbf{y}|\mathbf{x})} \left[\log \frac{p(\mathbf{u}) p_\theta(\mathbf{z}|\mathbf{u}, \mathbf{x}) p_\theta(\mathbf{y}|\mathbf{u}, \mathbf{z})}{q_\phi(\mathbf{u}, \mathbf{z}, \mathbf{y}|\mathbf{x})} \right] \\
&= \mathbb{E}_{q_\phi(\mathbf{y},\mathbf{u}_l,\mathbf{u}_g,\mathbf{z}|\mathbf{x})} [p_\theta(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})] - \mathbb{E}_{q_\phi(\mathbf{y},\mathbf{u}_l,\mathbf{u}_g,\mathbf{z}|\mathbf{x})} [q_\phi(\mathbf{y}, \mathbf{u}_l, \mathbf{u}_g, \mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi(\mathbf{u},\mathbf{z},\mathbf{y}|\mathbf{x})} [\log p(\mathbf{u})] + \mathbb{E}_{q_\phi(\mathbf{u},\mathbf{z},\mathbf{y}|\mathbf{x})} [\log p_\theta(\mathbf{z}|\mathbf{u}, \mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{u},\mathbf{z},\mathbf{y}|\mathbf{x})} [\log p_\theta(\mathbf{y}|\mathbf{u}, \mathbf{z})] - \mathbb{E}_{q_\phi(\mathbf{u},\mathbf{z},\mathbf{y}|\mathbf{x})} [\log q_\phi(\mathbf{u}, \mathbf{z}, \mathbf{y}|\mathbf{x})]
\end{aligned}
$$

B. More Details on Counterfactual Inference

B.1. Proof on Identifiability

Theorem B.1 (Identifiability). *Given the posterior distribution of exogenous variable $p(\mathbf{u}_l, \mathbf{u}_g|\mathbf{x}, \mathbf{y})$, the effect of action $p(\mathbf{y} = \mathbf{y}^{cf}|do(\mathbf{x} = \mathbf{x}'), \mathbf{u}_l, \mathbf{u}_g)$ can be identified using $\mathbb{E}_{p(\mathbf{z}|\mathbf{x}',\mathbf{u}_l,\mathbf{u}_g)} p\left(\mathbf{y}^{cf}|\mathbf{z}, \mathbf{u}_l, \mathbf{u}_g\right)$.*

Proof. With $\mathbf{u} = (\mathbf{u}_l, \mathbf{u}_g)$ and applying Rule 2 and 3 in do-calculus (Pearl, 2009), we have

$$
\begin{aligned}
&p\left(\mathbf{y}^{cf}|do\left(\mathbf{x} = \mathbf{x}'\right), \mathbf{u}\right) \\
&= \int_{\mathbf{z}} p\left(\mathbf{y}^{cf}|\mathbf{z}, do\left(\mathbf{x}'\right), \mathbf{u}\right) p\left(\mathbf{z}|do\left(\mathbf{x}'\right), \mathbf{u}\right) d\mathbf{z} \\
&= \mathbb{E}_{p(\mathbf{z}|do(\mathbf{x}'),\mathbf{u})} \left[p\left(\mathbf{y}^{cf}|\mathbf{z}, do\left(\mathbf{x}'\right), \mathbf{u}\right)\right] \\
&\overset{\text{Rule2}}{=} \mathbb{E}_{p(\mathbf{z}|\mathbf{x}',\mathbf{u})} \left[p\left(\mathbf{y}^{cf}|do(\mathbf{z}), do\left(\mathbf{x}'\right), \mathbf{u}\right)\right] \\
&\overset{\text{Rule3}}{=} \mathbb{E}_{p(\mathbf{z}|\mathbf{x}',\mathbf{u})} \left[p\left(\mathbf{y}^{cf}|do(\mathbf{z}), \mathbf{u}\right)\right] \\
&\overset{\text{Rule2}}{=} \mathbb{E}_{p(\mathbf{z}|\mathbf{x}',\mathbf{u})} \left[p\left(\mathbf{y}^{cf}|\mathbf{z}, \mathbf{u}\right)\right],
\end{aligned} \tag{14}
$$

concluding the proof. □

B.2. Counterfactual Explanation Algorithm

The pseudo-code for counterfactual explanation is shown in Algorithm 1.

C. More Details on Experiments

C.1. Details on Datasets

***Spike* Dataset.** The generation process of the *Spike* dataset is summarized below:

We generate $D = 3$ independent channels of non–linear auto-regressive moving average (NARMA) time series data using the following formula:

$$
x_{d,t+1} = 0.5 x_{d,t} + 0.5 x_{d,t} \sum_{i=0}^{l-1} x(t-l) + 1.5 u(t-(l-1)) u(t) + 0.5 + \alpha_d t \tag{15}
$$

for $t = [1, \ldots, 80]$, order $l = 2$, $u \sim \mathcal{N}(0, 0.03)$, and α_d is set differently for each channel ($\alpha_1 = 0.1$, $\alpha_2 = 0.065$ and $\alpha_3 = 0.003$). We use d to index the 3 channels.

Algorithm 1 Generating Counterfactual Explanations

Input: data $\mathbf{x}$, threshold ϵ, number of samples n and m.
Sample $\mathbf{y}^{pred}$ from $q_\phi(\mathbf{y}|\mathbf{x})$.
Set a target counterfactual output $\mathbf{y}^{cf} \neq \mathbf{y}^{pred}$
for $i = 1$ **to** m **do**
  Sample $\mathbf{u}_i$ from $q_\phi(\mathbf{u}|\mathbf{x}, \mathbf{y})$
  for $j = 1$ **to** n **do**
    Sample $\mathbf{z}_{ij} \sim p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{u}_i)$
    Sample $\mathbf{y}_{ij} \sim p_\theta(\mathbf{y}|\mathbf{z}_{ij}, \mathbf{u}_i)$
  end for
end for
Calculate $\mathbf{y}^{cf} = \frac{\sum_i \sum_j \mathbf{y}_{ij}}{n \times m}$
Set $\mathbf{x}^{cf} = \mathbf{x}$
while $|\mathbf{y} - \mathbf{y}^{cf}| \geq \epsilon$ **do**
  Update $\mathbf{x}^{cf} = \mathbf{x}^{cf} - \lambda \frac{\partial |\mathbf{y} - \mathbf{y}^{cf}|}{\partial \mathbf{x}^{cf}}$
end while
return $\mathbf{x}^{cf}$

$$
\begin{aligned}
&\boldsymbol{\theta} = [0.8, 0.4, 0]; \\
&n_d \sim \text{Bernoulli}(\theta_d); \\
&m_d \sim \text{Bernoulli}(\theta_d); \\
&\eta_d = \begin{cases} \text{Poisson}(\lambda = 2) & \mathbf{1} \text{ if } (n_d == 1) \\ 0 & \text{otherwise} \end{cases} \\
&\mathbf{g}_d = \text{Sample}\left([T], \eta_d\right) \\
&x_{d,t} = x_{d,t} + (\kappa_{d,t} + \theta_d) \quad \text{where } \kappa_{d,t} \sim \mathcal{N}(1, 0.3) \quad \forall t \in \mathbf{g}_d \\
&y_t = \begin{cases} 0 & t \leqslant \min\left(\mathbf{g}_d\right) \text{ where } m_d = 1 \\ 1 & \text{otherwise} \end{cases}
\end{aligned}
\tag{16}
$$

Real-World Datasets. The *real-world* datasets also include additional patient information such as age, gender and race. The age range is $[44, 90]$ for SHHS, $[54, 95]$ for MESA and $[71, 90]$ for SOF. In total, there are $2{,}651$, $2{,}055$ and 453 patients in SHHS, MESA, and SOF, respectively, with an average of $1{,}043$ breathing signal segments (approximately 8.7 hours).

C.2. Details on Baselines

We use the following five types of state-of-the-art baselines:

- Gradient-based methods. Regularized Gradient Descent (**RGD**) (Wachter et al., 2017a) directly models $p(y|x)$ and provide the explanation by modifying input with gradients along with L1 regularization; it is therefore it is an actionable explanation method. Gradient-weighted Class Activation Mapping (**GradCAM**) (Selvaraju et al., 2017) is originally designed to models with convolutional layers. We added convolutional layers to our model to adapt GradCAM for our time series data. Gradient SHapley Additive exPlanations (**GradSHAP**) (Lundberg & Lee, 2017a) is a game theoretic approach that uses the expectation of gradients to approximate the SHAP values.
- Perturbation-based method. Local Interpretable Model-agnostic Explanations (**LIME**) (Ribeiro et al., 2016) is a local interpretation approach based on the local linearity assumption and provides explanations by fitting the output of the model given locally perturbed input.
- Distribution shift method. Feature Importance in Time (**FIT**) (Tonekaboni et al., 2020) is a time-series back-box model explanation method that evaluates the importance of the input data based on the temporal distribution shift and unexplained distribution shift.
- Association rule method. Case-crossover APriori (**CAP**) (Dhaou et al., 2021) applies association rule mining algorithm, Apriori, to explore the causal relationship in time-series data.

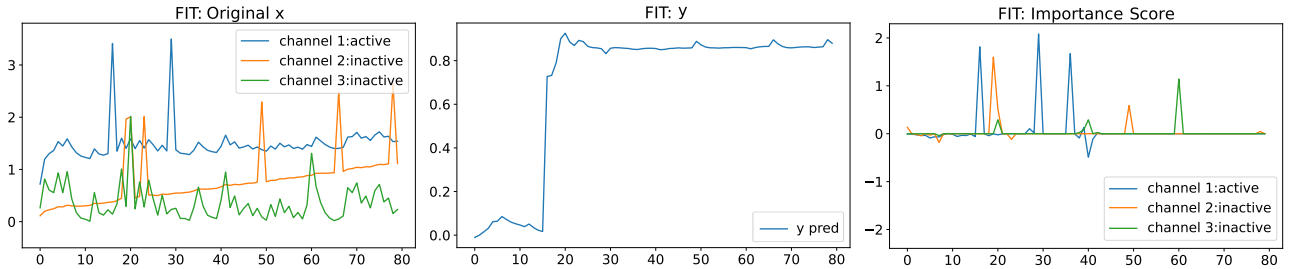

Figure 5. Qualitative results from FIT in the *Spike* dataset.

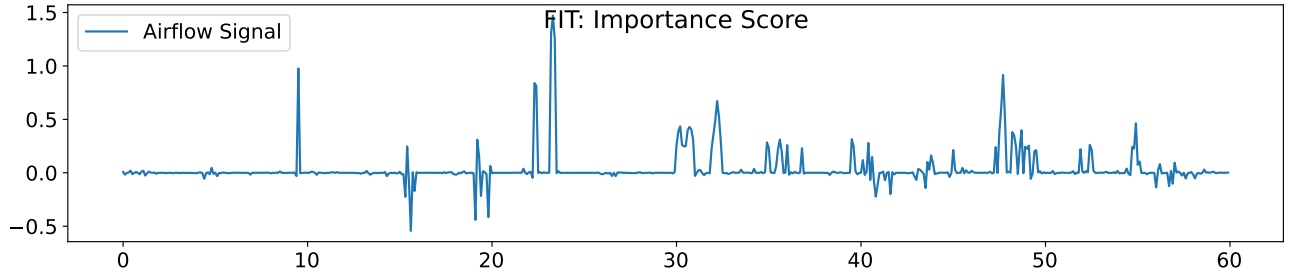

Figure 6. Qualitative results on FIT in a *MESA* intra-dataset setting.

- Generative method. Counterfactual Residual Generative Adversarial Network (**CounteRGAN**) (Nemirovsky et al., 2022) combines a Residual GAN (RGAN) and a classifier to generate counterfactual output. The generated residual output is considered as the result of the *do* operation. CounteRGAN is an actionable explanation method.

D. Additional Results

Results on the *Spike* Dataset. Note that methods such as FIT are *not* an actionable explanation method. It cannot provide counterfactual explanation $\mathbf{x}^{cf}$ that could shift the model prediction from $\mathbf{y}^{pred}$ to $\mathbf{y}^{cf}$; it can only provide importance scores to explain the current prediction $\mathbf{y}^{pred}$. Fig. 5 shows the importance scores produced by FIT to explain the same model in Fig. 3 in the *Spike* dataset.

Results on Real-World Datasets. As FIT cannot provide actionable explanations, the importance scores produced by FIT to explain the same model in Fig. 4 in *real-world dataset* are separately shown in Fig. 6.

Table 7∼11 show more detailed CCR results with different counterfactual action settings and dataset settings. Table 12 shows detailed counterfactual accuracy for each dataset settings and each counterfactual action settings.

Table 7. CCR for different counterfactual action settings (e.g., A→D means 'Awake'→'Deep Sleep') in *SOF*.

	A→D	L→D	R→D	R→A	D→A	L→A	Average
CounteRGAN	**1.356**	1.138	1.165	1.127	0.908	1.060	1.126
RGD	0.996	1.103	1.174	0.951	1.108	1.194	1.088
GradCAM	1.012	0.926	0.644	0.727	1.012	1.003	0.887
GradSHAP	0.843	1.177	1.219	0.913	0.843	0.981	0.996
LIME	1.220	0.857	1.095	0.936	1.220	1.139	1.078
FIT	1.064	1.150	1.169	1.202	1.064	0.667	1.053
CAP	0.989	0.952	1.019	1.139	0.989	0.689	0.963
CounTS (Ours)	1.179	**1.161**	**1.240**	**1.475**	**1.245**	**1.387**	**1.281**

Table 8. CCR for different counterfactual action settings in *SHHS+SOF→ MESA*.

	A→D	L→D	R→D	R→A	D→A	L→A	Average
CounteRGAN	1.381	**1.379**	1.068	1.373	1.355	1.158	1.285
RGD	1.226	1.130	1.089	1.215	1.177	1.293	1.188
GradCAM	1.056	1.014	0.743	1.218	1.056	1.101	1.031
GradSHAP	1.174	0.822	1.046	0.948	1.174	0.988	1.025
LIME	0.904	1.246	1.138	1.017	0.904	0.836	1.007
FIT	1.143	1.228	1.193	1.206	1.143	0.938	1.142
CAP	0.969	1.179	0.947	1.418	0.969	0.703	1.031
CounTS (Ours)	**1.391**	1.331	**1.253**	**1.470**	**1.465**	**1.355**	**1.377**

Table 9. CCR for different counterfactual action settings in *MESA*.

	A→D	L→D	R→D	R→A	D→A	L→A	Average
CounteRGAN	1.183	**1.345**	1.160	1.117	0.974	1.200	1.163
RGD	1.159	1.088	1.149	1.204	1.190	1.207	1.166
GradCAM	1.078	0.790	0.419	0.954	1.078	0.977	0.883
GradSHAP	0.885	1.027	1.039	1.273	0.885	1.145	1.042
LIME	0.903	1.003	0.930	1.233	0.903	0.952	0.987
FIT	1.210	1.118	1.024	1.082	1.210	0.887	1.089
CAP	0.786	1.001	1.176	1.243	0.786	0.783	0.962
CounTS (Ours)	**1.213**	1.309	**1.294**	**1.526**	**1.350**	**1.285**	**1.329**

Table 10. CCR for different counterfactual action settings in *MESA+SOF→SHHS*.

	A→D	L→D	R→D	R→A	D→A	L→A	Average
CounteRGAN	**1.317**	1.157	**1.281**	1.300	1.270	1.060	1.231
RGD	1.193	1.266	1.115	1.313	1.304	1.255	1.241
GradCAM	1.131	1.093	0.725	1.306	1.131	0.953	1.056
GradSHAP	1.025	0.910	1.092	0.886	1.025	0.910	0.975
LIME	1.018	1.092	1.118	0.944	1.018	0.930	1.020
FIT	1.270	1.259	1.105	1.263	1.270	0.991	1.193
CAP	0.994	0.819	1.033	0.953	0.994	0.777	0.929
CounTS (Ours)	1.301	**1.452**	1.167	**1.560**	**1.417**	**1.443**	**1.390**

Table 11. CCR for different counterfactual action settings in *SHHS+MESA→SOF*.

	A→D	L→D	R→D	R→A	D→A	L→A	Average
CounteRGAN	**1.210**	1.050	1.034	**1.403**	1.065	1.248	1.153
RGD	1.183	0.993	1.015	1.260	1.155	1.201	1.135
GradCAM	1.108	1.018	1.020	1.215	1.108	1.286	1.126
GradSHAP	0.983	0.929	0.780	1.044	0.983	0.728	0.908
LIME	0.738	1.051	1.199	1.155	0.738	0.708	0.932
FIT	1.110	1.175	0.954	1.206	1.110	0.876	1.072
CAP	1.106	0.951	0.833	1.145	1.106	1.060	1.034
CounTS (Ours)	1.203	**1.266**	**1.202**	1.395	**1.472**	**1.341**	**1.313**

Table 12. Couterfactual accuracy for each dataset settings and each counterfactual action settings.

	Method	A→D	L→D	R→D	R→A	D→A	L→A	Average
SHHS+MESA ↓ SOF	RGD	0.867	0.903	0.921	**0.932**	**0.931**	**0.922**	0.913
	CounteRGAN	0.848	**0.917**	0.903	0.917	0.896	0.900	0.897
	FIT	0.285	0.296	0.328	0.384	0.285	0.348	0.331
	CounTS (Ours)	**0.868**	0.893	**0.929**	0.918	0.920	0.913	**0.916**
MESA+SOF ↓ SHHS	RGD	0.828	**0.882**	0.916	**0.910**	**0.903**	**0.890**	**0.887**
	CounteRGAN	0.828	0.869	0.920	0.867	0.880	0.875	0.868
	FIT	0.334	0.263	0.374	0.354	0.334	0.348	0.327
	CounTS (Ours)	**0.837**	0.878	**0.923**	0.899	0.899	**0.890**	**0.887**
SHHS+SOF ↓ MESA	RGD	**0.882**	**0.898**	0.933	**0.910**	**0.938**	0.888	**0.907**
	CounteRGAN	0.872	0.878	0.926	0.905	0.901	**0.897**	0.880
	FIT	0.335	0.269	0.301	0.357	0.335	0.397	0.346
	CounTS (Ours)	**0.882**	0.888	**0.936**	**0.910**	0.930	0.881	0.889
SOF	RGD	0.823	0.820	**0.919**	0.856	0.861	**0.891**	**0.862**
	CounteRGAN	**0.831**	0.800	0.899	**0.867**	0.832	0.875	0.844
	FIT	0.257	0.328	0.311	0.328	0.257	0.310	0.248
	CounTS (Ours)	0.813	**0.831**	**0.907**	0.843	**0.864**	0.881	0.852
SHHS	RGD	**0.880**	**0.892**	**0.940**	**0.944**	0.887	**0.900**	**0.907**
	CounteRGAN	0.841	0.854	0.916	0.913	**0.888**	0.875	0.888
	FIT	0.335	0.272	0.353	0.337	0.335	0.348	0.322
	CounTS (Ours)	0.868	0.880	0.930	**0.931**	0.875	0.889	0.903
MESA	RGD	**0.846**	**0.888**	0.912	**0.907**	**0.909**	0.888	**0.892**
	CounteRGAN	0.828	0.869	**0.920**	0.867	0.880	0.875	0.868
	FIT	0.290	0.316	0.320	0.307	0.290	0.301	0.294
	CounTS (Ours)	0.837	0.878	0.903	0.899	0.899	**0.890**	0.887