Explain Temporal Black-Box Models via Functional Decomposition

Linxiao Yang [1] **Yunze Tong** [1 2] **Xinyue Gu** [1] **Liang Sun** [1]

Abstract

How to explain temporal models is a significant challenge due to the inherent characteristics of time series data, notably the strong temporal dependencies and interactions between observations. Unlike ordinary tabular data, data at different time steps in time series usually interact dynamically, forming influential patterns that shape the model's predictions, rather than only acting in isolation. Existing explanatory approaches for time series often overlook these crucial temporal interactions by treating time steps as separate entities, leading to a superficial understanding of model behavior. To address this challenge, we introduce FDTempExplainer, an innovative model-agnostic explanation method based on functional decomposition, tailored to unravel the complex interplay within black-box time series models. Our approach disentangles the individual contributions from each time step, as well as the aggregated influence of their interactions, in a rigorous framework. FDTempExplainer accurately measures the strength of interactions, yielding insights that surpass those from baseline models. We demonstrate the effectiveness of our approach in a wide range of time series applications, including anomaly detection, classification, and forecasting, showing its superior performance to the state-of-the-art algorithms.

1. Introduction

Explanatory methods hold paramount significance across diverse realms of scientific research, where the need to understand the predictions of complex models is essential for both validation and application (Tjoa & Guan, 2020; Yang et al., 2023; Amann et al., 2020). Time series data, by virtue of its omnipresence, capture the dynamic evolution of variables across a spectrum of fields including economics (Kendall & Hill, 1953), meteorology (Campbell & Diebold, 2005), and public health (Zeger et al., 2006). However, the investigation into explaining the predictions of time series models—a critical piece in the puzzle of model transparency—has not kept pace with the demands of researchers and practitioners. The ability to explain time series model predictions is crucial; it not only demystifies the decision-making process (Koo et al., 2016) but also instills trust and fosters practical insights for stakeholders (Ivaturi et al., 2021; Mobley, 2002; Ismail Fawaz et al., 2019).

The unique characters of time series data pose distinct challenges when it comes to elucidating different model predictions. Firstly, the high degree of temporal dependency (including the commonly used autocorrelation) inherent in time series data complicates the task of identifying the influence of specific inputs on the model outputs. In particular, many of existing explanatory methods (e.g., perturbation-based methods) rely on generating artificial samples for explanation, which do not follow the underlying complex temporal dependency relationship in the time series data, leading to the out-of-distribution (OOD) problem. Secondly, events occurring at a given time point can have enduring effects on subsequent observations, encapsulating phase shifts that ripple through future data points. In fact, the information of a phase change at one time step is embedded in all the following steps, and the resulting accumulation effect of events make accurate explaining challenging. Thirdly, we need to consider not only a single time point (e.g., points corresponding to events), but also time series subsequence in explanation. As opposed to the impact of isolated observations as events, some subsequence plays important role in time series tasks. Take subsequence anomaly detection as an example, anomalies are determined based on the collective behavior of a time series subsequence rather than a singular time point (Boniol & Palpanas, 2020). Similarly, in forecasting, successive observations often compose periodic motifs or exhibit slowly evolving trends that have a profound influence on subsequent values within the series.

Although various general explanatory methods, such as LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2017), integrated gradient (Sundararajan et al., 2017), and

[1]DAMO Academy, Alibaba Group, Hangzhou, China [2]Department of Computer Science and Technology, Zhejiang University, Hangzhou, China. Correspondence to: Linxiao Yang <linxiao.ylx@alibaba-inc.com>, Liang Sun <liang.sun@alibaba-inc.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

DeepLIFT (Shrikumar et al., 2017), have been proposed in the literature (Ribeiro et al., 2016; Lundberg & Lee, 2017; Sundararajan et al., 2017; Shrikumar et al., 2017), the inherent complexities of time series data make these traditional methods not fully applicable (Ismail et al., 2020). In particular, many of them suffer from the OOD problem, and often ignore the effects of preceding time steps on subsequent observations. Recently, some methods have been proposed and mitigate the OOD problem by learning counterfactual samples (Tonekaboni et al., 2020; Leung et al., 2023; Meng et al., 2023; Enguehard, 2023; Sivill & Flach, 2022; Tonekaboni et al., 2020). To account for the effects of time steps on later observations, (Leung et al., 2023) proposes an approach that aggregates the impact of a time step across a subsequent time window. In an effort to neutralize the confounding effects of prior time steps, (Suresh et al., 2017; Tonekaboni et al., 2020) quantify the significance of a time step by calculating the change in the predictive power of the model before and after the observation of that specific time step. While these recent advances have led to explanatory techniques that consider temporal dependencies, there still remains a gap in adequately addressing the intricate interactions between observational points. To the best of our knowledge, there is no algorithm which can deal with three aforementioned challenges together.

Next we present a simplified toy example to illustrate the significance of interactions when explaining the predictions of temporal models. Fig. 1 showcases a comparison between two anomaly detection methods based on how they account for interactions across time steps. The input time series $\boldsymbol{x}$ exhibits two distinct spikes at time steps t_1 and t_2. The function $f_t^{(1)}(\boldsymbol{x})$ signals an anomaly (value changes to 1) if and only if a spike is present at the current time step t. Conversely, $f_t^{(2)}(\boldsymbol{x})$ reports an anomaly (value changes to 1) if a spike occurred in the past, up to and including the current time step. It is evident that $f_t^{(1)}(\boldsymbol{x})$ neglects the potential interplay between time steps within the series $\boldsymbol{x}$. In contrast, $f_t^{(2)}(\boldsymbol{x})$ incorporates the influence of all preceding time steps, reflecting their cumulative interactions. To simplify the discussion and further delve into analyzing the interaction effect between spikes at t_1 and t_2, we assume $t_1 < t_2 \leq t$ and there is no other spikes. We define E_1 as the occurrence of a spike at t_1, and E_2 as the occurrence of a spike at t_2. Formally, we reformulate $f_t^{(2)}(\boldsymbol{x})$ as

$$f_t^{(2)}(\boldsymbol{x}) = E_1 \text{ or } E_2 = \mathcal{I}(E_1) + \mathcal{I}(E_2) - \mathcal{I}(E_1)\mathcal{I}(E_2),$$

where $\mathcal{I}(E)$ denotes an indicator function that equals 1 if event E occurs and 0 otherwise. This decomposition clarifies how both spikes at t_1 and t_2 influence the value of $f_t^{(2)}(\boldsymbol{x})$. Notably, the interaction between spikes at t_1 and t_2 (described as $\mathcal{I}(E_1)\mathcal{I}(E_2)$) exerts a diminishing effect on the value of $f_t^{(2)}(\boldsymbol{x})$. When we consider the update at time step t_2, it becomes apparent that the effect of t_2 is not isolated; rather, its interrelation with t_1 also contributes to the outcome of $f_t^{(2)}(\boldsymbol{x})$. If we fail to correctly disentangle this interaction and merely compare the value of $f_t^{(2)}(\boldsymbol{x})$ before and after the spike at t_2, we might erroneously conclude that t_2 has no impact. This oversight occurs because the interactive effect between t_1 and t_2 neutralizes the primary influence of the spike at t_2.

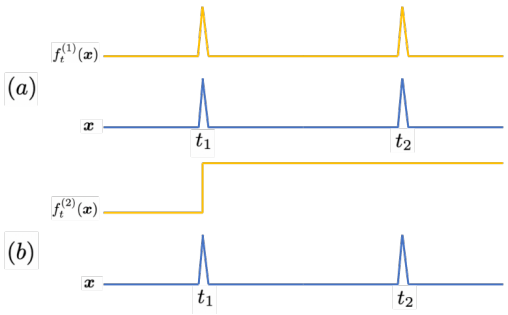

Figure 1: Example of two anomaly detection functions with the same input. (a) and (b) show the response of $f_t^{(1)}(\boldsymbol{x})$ and $f_t^{(2)}(\boldsymbol{x})$ given input $\boldsymbol{x}$, respectively. While both functions process the same input, interaction between the two spikes exists in $f_t^{(2)}(\boldsymbol{x})$, while absents in $f_t^{(1)}(\boldsymbol{x})$.

To accurately quantify the interactions, we introduce a method that decomposes black-box models into a sum of individual time step effects and all potential interactions between them. Recognizing the temporal dependencies inherent in time series data, we posit that interactions occur exclusively between consecutive time steps. Furthermore, to mitigate the bias introduced by preceding sequences, we introduce the notion of "pure" interaction. Specifically, an interaction is considered pure if it is unaffected by prior time steps. Intuitively, if a time step's value can be exclusively predicted by preceding data, then its interactions with this preceding sequence are zero. We then demonstrate that, given a decomposition where all interactions are pure, these interactions can be determined analytically through closed-form solutions. Once the interactions are established, we introduce a model-agnostic explainer FDTempExplainer for black-box time series models. This technique equitably distributes the effect of interactions across involved time steps, ensuring each time step is assigned its due influence. As the temporal interaction is fully considered, FDTempExplainer takes account of past events and subsequence in explanation. It enjoys the solid mathematical foundation, and can be applied to major time series tasks, including time series anomaly detection, classification, and forecasting. Our empirical studies demonstrate that our method not only accurately calculates the interactions but also provides more valid significance scores to individual time steps when compared to current state-of-the-art methods.

2. Preliminaries

Let $f(\boldsymbol{X}, \boldsymbol{C})$ be a temporal black-box model, which takes multi-variate time series $\boldsymbol{X}$ and $\boldsymbol{C}$ as input, and outputs an L-dimensional vector $\boldsymbol{y} \in \mathbb{R}^L$. Here $\boldsymbol{X} \in \mathbb{R}^{D \times T}$ is a multi-variate time series with D features and T time steps, and $\boldsymbol{C}$ denotes external context variables which are not focused in this paper. Here we do not assume the sequential output of the model. We denote $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T]$, where $\boldsymbol{x}_t \in \mathbb{R}^d$ is the t-th column of $\boldsymbol{X}$. We define $\boldsymbol{X}_{t_1:t_2} = [\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_2}]$ as a segmentation of $\boldsymbol{X}$ within time interval $t_1 \leq t \leq t_2$, i.e., the submatrix of $\boldsymbol{X}$ corresponds to the time range $[t_1, t_2]$. In our framework, we treat $\boldsymbol{X}$ and $\boldsymbol{C}$ as random variables, and denote their specific realization as $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{C}}$, respectively.

3. Proposed Method

In this section, we present our functional decomposition-based method designed to explain temporal models. Formally, with a given input $\bar{\boldsymbol{X}}$ and a context variable $\bar{\boldsymbol{C}}$, our goal is to determine the impact of features within each time step on the model's output. Specifically, we aim to measure the contribution of $\bar{\boldsymbol{x}}_t$, denoted as ϕ_t, for each time step where $1 \leq t \leq T$. The contributions of $\bar{\boldsymbol{x}}_t$ stem from multiple factors. The inherent value of $\bar{\boldsymbol{x}}_t$ independently influences the output. Moreover, the features at time step t, $\bar{\boldsymbol{x}}_t$, interact with features at neighboring time steps, such as those at $t-1$ and $t+1$, creating temporal patterns that jointly impact the output. To accurately assess these various types of impact, we reframe the function $f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{C}})$ into an additive formulation as follows:

$$
\begin{aligned}
& f(\boldsymbol{X}, \bar{\boldsymbol{C}}) \\
=& C_0 + \sum_{t=1}^{T} m(\boldsymbol{x}_t) + \sum_{t=1}^{T-1} I^2(\boldsymbol{X}_{t:t+1}) + \sum_{t=1}^{T-2} I^3(\boldsymbol{X}_{t:t+2}) \\
& + \cdots + \sum_{t=1}^{2} I^{T-1}(\boldsymbol{X}_{t:t+T-2}) + I^T(\boldsymbol{X}),
\end{aligned} \tag{1}
$$

where C_0 is a constant, $m(\cdot)$ and $\{I^k(\cdot)\}_{k=2}^{T}$ are functions. Eq. (1) decomposes the function $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$ into multiple terms. Specifically, the function $m(\boldsymbol{x}_t)$ models the effect on the output caused by $\boldsymbol{x}_t$ itself. Thus, it is only related to the features at time step t and irrelevant to other time steps. Similarly, function $I^{k+1}(\boldsymbol{X}_{t:t+k})$ is only related to the features from the time step t to $t+k$, and irrelevant to the rest time steps. Thus, function $I^{k+1}(\boldsymbol{X}_{t:t+k})$ computes the effect of features in segment $\boldsymbol{X}_{t:t+k}$ on the output due to incorporation of these features. Here we call $I^{k+1}(\boldsymbol{X}_{t:t+k})$ as the $k+1$th order interaction of $\boldsymbol{X}_{t:t+k}$.

Remark. The decomposition presented in Eq. (1) can be regarded as an augmentation of generalized additive models (GAMs) that incorporates higher-order interactions among consecutive time steps. It is important to note that Eq. (1) deliberately excludes interactions between non-consecutive time steps. The rationale behind this assumption is rooted in the observed smoothness of time series transitions, which implies a stronger correlation among adjacent time steps. This proximity-based correlation is often integral to the formation of patterns that significantly influence the model's outputs. Even for time series characterized by distinct cycles, there is no inherent necessity for direct interactions between non-adjacent time steps. More commonly, the cyclic behavior emerges from the aggregated influence of all time steps within the period, rather than from discrete interactions among non-consecutive steps. It is critical to clarify that "interaction" pertains to a model's behavior, not to an intrinsic property of the time series itself. As illustrated in Fig. 1, with the same input time series, observable interactions occur between time steps in $f_t^{(2)}$ but not in $f_t^{(1)}$. Consequently, scenarios wherein features widely separated in time affect the current target, or previous features influence current ones, are consistent with our assumption. We assert that this assumption is defensible in most cases. Moreover, where exceptions arise, we propose specific remedies. Should interactions between non-consecutive time steps occur, Eq. (1) would subsume them under their broader consecutive context. For instance, the influence of the interaction between $\boldsymbol{x}_{t-2}$ and $\boldsymbol{x}_t$ would be incorporated into the term $I^3(\boldsymbol{X}_{t-2:t})$, potentially causing an inflated estimation of $\boldsymbol{x}_{t-1}$'s impact. Nonetheless, these areas of potential inattention notwithstanding, Eq. (1) effectively isolates the interaction between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-2}$ from $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$. As a result, it accurately captures low-order interactions and primary effects of time steps, which predominantly determine their effects to the model's output. By limiting the interaction to contiguous time steps, we streamline the model, focusing on the proximal relationships that are most salient for time series analysis.

As previously discussed, occurrences at a given time step have the potential to influence subsequent time steps in a time series. To neutralize such effects, it is necessary to consider that, for time steps whose values are completely determined by preceding steps, their interactions with those preceding steps should be nullified. Furthermore, if the value at a particular time step is unobserved, then we would anticipate that the expected value of its interaction with other time steps should be zero since it does not contribute any new information. Formally, we introduce the "pure interaction" to describe this property.

Definition 3.1 (pure interaction)**.** Given an interaction function $I^{k+1}(\boldsymbol{X}_{t:t+k})$, we say it is pure if and only if it satisfies

$$
\int I^{k+1}(\mathcal{Z}(t_1, t_2)) p(\boldsymbol{X}_P | \bar{\boldsymbol{X}}_{t_1:t_2}, \bar{\boldsymbol{C}}) d\boldsymbol{X}_P = 0 \tag{2}
$$

for all $t \leq t_1 < t_2 \leq t+k$, where $\mathcal{Z}(t_1, t_2) =$

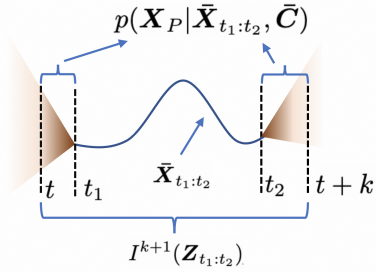

Figure 2: Illustration of Definition 3.1 (pure interaction)

$[\boldsymbol{X}_{t:t_1-1}, \bar{\boldsymbol{X}}_{t_1:t_2}, \boldsymbol{X}_{t_2+1:t+k}]$, $\boldsymbol{X}_P$ denotes features in $\boldsymbol{X}_{t:t+k}$ but not in $\boldsymbol{X}_{t_1:t_2}$.

An illustration of Definition 3.1 is shown at Fig. 2. Obviously, if $\boldsymbol{X}_P$ is fully determined by $\bar{\boldsymbol{X}}_{t_1:t_2}$, then $p(\boldsymbol{X}_P|\bar{\boldsymbol{X}}_{t_1:t_2}, \bar{\boldsymbol{C}})$ will reduce to a Dirac delta function, and eventually leads to $I^{k+1}(\bar{\boldsymbol{X}}_{t:t+k}) = 0$. Note that the left term of Eq.(2) may be a function if $\boldsymbol{X}_P$ does not cover all the features in $\boldsymbol{X}_{t:t+k}$ but not in $\boldsymbol{X}_{t_1:t_2}$.

Remark. The concept of pureness is also defined in PureGAM (Sun et al., 2022) for GAMs. In PureGAM, an interaction between features is considered to be pure if integrating it with respect to the marginal distribution of the involved features results in zero. This definition is grounded in a global perspective of feature distributions across the entire dataset. In contrast, the pureness described in Eq.(2) is designed for a sample-specific scenario. It is the prediction on the distribution conditioned upon a segmented sample $\bar{\boldsymbol{X}}$. This distinction is crucial because when we aim to interpret the model locally for the sample $\bar{\boldsymbol{X}}$, it is reasonable to focus on interactions that exhibit local properties in the vicinity of the given sample. Furthermore, as we will discuss later, another appealing feature of the pure interaction is its prevention of generating OOD samples due to its sample-centric definition.

Lemma 3.2. *For any function $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$, there exists a decomposition presented in Eq.(1) with all the interactions be pure.*

Lemma 3.2 shows that there is always a decomposition satisfying pure interactions, which enable us to define attribution method based on the pure interactions. In the following, we will show that with the pureness constraints, one can estimate all the interactions efficiently.

3.1. Estimation of Interactions

Directly estimating the interactions is difficult, as we can only get the summation of all the $T(T-1)$ interaction terms and N main effect terms, i.e., $f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{C}})$. One approach to estimate the interaction terms in Eq. (1) is to train a surrogate function which admits decomposition structure intrinsically. Nevertheless, we can only make sure the outputs of the surrogate function approximating the original one, but cannot guarantee that the interactions in the surrogate function convey the precise information of the interactions in original function. In this section, we propose a method to estimate the pure interactions without retain surrogate functions.

The difficulty of the interaction estimation arises from the fact that the output of the function naturally mixes the effects of different order interactions. Fortunately, if we restrict the interaction to be pure, we can delete some specific interactions by using the pure interaction assumption. For example, if we want to remove the effect of the interaction $I^T(\boldsymbol{X}_{1:T})$ from $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$, we can compute the expectation of $f([\boldsymbol{X}_{1:T-1}, \boldsymbol{X}_T], \bar{\boldsymbol{C}})$ with respect to $\boldsymbol{X}_T$ over the distribution $p(\boldsymbol{X}_T|\bar{\boldsymbol{X}}_{T-1})$. Following the pure interaction definition, the expectation of interactions $\{I^{T-k+1}([\boldsymbol{X}_{k:T-1}, \bar{\boldsymbol{X}}_{T-1}, \boldsymbol{X}_T])\}_{k=1}^{T-1}$ will be zero, while the rest interactions keep unchanged.

The above observation motivates us to estimate the interactions using the their pureness. Intuitively, if we are able to compute two terms, where one term is the expectation of $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$ with the effects of the interactions $\{I^{T-k+1}(\bar{\boldsymbol{X}}_{k:T})\}_{k=1}^{T-1}$ removed, and the other term removes the effects of the interactions $\{I^{T-k+1}(\bar{\boldsymbol{X}}_{k:T})\}_{k=2}^{T-1}$. Then the difference of two terms should be the effect of interaction of $I^T(\bar{\boldsymbol{X}}_{1:T})$.

In the following, we introduce our efficient method to estimate interactions. Before proceeding, we first define operator $\mathbb{E}_{t_1:t_2|t_3:t_4}(\cdot)$ as

$$\mathbb{E}_{t_1:t_2|t_3:t_4}(f(\cdot)) = \int f(\cdot) p(\boldsymbol{X}_{t_1:t_2}|\bar{\boldsymbol{X}}_{t_3:t_4}, \bar{\boldsymbol{C}}) d\boldsymbol{X}_{t_1:t_2},$$

which measures the mean output of the model given the features between t_3 and t_4 are observed. Then we have following lemma.

Lemma 3.3. *Given a temporal model $f(\boldsymbol{X}, \boldsymbol{C})$ and specific inputs $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{C}}$. Let $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$ admit decomposition as shown in Eq. (1), if $I^{t_2-t_1+1}$ is pure and $t_1 \leq t_2 - 2$, then we have*

$$I^{t_2-t_1+1}(\bar{\boldsymbol{X}}_{t_1:t_2}) = (a) - (b) - (c) + (d), \qquad (3)$$

where $(a), (b), (c)$ and (d) are defined as follows:

$$\begin{aligned}
(a) =& \mathbb{E}_{1:t_1-1|t_1:t_2-1}\left[\mathbb{E}_{t_2+1:T|t_1+1:t_2}(f(\mathcal{Z}(t_1, t_2)))\right], \\
(b) =& \mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[\mathbb{E}_{t_2+1:T|t_1+1:t_2}(f(\mathcal{Z}(t_1+1, t_2)))\right], \\
(c) =& \mathbb{E}_{1:t_1-1|t_1:t_2-1}\left[\mathbb{E}_{t_2:T|t_1+1:t_2-1}(f(\mathcal{Z}(t_1, t_2-1)))\right], \\
(d) =& \mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[\mathbb{E}_{t_2:T|t_1+1:t_2-1}(f(\mathcal{Z}(t_1+1, t_2-1)))\right],
\end{aligned}$$

and $\mathcal{Z}(a, b) = [\boldsymbol{X}_{1:a-1}, \bar{\boldsymbol{X}}_{a:b}, \boldsymbol{X}_{b+1:T}]$.

Lemma 3.3 shows that the interactions can be efficiently computed by computing the expectations with some specific distributions. Fig. 3 illustrates the underlying rational behind the Lemma 3.3. From Fig. 3 we can see that each term

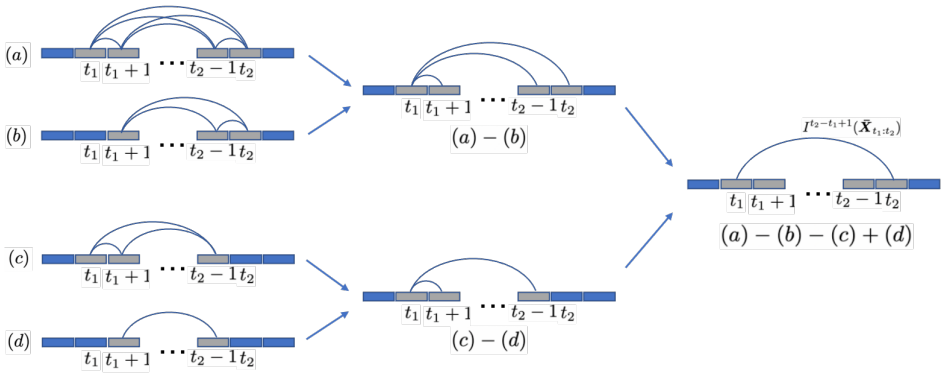

Figure 3: Illustration of our proposed interaction estimation method.

removes the effects of some interactions, and after linear combination of four terms, only the effect of desired interactions leaves. Details about which interactions are cancelled are provided in the Appendix.

Lemma 3.3 provides a method to compute the interactions when $t_1 \geq t_2 - 2$, i.e., the interactions that order large than 3. Nevertheless, for the second order interaction, i.e. $t_1 = t_2 - 1$, the terms (c) and (d) become invalid as they involve the term $\bar{\boldsymbol{X}}_{t_1+1:t_2-1}$. To address this issue, we approximate the second order interaction using following lemma.

Lemma 3.4. *Given a temporal model $f(\boldsymbol{X}, \boldsymbol{C})$ and specific inputs $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{C}}$. Let $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$ admit decomposition as shown in Eq. (1), if all interactions are pure, then we have*

$$\begin{aligned} I^2(\bar{\boldsymbol{X}}_{t:t+1}) \approx & \mathbb{E}_{1:t-1|t}\left[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}(t,t+1), \bar{\boldsymbol{C}}))\right] \\ & - \mathbb{E}_{1:t-1|t}\left[\mathbb{E}_{t+1:T|t}(f(\mathcal{Z}_t, \bar{\boldsymbol{C}}))\right] \\ & - \mathbb{E}_{1:t|t+1}\left[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}_{t+1}, \bar{\boldsymbol{C}}))\right] \\ & + \mathbb{E}_{1:t|t+1}\left[\mathbb{E}_{t+1:T|t}(f(\boldsymbol{X}, \bar{\boldsymbol{C}}))\right], \end{aligned} \tag{4}$$

where $\mathcal{Z}(a,b) = [\boldsymbol{X}_{1:a-1}, \bar{\boldsymbol{X}}_{a:b}, \boldsymbol{X}_{b+1:T}]$ and $\mathcal{Z}(a) = [\boldsymbol{X}_{1:a-1}, \bar{\boldsymbol{X}}_{a}, \boldsymbol{X}_{a+1:T}]$.

3.2. Estimation of Main Effect

Given the features in time step $\bar{\boldsymbol{x}}_t$, estimating its main effect is difficult, as $\bar{\boldsymbol{x}}_t$ is highly correlated with its neighborhood, such as $\bar{\boldsymbol{x}}_{t-1}$ and $\bar{\boldsymbol{x}}_{t+1}$. Instead of directly estimating $m(\bar{\boldsymbol{x}}_t)$, we estimate the marginal main effects of $\bar{\boldsymbol{x}}_t$, i.e., $\nu(\bar{\boldsymbol{x}}_t) = m(\bar{\boldsymbol{x}}_t) - \int m(\boldsymbol{x}_t) p(\boldsymbol{x}_t|\bar{\boldsymbol{x}}_{t-1}) d\boldsymbol{x}_t$. The marginal main effect of $\bar{\boldsymbol{x}}_t$ measures the conditional gain of $\bar{\boldsymbol{x}}_t$ given $\bar{\boldsymbol{x}}_{t-1}$, which is more suitable to quantify the amount of new information introduced by $\bar{\boldsymbol{x}}_t$.

Lemma 3.5. *Given a temporal model $f(\boldsymbol{X}, \boldsymbol{C})$ and specific inputs $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{C}}$. Let $f(\boldsymbol{X}, \bar{\boldsymbol{C}})$ admit decomposition as shown in Eq. (1) and all the interactions are pure, then we have*

$$\begin{aligned} & m(\bar{\boldsymbol{x}}_t) - \int m(\boldsymbol{x}_t) p(\boldsymbol{x}_t|\bar{\boldsymbol{x}}_{t-1}) d\boldsymbol{x}_t \\ = & \mathbb{E}_{t+1:T|t-1}(f(\mathcal{Z}_{1:t}, \bar{\boldsymbol{C}})) - \mathbb{E}_{t:T|t-1}(f(\mathcal{Z}_{1:t-1}, \bar{\boldsymbol{C}})) \\ & - \sum_{k=1}^{t-1} I^{k+1}(\bar{\boldsymbol{X}}_{t-k:t}). \end{aligned} \tag{5}$$

Lemma 3.5 provides an estimator of marginal main effect of $\bar{\boldsymbol{x}}_t$, where the interactions $\sum_{k=1}^{t-1} I^{k+1}(\bar{\boldsymbol{X}}_{t-k:t})$ can be estimated using the method introduced in Section 3.1.

3.3. Attribution Calculation

Based on the estimated interactions and main effect, we can evaluate the contribution of each time step to the function f. Specifically, to quantify the contribution of each time step, we need to define an attribution method to assign an importance score to time steps according to their main effects and interactions. We propose to assign the effect of interactions equally to the time steps involved. Each time step contributes equally to the interactions, and the importance score of $\boldsymbol{x}_t$ is the summation of its main effect and its contribution to all interactions. Formally, the feature importance in each time step is given in Eq. (6):

$$\phi_t = \nu(\bar{\boldsymbol{x}}_t) + \sum_{k=1}^{T-1} \sum_{i=t}^{\min(t+k,T)} \frac{1}{k+1} I^{k+1}(\bar{\boldsymbol{X}}_{i-k:i}). \tag{6}$$

So far, we have introduced a functionally decomposed temporal model explainer, abbreviated as FDTempExplainer, which assigns importance score to each time step. Our method does not rely on the structure of the model, making it model-agnostic. Moreover, all distributions used are based on the observation $\bar{\boldsymbol{X}}$, which helps to avoid the inclusion of out-of-distribution (OOD) samples.

3.4. Estimation of the Conditional Distribution using Generative Model

In the estimation of interactions and main effects, we need to evaluate the value of $\mathbb{E}_{t_1:t_2|t_3:t_4}(f(\cdot))$, which can be approximated as

$$\begin{aligned}\mathbb{E}_{t_1:t_2|t_3:t_4}(f(\cdot)) &= \int f(\cdot)p(\boldsymbol{X}_{t_1:t_2}|\bar{\boldsymbol{X}}_{t_3:t_4}, \bar{\boldsymbol{C}})d\boldsymbol{X}_{t_1:t_2} \\ &= \sum_{n=1}^{N} f([\boldsymbol{X}_{1:t_1-1}, \tilde{\boldsymbol{X}}^n_{t_1:t_2}, \boldsymbol{X}_{t_2+1:T}], \bar{\boldsymbol{C}}),\end{aligned}$$

where $\{\tilde{\boldsymbol{X}}^n_{t_1:t_2}\}_{n=1}^N$ are samples generated according to the distribution $p(\boldsymbol{X}_{t_1:t_2}|\bar{\boldsymbol{X}}_{t_3:t_4}, \bar{\boldsymbol{C}})$. In our model, we train a conditional variational autoencoder (CVAE) (Sohn et al., 2015) to generate samples according to $\bar{\boldsymbol{X}}_{t_3:t_4}, \bar{\boldsymbol{C}}$. Specifically, we treat the features in time step less than t_3 or larger than t_4 as missing values, and then reconstruct the overall $\bar{\boldsymbol{X}}$. To streamline the generative process, we introduce a masking technique to indicate which part of the time series is conditional on. Specifically, we first define a mask matrix $\boldsymbol{M}$, assigning $M_{t_3:t_4} = 1$ and setting the remaining elements to zero. We then feed $\boldsymbol{M} * \boldsymbol{X}$ together with $\boldsymbol{M}$ into the CVAE and train the reconstruction model for $\boldsymbol{X}$, where $*$ denotes the element-wise multiplication. Once an reconstruction is generated by CVAE, we choose the segment from t_1 to t_2 as the samples sampling from $p(\boldsymbol{X}_{t_1:t_2}|\bar{\boldsymbol{X}}_{t_3:t_4}, \bar{\boldsymbol{C}})$. We train the CVAE on the overall dataset. For the underlying Variational Autoencoder (VAE), where the latent representation is presumed to conform to a Gaussian distribution, we further impose a regularization constraint on the variance of the distribution to encourage the generation of varied samples.

3.5. Computational Complexity Analysis

Note that the computational complexity of our method is dominated by the computation of the interactions between time steps. As there are $T^2 - T$ interaction terms, the overall complexity of our method is $\mathcal{O}(T^2LN)$, where T denotes the number of time steps to explain, L denotes the output length of the black-box model, and N denotes the number of samples used to estimate the expectations. For tasks where both input time series and predictions are long, such as anomaly detection, the proposed method may lead to a heavy computational burden. We note that in some tasks, the interaction between time steps tend to reduce as the distance between these time steps increases. In order to reduce the computational complexity, a straightforward method is to set a threshold η, and only the interactions whose order smaller than η are computed, and the interactions with order larger than η are ignored.

Another more practical method to reduce the computational complexity is to divide the time series into patches, and then compute importance score for each patch instead of each time step. As the features in specific time step usually contribute to the prediction along with its context, thus assigning importance score to a patch is actually meaningful. Assuming the time series are divided into K patch, the computational complexity of our method will reduce to $\mathcal{O}(K^2LN)$. Note that our method does not require the patches with equal length. Thus, sophisticated change point detection algorithms (Guédon, 2013; Killick et al., 2012) can be applied to generate more meaningful patches. This patch based strategy not only reduce complexity but also provide more options for explanations, allowing explanations for a single time point as well as for arbitrary lengths of time segments.

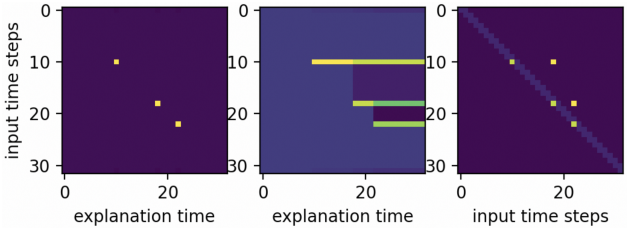

Figure 4: Explanation results of our method on $f^{(1)}(x)$ and $f^{(2)}(x)$ with the same input signals, as well as the absolute values of interactions between different time steps detected by our method for $f^{(2)}(x)$. For the left and middle images, the ith row and jth column of the image correspond to contribution of $\boldsymbol{x}_{1,i}$ to $f^{(1)}(x)_j$ and $f^{(2)}(x)_j$, respectively. For the right image, the ij-th entry of the image corresponds to the strength of the interaction $I^{i-j+1}(x_{1,i:j})$. For all images, the darker pixels denote the smaller values.

Table 1: Top-3 accuracy and AUPRC of all methods on explaining the two anomaly detectors.

	$y_t^{(1)}$		$y_t^{(2)}$	
	Acc	AUPRC	Acc	AUPRC
FIT	1.0	1.0	0.183	0.209
WinIT	1.0	1.0	0.313	0.361
FO	1.0	1.0	0.257	0.276
AFO	1.0	1.0	0.313	0.361
DynamicMask	1.0	1.0	0.535	0.464
IG	1.0	1.0	0.872	0.936
GRADSHAP	1.0	1.0	0.665	0.771
DeepLIFT	1.0	1.0	0.782	0.893
FDTempExplainer	**1.0**	**1.0**	**0.9967**	**0.9968**

4. Experiments

To measure the performance of our method quantitatively, we conduct experiments in three major types of time series tasks, including time series anomaly detection, time series classification, and time series forecasting. As there is no

ground truth explanation on real-world datasets, some synthetic datasets are used in our experiments. Throughout the experiments, we set the number of samples generated by the generator as 50.

Baselines. We compare our method with multiple state-of-the-art model-agnostic explanation methods. Specifically, our baselines include: *i)* three general explanation methods, integrated gradient (IG) (Sundararajan et al., 2017), GradSHAP (Lundberg et al., 2018), and DeepLIFT (Shrikumar et al., 2017); *ii)* two feature occlusion based methods, feature occlusion (FO) (Suresh et al., 2017) and augmented feature occlusion (AFO) (Tonekaboni et al., 2020); *iii)* one mask based method dynamicMask (Crabbé & Van Der Schaar, 2021); and *iv)* two distribution-awared methods Feature Importance in Time (FIT) (Tonekaboni et al., 2020) and Windowed Feature Importance in Time (WinIT) (Leung et al., 2023). Throughout the experiments, we set the window size of WinIT to 10. Note that both FIT and WinIT involve computing the distribution shift introduced by the observation of some time steps, so it is more suitable for the model with discrete outputs, which is not applicable for the time series forecasting tasks.

Metrics. For a given test sample $\boldsymbol{x}$ and a model $f(\cdot)$, explanation methods can assign importance scores to each feature-time step pair. We assess whether pairs with higher scores are indeed more influential to the model's output. When ground truth explanations are available, we compare the explanations generated by the methods with the ground truth, reporting their top-k accuracy and the area under the precision-recall curve (AUPRC). Let $S = (i, t)$ represent the set of true feature-time step pairs that contribute to the output. We define top-k accuracy as $\frac{|S \cap T|}{|S|}$, where T is the set of top-k feature-time step pairs identified by the explanation method.

In the absence of ground truth, we evaluate explanations by observing the deterioration in the black-box model's performance. Specifically, we select the k feature-time step pairs with the highest importance scores and replace these features at their respective time steps with non-informative values to create a set of counterfactual samples. If these pairs are genuinely important, removing them should significantly degrade the model's performance. Therefore, we assess explanation quality based on the reduction in AUPRC and accuracy after important features are replaced with non-informative values. For multi-class classification tasks, consistency is also used to gauge each method's performance. Consistency is defined as $\frac{1}{N}\sum_i \mathcal{I}(y_i = \hat{y}_i)$, where N is the number of samples, y_i is the black-box model's prediction for the i-th sample, and $\hat{y}_i$ is the prediction for the i-th sample when the feature-time step pairs with top-k importance score are masked. In our experiments, we use the average value of a given feature over time as the non-informative value for the i-th feature of a sample, denoted by $\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_{i,t}$.

4.1. Explaining Time Series Anomaly Detection

Datasets. Our experiments begin with two simulated datasets, which share the same features but with different labels. Specially, we generate 1000 time series of length 32 with 3 variables, where the values of the time series are assumed to be independent and following a Gaussian distribution with zero mean and 0.01 variance. We randomly select 3 time step for each time series and set the values of the first feature to 1 and treat them as anomalies. Given a time series $\boldsymbol{x}$, we generate two different labels $y_t^{(1)}$ and $y_t^{(2)}$ at the time step t according to $\boldsymbol{x}_{1:t}$, i.e., $y_t^{(1)}$ equals to one if $\boldsymbol{x}_{1,t} \geq 0.9$ and zero otherwise, while $y_t^{(2)}$ turns to one if $\boldsymbol{x}_{1,t'} \geq 0.9$ for some $t' \leq t$ and keep zero otherwise. Obviously, $y_t^{(1)}$ indicates whether there is an anomaly occurs in first feature of input at time step t, and thus only related to $\boldsymbol{x}_t$. In contrast, $y_t^{(2)}$ reports whether there are anomalies in history, which takes account of the effect of all the time steps no large than t. Two datasets are formed using the 1000 time series as features, and $y_t^{(1)}$ and $y_t^{(2)}$ as labels, respectively.

Experiment. We randomly split each dataset into training, validation, and test sets, with the ratio of 0.6, 0.2, and 0.2, respectively. For each dataset, we train a black-box LSTM. The testing accuracy of both two LSTMs are 1.0. Let $f^{(1)}(x)$ and $f^{(2)}(x)$ denote the LSTMs trained with $y_t^{(1)}$ and $y_t^{(2)}$. We explain $f^{(1)}(x)$ and $f^{(2)}(x)$ using our proposed functional decomposition based method and other baselines. Given an output at specific time step t, we can get importance scores of all 32 input time steps using the respective methods to measure their contribution to $f_t^{(1)}(x)$ and $f_t^{(2)}(x)$. The left and middle image in Fig. 4 show the explanation results on the first feature of the data using two anomaly detectors with the input occurs spike at time step 10, 18, and 22. The left and middle images denote explanation of $f_t^{(1)}(x)$ and $f_t^{(2)}(x)$, respectively. The pixel at ith row and jth column denotes importance of the first feature at ith time step to jth output. From Fig. 4, we can see that the explanation results clearly reveals the facts that $y_t^{(1)}$ is only related to $\boldsymbol{x}_t$, while $y_t^{(2)}$ considers all the inputs before time t. We plot the interactions between time steps detected by our method in the right image of Fig. 4, from which we can observe that our method correctly identifies the interactions between different spikes.

To further compare our method with baselines, we compute explanations for all samples in test datasets, and compute the average accuracy and AUPRC of each method. For $f^{(1)}(x)$, due to the independent of each time steps, we only consider the time steps when $y_t^{(1)} = 1$, and compute the top-

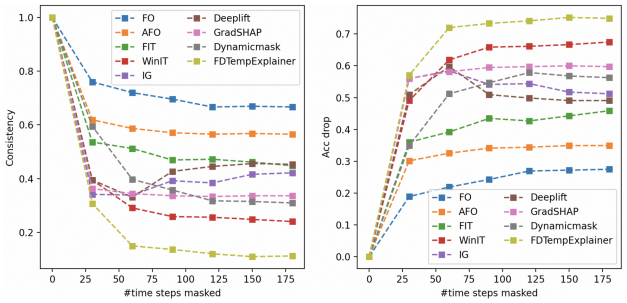

Figure 5: Consistency and ACC drop vs. number of time steps masked for different methods on Large Kitchen Appliances dataset.

1 accuracy and AUPRC for each of them. For $f^{(1)}(x)$, we compute the Top-3 accuracy and AUPRC for the attribution of the last output of the LSTM. We report the results in Table 1, from which we can observe

- All methods successfully explain $f^{(1)}(x)$. However, only FDTemExplainer achieves high accuracy in explaining $f^{(2)}(x)$.
- FDTemExplainer provides almost perfect explanations for $f^{(2)}(x)$ because it can accurately separate interaction effects from main effects.
- Methods that rely on marginal gains, such as FO, AFO, and FIT, struggle in explaining $f^{(2)}(x)$ since they do not effectively separate interactions from main effects. As outlined in Section 1, interactions between time steps have a detrimental effect on predictions, and conflating these with main effects can cause the methods to overlook critical time steps.

4.2. Explaining Time Series Classification

Datasets. We consider two real-world datasets, namely MIMIC-III and Large Kitchen Appliances datasets. MIMIC-III (Medical Information Mart for Intensive Care III) is a large, freely accessible dataset comprising de-identified health-related data associated with over 40000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center in Boston, Massachusetts. It was developed by the MIT Lab for Computational Physiology and contains data from stays occurring between 2001 and 2012. Following the preprocessing done in (Tonekaboni et al., 2020) , we obtain a dataset with 22988 samples. Each sample in the dataset contains 31 features and each feature is a time series of length 48. The label of each sample is binary, which indicates whether the patient dead in the next 48 hours. We randomly split the dataset into three parts, i.e., training set with 14942 samples, validation set with 3448 samples, and test set with 4598 samples.

Large Kitchen Appliances dataset is a public benchmark dataset from UCR [1], which contains 375 training samples and 375 testing samples. Each sample in the dataset is an univariate time series of length 720. The label is an integer number denoting the class label, including Washing Machine (1), Tumble Dryer (2), and Dishwasher (3).

Experiment. We train an LSTM for each dataset and explain the model using various explanation methods. To expedite the explanation process for the Large Kitchen Appliances dataset, we segment the time series into 72 patches, each encompassing 10 time steps. For a fair comparison, we also aggregate the importance scores for every 10 time steps produced by the baseline methods. In Table 2, we present the top-50 AUPRC drop and 95% AUPRC drop for each method on the MIMIC-III dataset, as well as the consistency and ACC drop with $k = 40$ on the Large Kitchen Appliances dataset.

The results indicate that FDTempExplainer outperforms all other methods, highlighting its proficiency in processing real-world datasets. To offer a comprehensive comparison, we also report the consistency and ACC drop with various values of k for each method on the Large Kitchen Appliances dataset in Fig. 5. The figure demonstrates that FDTempExplainer consistently outperforms other methods by a considerable margin. An additional insight from Fig. 5 is that the consistency of FDTempExplainer continuously decreases as the number of masked time steps increases. This suggests that FDTempExplainer not only excels at identifying the most crucial time steps, but also creates a more logical ordering for each time step.

4.3. Explaining Time Series Forecasting

Datasets. We conduct the experiments on a synthetic dataset, which contains a set of univariate time series $\{x_t\}$ of length 80. Each time series can be decomposed into trend and seasonal components, denoted by τ_t and s_t, respectively. Specifically, we generate τ_t, s_t and x_t according to

$$
\begin{aligned}
s_t =& \begin{cases} \beta_1 s_{t-1} + \sqrt{1-\beta_1^2}\epsilon_{1,t} & \text{if } t < 5 \\ \beta_1 s_{t-1} - \sqrt{1-\beta_1^2}\epsilon_{2,t} & \text{if } 5 \le t < 10 \\ \beta_2 s_{t-10} + \sqrt{1-\beta_2^2}\epsilon_{3,t} & \text{if } t \ge 10 \end{cases} \\
\tau_t =& \beta_3 \tau_{t-1} + \sqrt{1-\beta_3^2}\epsilon_{4,t} \\
x_t =& \tilde{s_t} + \tilde{\tau_t}, \qquad (7)
\end{aligned}
$$

where $\epsilon_{1,t}$ and $\epsilon_{2,t}$ follow the uniform distribution, while $\epsilon_{3,t}$ and $\epsilon_{4,t}$ follows normal distribution. The parameters β_1, β_2 and β_3 are set to 0.9, 0.99 and 0.99, respectively. $\tilde{s_t}$ and $\tilde{\tau_t}$ denote the normalized s_t and τ_t to ensure the unit variance. The final dataset contains 1000 samples generated by repeating the above process 1000 times.

[1] https://www.cs.ucr.edu/~eamonn/time_series_data/

Table 2: Performance comparison on MIMIC-III and Large Kitch Appicants datasets.

	MIMIC-III		Large Kitch Applicants	
	AUPRC-drop (95-pc) ↑	AUPRC-drop (k=50) ↑	Consistency (k=40) ↓	ACC drop (k=40) ↑
IG	0.017	0.030	0.3333	0.576
GRADSHAP	0.016	0.028	0.376	0.552
DeepLIFT	0.016	0.022	0.336	0.5733
FO	0.010	0.011	0.7493	0.2
AFO	0.015	0.019	0.608	0.3067
FIT	0.016	0.027	0.5387	0.36
WinIT	0.029	0.038	0.344	0.5573
DynamicMask	0.026	0.039	0.464	0.456
FDTempExplainer	**0.030**	**0.043**	**0.2293**	**0.6453**

Table 3: Top-2 accuracy and AUPRC of different methods on explaining the time series forecaster.

	FO	AFO	Dynamic Mask	IG	GRAD SHAP	DEEP LIFT	FDTemp Explainer
ACC	0.02	0.02	0.5	0.5375	0.6125	0.5	0.6438
AUPRC	0.060	0.0595	0.1375	0.6443	0.7194	0.6194	0.7306

Table 4: Running time, consistency and ACC drop of FDTempEXplainer on Large Kitch Appicants dataset with varying Q.

Q	Time (s)	Consistency (k=40)	ACC Drop (k=40)
5	15	0.2773	0.5973
10	32	0.216	0.6533
None	71	0.2293	0.6427

Experiment. We randomly split 1000 samples into training and test sets of size 800 and 200, respectively. We train an LSTM on the training set to predict the value of the next time step. Specifically, we input the LSTM with $x_{1:t}$ to predict x_{t+1}. The mean square error of LSTM on the test dataset is 0.27. Obviously, the ground truth of important time steps on predicting x_{t+1} is x_t and x_{t-10}, thus we report the top-2 accuracy of each method in Table 3. From the table we can see that the FDTempExplainer outperforms the rest methods.

4.4. Evaluations of Two Strategies to Reduce the Computational Complexity

We evaluate the two strategies proposed in Section 3.5 for fast computation, including 1) setting a threshold to circumvent calculating all the interactions; and 2) computing the importance of patch by segmenting time series into patches. We conduct experiments on the Large Kitchen Appliance dataset, showing that cutting high-order interactions and increasing patch size significantly reduce the running time, while the performances remain approximately similar.

In the first experiment, we investigate the strategy of ignoring higher-order interactions. We evaluate the performance of our method while disregarding interactions above the Q-order. Consistent with the experiments in Section 4.2, we divide the time series into 72 segments, and test the the running time of our method with $Q = 5$ and $Q = 10$. The results are summarized in Table 4. The accuracy drop and consistency when masking 40 time steps with the highest importance scores are also reported. The results indicate that by disregarding high-order interactions, the running time decreases significantly. However, we also observed that too small threshold, such as only considering interactions up to the fifth order, can result in overlooking important interactions and thus deteriorates the method's performance.

Table 5: Running time, consistency and ACC drop of FDTempExplainer on Large Kitch Appicants dataset with varying patch size.

Patch size	Runtime (s)	Consistency			ACC drop		
		k=10	k=20	k=40	k=10	k=20	k=40
5	285	0.6347	0.4533	0.2587	0.288	0.4453	0.6
10	71	0.664	0.4507	0.2293	0.264	0.448	0.6427
20	17	-	0.4987	0.2213	-	0.4027	0.6453

In the second experiment, we test our method with varying segment sizes, specifically 5, 10, and 20 time steps per segment, while considering all interactions. The results of this experiments are presented in Table 5. From Table 5, we can observe that increasing the segment size substantially reduces computational complexity. Contrary to setting a threshold, segmenting the time series allows the FDTempExplainer to maintain performance without significant degradation.

5. Conclusion

In this paper, we propose a model-agnostic explanation method FDTempExplainer based on our rigorous framework. It is applicable for all major types of time series tasks, including time series anomaly detection, classification, and forecasting. In the future, we plan to apply it in more real-world time series applications, such as electric load forecasting and its explanation. In particular, we notice that the temporal interactions learned by our framework also reveal important information about the sequence data, and we plan to explore it in sequence related tasks.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Amann, J., Blasimme, A., Vayena, E., Frey, D., Madai, V. I., and Consortium, P. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

Boniol, P. and Palpanas, T. Series2graph: graph-based subsequence anomaly detection for time series. *Proc. VLDB Endow.*, 13(12):1821–1834, jul 2020. ISSN 2150-8097. doi: 10.14778/3407790.3407792. URL https://doi.org/10.14778/3407790.3407792.

Campbell, S. D. and Diebold, F. X. Weather forecasting for weather derivatives. *Journal of the American Statistical Association*, 100(469):6–16, 2005.

Crabbé, J. and Van Der Schaar, M. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, pp. 2166–2177. PMLR, 2021.

Enguehard, J. Learning perturbations to explain time series predictions. In *International Conference on Machine Learning*, pp. 9329–9342. PMLR, 2023.

Guédon, Y. Exploring the latent segmentation space for the assessment of multiple change-point models. *Computational Statistics*, 28(6):2641–2678, 2013.

Ismail, A. A., Gunady, M., Corrada Bravo, H., and Feizi, S. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Accurate and interpretable evaluation of surgical skills from kinematic data using fully convolutional neural networks. *International journal of computer assisted radiology and surgery*, 14:1611–1617, 2019.

Ivaturi, P., Gadaleta, M., Pandey, A. C., Pazzani, M., Steinhubl, S. R., and Quer, G. A comprehensive explanation framework for biomedical time series classification. *IEEE journal of biomedical and health informatics*, 25(7):2398–2408, 2021.

Kendall, M. G. and Hill, A. B. The analysis of economic time-series-part i: Prices. *Journal of the Royal Statistical Society. Series A (General)*, 116(1):11–34, 1953.

Killick, R., Fearnhead, P., and Eckley, I. A. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500): 1590–1598, 2012.

Koo, J., Shin, D., Steinert, M., and Leifer, L. Understanding driver responses to voice alerts of autonomous car operations. *International journal of vehicle design*, 70(4): 377–392, 2016.

Leung, K. K., Rooke, C., Smith, J., Zuberi, S., and Volkovs, M. Temporal dependencies in feature importance for time series prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., Liston, D. E., Low, D. K.-W., Newman, S.-F., Kim, J., et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10): 749–760, 2018.

Meng, H., Wagner, C., and Triguero, I. Explaining time series classifiers through meaningful perturbation and optimisation. *Information Sciences*, pp. 119334, 2023.

Mobley, R. K. *An introduction to predictive maintenance*. Elsevier, 2002.

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, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

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.

Sivill, T. and Flach, P. Limesegment: Meaningful, realistic time series explanations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3418–3433. PMLR, 2022.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Sun, X., Wang, Z., Ding, R., Han, S., and Zhang, D. puregam: Learning an inherently pure additive model. In *Proceedings of the 28th ACM SIGKDD Conference on*

Knowledge Discovery and Data Mining, pp. 1728–1738, 2022.

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

Suresh, H., Hunt, N., Johnson, A., Celi, L. A., Szolovits, P., and Ghassemi, M. Clinical intervention prediction and understanding using deep networks. *arXiv preprint arXiv:1705.08498*, 2017.

Tjoa, E. and Guan, C. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2020.

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. *Advances in Neural Information Processing Systems*, 33:799–809, 2020.

Villani, M., Lockhart, J., and Magazzeni, D. Feature importance for time series data: Improving kernelshap. *arXiv preprint arXiv:2210.02176*, 2022.

Yang, W., Wei, Y., Wei, H., Chen, Y., Huang, G., Li, X., Li, R., Yao, N., Wang, X., Gu, X., et al. Survey on explainable ai: From approaches, limitations and applications aspects. *Human-Centric Intelligent Systems*, 3(3): 161–188, 2023.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D. I., and Ravikumar, P. K. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Zeger, S. L., Irizarry, R., and Peng, R. D. On time series analysis of public health and biomedical data. *Annu. Rev. Public Health*, 27:57–79, 2006.

A. Proofs.

A.1. Proof of Lemma 3.2

We prove Lemma 3.2 using the induction method. In our proof, we omit the context variable $\boldsymbol{C}$. We first demonstrate that any function with an input time series of length 2, that is, $f(\boldsymbol{X}_{1:2})$, can be decomposed into the form presented in (1) with all interactions being pure.

For $f(\boldsymbol{X}_{1:2})$, we define

$$
\begin{aligned}
I^2(\boldsymbol{X}_{1:2}) = f(\boldsymbol{X}_{1:2}) &- \int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1 \\
&- \int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \\
&+ \int\int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2, \quad (8)
\end{aligned}
$$

$$
m(\boldsymbol{x}_1) = \int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2, \quad (9)
$$

$$
m(\boldsymbol{x}_2) = \int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1, \quad (10)
$$

$$
C_0 = -\int\int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2. \quad (11)
$$

Clearly, we have $f(\boldsymbol{X}_{1:2}) = I^2(\boldsymbol{X}_{1:2}) + m(\boldsymbol{x}_1) + m(\boldsymbol{x}_2) + C_0$. Next, we will demonstrate that $I^2(\boldsymbol{X}_{1:2})$ is pure. This implies that both $\int I^2(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2$ and $\int I^2(\boldsymbol{x}_1, \bar{\boldsymbol{x}}_2) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1$ are equal to zero. We will show it for $\int I^2(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2$, the other term can be evaluated in a similar manner. Specifically, we have

$$
\begin{aligned}
&\int I^2(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \\
=&\int f(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \\
&- \int \left(\int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1 \right) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \\
&- \int \left(\int f(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \right) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \\
&+ \int\int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\
=&\int f(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_1 \\
&- \int\int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\
&- \int f(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) d\boldsymbol{x}_2 \\
&+ \int\int f(\boldsymbol{x}_1, \boldsymbol{x}_2) p(\boldsymbol{x}_2|\bar{\boldsymbol{x}}_1) p(\boldsymbol{x}_1|\bar{\boldsymbol{x}}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\
=&0. \quad (12)
\end{aligned}
$$

Thus, we can conclude that any function with an input time series of length 2 admits a decomposition with pure interaction. We then prove that for a function with an input time series of length $k+1$, a decomposition with all interactions being pure is possible, assuming that any function with the length of the input time series no greater than k can be similarly decomposed. Suppose $f(\boldsymbol{X}_{1:k+1})$ can be decomposed as

$$
f(\boldsymbol{X}_{1:k+1}) = I^{k+1}(\boldsymbol{X}_{1:k+1}) + Q_1(\boldsymbol{X}_{1:k}) + Q_2(\boldsymbol{X}_{2:k+1}),
$$

where $I^{k+1}(\boldsymbol{X}_{1:k+1})$ denotes the $(k+1)$-th order interaction, and $Q_1(\boldsymbol{X}_{1:k})$ and $Q_2(\boldsymbol{X}_{2:k+1})$ include the mean effects of each time step and interactions of order lower than $k+1$. As the inputs of $Q_1(\boldsymbol{X}_{1:k})$ and $Q_2(\boldsymbol{X}_{2:k+1})$ are of length k, they can be further decomposed, with all interactions being pure. Hence, our discussion focuses on purifying the $(k+1)$-th order interaction. Specifically, if there exists a subset P and time indices t_1 and t_2 such that $\int I^{k+1}(\boldsymbol{Z}_{t_1:t_2}) p(\boldsymbol{X}_P|\bar{\boldsymbol{X}}_{t_1:t_2}, \bar{\boldsymbol{C}}) d\boldsymbol{X}_P \neq 0$, where $\boldsymbol{Z}_{t_1:t_2} = [\boldsymbol{X}_{1:t_1-1}, \bar{\boldsymbol{X}}_{t_1:t_2}, \boldsymbol{X}_{t_2+1:k+1}]$. Then we can construct a new interaction, $\tilde{I}^{k+1}(\boldsymbol{X}_{1:k+1})$, as follows:

$$
\begin{aligned}
&\tilde{I}^{k+1}(\boldsymbol{X}_{1:k+1}) \\
=&I^{k+1}(\boldsymbol{X}_{1:k+1}) \\
&- \int I^{k+1}(\boldsymbol{X}_{1:k+1}) p(\boldsymbol{X}_P|\bar{\boldsymbol{X}}_{t_1:t_2}, \bar{\boldsymbol{C}}) d\boldsymbol{X}_P.
\end{aligned}
$$

It can be easily verified that $\tilde{I}^{k+1}(\boldsymbol{X}1:k+1)$ satisfies the condition that

$$
\int \tilde{I}^{k+1}(\boldsymbol{Z}_{t_1:t_2}) p(\boldsymbol{X}_P|\bar{\boldsymbol{X}}_{t_1:t_2}, \bar{\boldsymbol{C}}) d\boldsymbol{X}_P = 0. \quad (13)
$$

As $\int I^{k+1}(\boldsymbol{X}_{1:k+1}) p(\boldsymbol{X}_P|\bar{\boldsymbol{X}}_{t_1:t_2}, \bar{\boldsymbol{C}}) d\boldsymbol{X}_P$ is a function with an input length less than $k+1$, which can then be further decomposed. We can examine all constraints required for the purity of the interaction and repeat the above process if any unsatisfied constraint is found. Since the number of constraints is finite, we can ultimately obtain a pure $(k+1)$-th order interaction. Our proof is thus complete.

A.2. Proof of Lemma 3.3

We evaluate the value of $(a) - (b) - (c) + (d)$. As $f(\mathcal{Z}(t_1, t_2), \bar{\boldsymbol{C}})$ can be decomposed as

$$
\begin{aligned}
f(\mathcal{Z}(t_1, t_2), \bar{\boldsymbol{C}}) = &\sum_{k=1}^{T-1} \sum_{t=1}^{T-k+1} I^{k+1}\left(\mathcal{Z}(t_1, t_2)|_{t:t+k}\right) \\
&+ \sum_{t=1}^{T} m\left(\mathcal{Z}(t_1, t_2)|_t\right) + C_0, \quad (14)
\end{aligned}
$$

where $\mathcal{Z}(t_1, t_2)|_t$ denotes the tth column of $\mathcal{Z}(t_1, t_2)$, and $\mathcal{Z}(t_1, t_2)|_{t:t+k}$ is the submatrix of $\mathcal{Z}(t_1, t_2)$ containing the values from the tth column to $t+k$th column of $\mathcal{Z}(t_1, t_2)$.

Then we can rewrite $(a)-(b)-(c)+(d)$ as follows:

$$
\begin{aligned}
&(a)-(b)-(c)+(d)\\
=&\sum_{k=1}^{T-1}\sum_{t=1}^{T-k+1} h_1\left(I^{k+1}(\mathcal{Z}(t_1,t_2)|_{t:t+k})\right)\\
&+\sum_{t=1}^{T} h_1(m(\mathcal{Z})(t_1,t_2)|_t)\\
&-\sum_{k=1}^{T-1}\sum_{t=1}^{T-k+1} h_2\left(I^{k+1}(\mathcal{Z}(t_1+1,t_2)|_{t:t+k})\right)\\
&-\sum_{t=1}^{T} h_2(m(\mathcal{Z}(t_1+1,t_2)|_t))\\
&-\sum_{k=1}^{T-1}\sum_{t=1}^{T-k+1} h_3\left(I^{k+1}(\mathcal{Z}(t_1,t_2-1)|_{t:t+k})\right)\\
&-\sum_{t=1}^{T} h_3(m(\mathcal{Z}(t_1,t_2-1)|_t))\\
&+\sum_{k=1}^{T-1}\sum_{t=1}^{T-k+1} h_4\left(I^{k+1}(\mathcal{Z}(t_1+1,t_2-1)|_{t:t+k})\right)\\
&+\sum_{t=1}^{T} h_4(m(\mathcal{Z}(t_1+1,t_2-1)|_t))\\
=&\sum_{k=1}^{T-1}\sum_{t=1}^{T-k+1} Q_1(t,t+k)+\sum_{t=1}^{T} Q_2(t),
\end{aligned}
\tag{15}
$$

where

$$
\begin{aligned}
h_1(\cdot)=&\mathbb{E}_{1:t_1-1|t_1:t_2-1}\left[\mathbb{E}_{t_2+1:T|t_1+1:t_2}(\cdot)\right],\\
h_2(\cdot)=&\mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[\mathbb{E}_{t_2+1:T|t_1+1:t_2}(\cdot)\right],\\
h_3(\cdot)=&\mathbb{E}_{1:t_1-1|t_1:t_2-1}\left[\mathbb{E}_{t_2:T|t_1+1:t_2-1}(\cdot)\right],\\
h_4(\cdot)=&\mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[\mathbb{E}_{t_2:T|t_1+1:t_2-1}(\cdot)\right],
\end{aligned}
$$

and

$$
\begin{aligned}
Q_1(t,t+k)=&\, h_1\left(I^{k+1}(\mathcal{Z}(t_1,t_2)|_{t:t+k})\right)\\
&-h_2\left(I^{k+1}(\mathcal{Z}(t_1+1,t_2)|_{t:t+k})\right)\\
&-h_3\left(I^{k+1}(\mathcal{Z}(t_1,t_2-1)|_{t:t+k})\right)\\
&+h_4\left(I^{k+1}(\mathcal{Z}(t_1+1,t_2-1)|_{t:t+k})\right)\\
Q_2(t)=&\, h_1(m(\mathcal{Z})(t_1,t_2)|_t)-h_2(m(\mathcal{Z}(t_1+1,t_2)|_t))\\
&-h_3(m(\mathcal{Z}(t_1,t_2-1)|_t))+h_4(m(\mathcal{Z}(t_1+1,t_2-1)|_t)).
\end{aligned}
\tag{16}
$$

We now explore $Q_1(n_1,n_2)$ and $Q_1(n_1)$ by enumerating all possible scenarios regarding n_1 and n_2. Specifically, we analyze the value of $Q_1(n_1,n_2)$ for the following cases: $n_2<t_2$, $n_1<t_1<t_2\le n_2$, $n_1=t_1<t_2=n_2$, $t_1=n_1<t_2<n_2$, and $n_1>t_1$, respectively.

We first discuss the scenarios $n_2<t_2$. In this case, $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})$, and $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})$ are independent of $\boldsymbol{X}_{t_2:T}$. Thus the inner expectations in h_1, h_2, h_3, and h_4 can be omitted, i.e.,

$$
\begin{aligned}
&h_1\left(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})\right)\\
&\quad=\mathbb{E}_{1:t_1-1|t_1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2}\right]\\
&h_2(I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2}))\\
&\quad=\mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2})\right],\\
&h_3(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})\\
&\quad=\mathbb{E}_{1:t_1-1|t_1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})\right],\\
&h_4(I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2}))\\
&\quad=\mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})\right].
\end{aligned}
$$

Recalling the definition of $\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2}$, we observe that $\mathcal{Z}(t_1,t_2)|_{n_1:n_2}=\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2}$ when $n_2<t_2$. Consequently, we can derive

$$
\begin{aligned}
&h_1\left(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})\right)\\
=&h_3(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2}).
\end{aligned}
$$

And by a similar argument, we obtain

$$
\begin{aligned}
&h_2(I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2}))\\
&=h_4(I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})).
\end{aligned}
$$

These results lead us to conclude that $Q_1(n_1,n_2)=0$ for $n_2<t_2$.

We next consider the scenarios $n_1<t_1<t_2\le n_2$ and $t_1=n_1<t_2<n_2$. Since the interactions $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})$, and $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})$ are pure, their expectations conditioned on $p(\boldsymbol{X}_{t_2+1:T}|\bar{\boldsymbol{X}}_{t_1+1:t_2})$ and $p(\boldsymbol{X}_{t_2:T}|\bar{\boldsymbol{X}}_{t_1+1:t_2-1})$ are zero. This results in $Q_1(n_1,n_2)=0$ for these scenarios.

In the following, we discuss the scenario where $n_1=t_1<t_2=n_2$. In this scenario, $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})$ is independent of both $\boldsymbol{X}_{1:t_1-1}$ and $\boldsymbol{X}_{t_2:T}$. Thus, we can omit all expectations in h_1 and obtain

$$
h_1\left(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})\right)=I^{n_1-n_2+1}(\bar{\boldsymbol{X}}_{t_1:t_2}).
$$

On the other hand, since the interactions $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})$, and $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})$ are pure, their expectations conditional on the relevant segments are zero, i.e.,

$$
\begin{aligned}
&\mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2})\right]=0,\\
&\mathbb{E}_{t_2:T|t_1+1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})\right]=0,\\
&\mathbb{E}_{1:t_1|t_1+1:t_2-1}\left[I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})\right]=0.
\end{aligned}
$$

Consequently, we determine that $Q_1 = I^{n_1-n_2+1}(\bar{\boldsymbol{X}}_{t_1:t_2})$ for the case where $n_1 = t_1 < t_2 = n_2$.

Finally, we consider the scenarios where $n_1 > t_1$, which are similar to those where $n_2 < t_2$. Since the interactions $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2})$, and $I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})$ are independent of $\boldsymbol{X}_{1:t_1}$, thus we can remove the outer expectation in h_1, h_2, h_3 and h_2. We can then derive the following relations:

$$\begin{aligned}&h_1\left(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2)|_{n_1:n_2})\right)\\=&h_2(I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2)|_{n_1:n_2}).\end{aligned}$$

Similarly, we have

$$\begin{aligned}&h_3(I^{n_1-n_2+1}(\mathcal{Z}(t_1,t_2-1)|_{n_1:n_2}))\\&=h_4(I^{n_1-n_2+1}(\mathcal{Z}(t_1+1,t_2-1)|_{n_1:n_2})).\end{aligned}$$

This leads to the conclusion that $Q_1(n_1,n_2)=0$ for $n_1 > t_1$.

Summing the values of Q_1 across all scenarios, we can conclude that Q_1 is equal to $I^{n_1-n_2+1}(\bar{\boldsymbol{X}}_{t_1:t_2})$ if $n_1 = t_1$ and $n_2 = t_2$, and it is zero otherwise. We then need to prove that $Q_2(t) = 0$.

In the following, we discuss the value of $Q_2(t)$. Clearly, if $t \notin t_1-1, t_1, t_1+1, t_2-1, t_2, t_2+1$, then $h_1(m(\mathcal{Z})(t_1,t_2)|_t) = h_2(m(\mathcal{Z}(t_1+1,t_2)|_t)) = h_3(m(\mathcal{Z}(t_1,t_2-1)|_t)) = h_4(m(\mathcal{Z}(t_1+1,t_2-1)|_t))$, and consequently, $Q_2(t) = 0$. If $t \in t_1-1, t_1, t_1+1$, then $h_1(m(\mathcal{Z})(t_1,t_2)|_t) = h_3(m(\mathcal{Z}(t_1,t_2-1)|_t))$ and $h_2(m(\mathcal{Z}(t_1+1,t_2)|_t)) = h_4(m(\mathcal{Z}(t_1+1,t_2-1)|_t))$, which implies $Q_2(t) = 0$. Similarly, if $t \in t_2-1, t_2, t_2+1$, then $h_1(m(\mathcal{Z})(t_1,t_2)|_t) = h_2(m(\mathcal{Z}(t_1+1,t_2)|_t))$ and $h_3(m(\mathcal{Z}(t_1,t_2-1)|_t)) = h_4(m(\mathcal{Z}(t_1+1,t_2-1)|_t))$, leading to $Q_2(t) = 0$. Therefore, we conclude that $Q_2(t) = 0$ for any $1 \le t \le T$. Our proof is complete.

A.3. Proof of Lemma 3.4

The proof is similar to that of Lemma 3.3. We evaluate the following expression by discussing the values of the interactions between time steps n_1 and n_2:

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}(t,t+1),\bar{\boldsymbol{C}}))]\\&-\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+1:T|t}(f(\mathcal{Z}(t)),\bar{\boldsymbol{C}}))]\\&-\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}(t+1)),\bar{\boldsymbol{C}}))]\\&+\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(f(\boldsymbol{X},\bar{\boldsymbol{C}}))].\end{aligned}$$

The derivation regarding the main effects is the same as in Lemma 3.3, so we omit it here. For $n_2 \le t$, $I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2})$ is independent of $\boldsymbol{X}_{t+1:T}$. We then have

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\=&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2}))],\end{aligned}\tag{17}$$

and

$$\begin{aligned}&\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t+1)|_{n_1:n_2}))]\\=&\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2}))].\end{aligned}\tag{18}$$

Thus, it leads to

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\&-\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2}))]\\&-\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\&+\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2}))]\\=&0.\end{aligned}$$

Next, we consider the scenarios where $n_1 < t < t+1 \le n_2$ or $n_1 \le t < t+1 \le n_2$. Since the interactions $I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2})$, $I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2})$, and $I^{n_1-n_2+1}(\mathcal{Z}(t+1)|_{n_1:n_2})$ are pure, we obtain

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\&-\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2}))]\\&-\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t+1)|_{n_1:n_2}))]\\&+\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2}))]\\=&\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2}))]\\\approx&0.\end{aligned}\tag{19}$$

The approximation in Eq. (19) holds due to the fact that time series data is highly correlated and $\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2})) \approx \mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2})$.

We then address the scenarios where $n_1 = t < t+1 = n_2$. Similarly, since $I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2})$ is independent of $\boldsymbol{X}_{1:t-1}$ and $\boldsymbol{X}_{t+1:T}$, we have

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\=&I^2(\boldsymbol{X}_{t:t+1}).\end{aligned}$$

Furthermore, due to the purity of $I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2})$ and $I^{n_1-n_2+1}(\mathcal{Z}(t+1)|_{n_1:n_2})$, we have

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\&-\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2}))]\\&-\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}(t+1)),\bar{\boldsymbol{C}}))]\\&+\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2})]\\=&I^2(\boldsymbol{X}_{t:t+1})+\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2}))]\\\approx&I^2(\boldsymbol{X}_{t:t+1}).\end{aligned}\tag{20}$$

Finally, we discuss the scenarios where $n_1 > t$. Note that $I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2})$ and $I^{n_1-n_2+1}(\mathcal{Z}(t+1)|_{n_1:n_2})$ are independent of $\boldsymbol{X}_{1:t}$. This leads to

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t,t+1)|_{n_1:n_2}))]\\=&\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+2:T|t+1}(I^{n_1-n_2+1}(\mathcal{Z}(t+1)|_{n_1:n_2}))],\end{aligned}$$

and

$$\begin{aligned}&\mathbb{E}_{1:t-1|t}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\mathcal{Z}(t)|_{n_1:n_2}))]\\=&\mathbb{E}_{1:t|t+1}[\mathbb{E}_{t+1:T|t}(I^{n_1-n_2+1}(\boldsymbol{X}_{n_1:n_2}))].\end{aligned}$$

Considering all the above discussions, we can conclude

$$\begin{aligned}I^2(\bar{\boldsymbol{X}}_{t:t+1}) \approx&\mathbb{E}_{1:t-1|t}\left[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}(t,t+1),\bar{\boldsymbol{C}}))\right]\\&-\mathbb{E}_{1:t-1|t}\left[\mathbb{E}_{t+1:T|t}(f(\mathcal{Z}_t,\bar{\boldsymbol{C}}))\right]\\&-\mathbb{E}_{1:t|t+1}\left[\mathbb{E}_{t+2:T|t+1}(f(\mathcal{Z}_{t+1},\bar{\boldsymbol{C}}))\right]\\&+\mathbb{E}_{1:t|t+1}\left[\mathbb{E}_{t+1:T|t}(f(\boldsymbol{X},\bar{\boldsymbol{C}}))\right].\end{aligned}\tag{21}$$

Our proof is now complete.

A.4. Proof of Lemma 3.5

We evaluate the value of

$$\mathbb{E}_{t+1:T|t-1}(f(\mathcal{Z}_{1:t},\bar{\boldsymbol{C}}))-\mathbb{E}_{t:T|t-1}(f(\mathcal{Z}_{1:t-1},\bar{\boldsymbol{C}})),$$

by utilizing the decomposition structure of $f(\mathcal{Z}_{1:t},\bar{\boldsymbol{C}})$, then we have

$$\begin{aligned}&\mathbb{E}_{t+1:T|t-1}(f(\mathcal{Z}_{1:t},\bar{\boldsymbol{C}}))-\mathbb{E}_{t:T|t-1}(f(\mathcal{Z}_{1:t-1},\bar{\boldsymbol{C}}))\\=&\mathbb{E}_{t+1:T|t-1}(\sum_{k=1}^{T-1}\sum_{i=1}^{T-k+1}I^{k+1}\left(\mathcal{Z}(1,t)|_{i:i+k}\right))\\&+\mathbb{E}_{t+1:T|t-1}(\sum_{i=1}^{T}m\left(\mathcal{Z}(1,t)|_i\right))+C_0\\&-\mathbb{E}_{t:T|t-1}(\sum_{k=1}^{T-1}\sum_{i=1}^{T-k+1}I^{k+1}\left(\mathcal{Z}(1,t-1)|_{i:i+k}\right))\\&-\mathbb{E}_{t:T|t-1}(\sum_{i=1}^{T}m\left(\mathcal{Z}(1,t-1)|_i\right))-C_0\\=&\mathbb{E}_{t+1:T|t-1}(\sum_{k=1}^{T-1}\sum_{i=1}^{T-k+1}I^{k+1}\left(\mathcal{Z}(1,t)|_{i:i+k}\right))\\&-\mathbb{E}_{t:T|t-1}(\sum_{k=1}^{T-1}\sum_{i=1}^{T-k+1}I^{k+1}\left(\mathcal{Z}(1,t-1)|_{i:i+k}\right))\\&+m\left(\mathcal{Z}(1,t)|_t\right))-\mathbb{E}_{t|t-1}m\left(\mathcal{Z}(1,t-1)|_t\right)\\=&\sum_{k=1}^{t-1}I^{k+1}\left(\mathcal{Z}(1,t)|_{t-k:t}\right)\\&+m\left(\mathcal{Z}(1,t)|_t\right))-\mathbb{E}_{t|t-1}m\left(\mathcal{Z}(1,t-1)|_t\right)\\=&\sum_{k=1}^{t-1}I^{k+1}\left(\bar{\boldsymbol{X}}_{t-k:t}\right)+m\left(\bar{\boldsymbol{x}}_t\right))-\int m(\boldsymbol{x}_t)p(\boldsymbol{x}_t|\bar{\boldsymbol{x}}_{t-1})d\boldsymbol{x}_t.\end{aligned}$$

Our proof is now complete.

B. Related Work

A wide range of explanation methods for temporal models have been proposed in the literature. Due to the space limit, in this section we focus on model-agnostic methods only.

Local Approximation-based Methods. A common class of explanation methods involves locally approximating the black-box model with an interpretable model. Notable examples include LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2017), and Integrated Gradient (Sundararajan et al., 2017), which all fall into this category (Yeh et al., 2019). These methods generally start by generating a set of counterfactual samples based on a predefined policy. The significance of each sample is then measured, and an importance score is obtained by training a simple interpretable model with these samples. However, since these methods often overlook the temporal correlations in the data, applying them directly to time series can result in out-of-distribution (OOD) samples, leading to inaccurate explanations. To address this issue, (Sivill & Flach, 2022) proposes generating counterfactual samples using harmonic analysis to find realistic background patterns for perturbations. Another strand of research attempts to incorporate temporal correlations by employing auto-regressive models as interpretable white-box models (Villani et al., 2022).

Methods based on Masking and Perturbation. A different method involves obscuring parts of the time series to assess the impact on the model's predictions. The assumption is that some parts of the data that can be obscured without changing the predictions are deemed less important. The challenge is in deciding which parts to obscure and how to do so without creating data that is uncharacteristic of the model's training data. One early solution Dynamask (Crabbé & Van Der Schaar, 2021), minimizes the alteration to the model's predictions by applying selective changes to the data and evaluating the outcomes. It employs blurring methods to ensure these changes are in line with the data distribution. Nevertheless, this method may not capture long-term dependencies in the data. Addressing this, recent advancements (Enguehard, 2023) have introduced neural networks which learn to make changes consistent with the data distribution.

Distribution-Aware Methods. Another approach to generating explanations evaluates the distribution of input features and the effect that variations within this distribution have on model outputs. The Feature Importance in Time (FIT) method (Tonekaboni et al., 2020) is a typical example, investigating the impact of plausible input variations on the predictions made by the model. This method employs a metric based on KL-divergence to assess the significance of each input. Recognizing the importance of temporal dependencies between time steps, (Leung et al., 2023) introduces the Windowed Feature Importance in Time (WinIT).

This method highlights the fluctuating relevance of a feature across time by evaluating its importance within a specified window of preceding time steps, thus capturing the dynamic nature of feature influence in temporal data.

C. Implementation Details

Handling Multivariate Time Series. We outline the detailed process of determining feature importance for multivariate time series data using our method. Let $\tilde{\boldsymbol{X}} \in \mathbb{R}^{D\times T}$ represent the multivariate input time series, which consists of D features across T time steps. To gauge the importance of feature d, we isolate the dth row of $\tilde{\boldsymbol{X}}$ as $\boldsymbol{X} \in \mathbb{R}^{1\times T}$, and treat the remaining features as the matrix $\boldsymbol{C} \in \mathbb{R}^{(D-1)\times T}$. We then apply our FDTempExplainer to derive the importance score for each time step within $\boldsymbol{X}$. This procedure is iterated by sequentially treating each row of $\tilde{\boldsymbol{X}}$ as $\boldsymbol{X}$, with the remaining rows as $\boldsymbol{C}$. The process is repeated for all features, ultimately yielding the importance score for each feature at every time step.

Configure of LSTMs. The configuration of LSTMs as black-box models in each task is summarized in Table 6.

Table 6: Configuration of LSTMs in respective tasks.

Parameters	Anomaly detection	MIMIC-III	Large Kitch Applicants	Forecasting
Latent size	20	200	120	120
# layers	3	4	3	4
Drop out	0.4	0.6	0.4	0.4
Optimizer	Adam lr=0.01, $\beta_1 = 0.8$, $\beta_2 = 0.9$	Adam lr=0.002, $\beta_1 = 0.8$, $\beta_2 = 0.9$	Adam lr=0.001, $\beta_1 = 0.8$, $\beta_2 = 0.9$	Adam lr=0.01, $\beta_1 = 0.8$, $\beta_2 = 0.9$
epoch	100	100	200	300

D. Effect of Conditional Generative Model

The generator in our algorithm is only used to capture the interdependencies among various time steps, ensuring that the samples used to compute the conditional expectation are not out-of-distribution. In fact, the requirement for a good generator is mild. Specifically, we consider two objectives in the generator. Firstly, the generated samples should be diverse enough to ensure that our algorithm does not mistakenly interpret partial input as complete information. Secondly, the generated samples should stay within the distribution and do not cause erratic behavior in the model. We emphasize that CVAE is not the sole choice for sample generation; other state-of-the-art generative models could also be applied to estimate the conditional expectation described in our study.

Note that many methods for time series explanation rely on training such a generator, such as FIT and WinIT. Although generating accurate samples can be difficult, it is still feasible to produce time series samples that have similar trends and means. Moreover, some models, especially those designed for time series classification and anomaly detection, are quite robust to the generated samples. These models typically output a '1' only when a particular pattern is present in the sequence, and generating such specific patterns is challenging. This observation makes the final explanation method relatively robust to the dependency on the generator.

To demonstrate the impact of the generator on the explanation outcomes, we conduct tests using varying numbers of samples generated by the generator. We have repeated our method 10 times on the Large Kitchen Appliance dataset and reported the standard derivation in our explanations in Table 7.

From the table, it is evident that the proposed method can generate stable explanations on the Large Kitchen Appliance dataset. Furthermore, as the number of samples used increases, the variance of the explanations decreases, indicating enhanced explanation consistency.

E. Results of Directly Explaining White-box Models in Section 4.1 by FDTempExplainer

In Section 4.1, we generate $y^{(1)}$ and $y^{(2)}$ according to

$$y_t^{(1)} = \begin{cases} 1 & \text{if } \boldsymbol{x}_{1,t} \geq 0.9 \\ 0 & \text{otherwise} \end{cases}, \tag{22}$$

and

$$y_t^{(2)} = \begin{cases} 1 & \text{if } \boldsymbol{x}_{1,t'} \geq 0.9 \qquad \forall t' \leq t \\ 0 & \text{otherwise} \end{cases}. \tag{23}$$

We then explain the LSTM trained using the generated datasets. In the following, we report the results of FDTempExplainer on directly explaining the above white-box models in Table 8. From the table we can see that the proposed method provides almost perfect explanations on both y_1 and y_2.

F. Examples of Explanations Generated by FDTempExplainer

Table 7: Standard derivation of our methods on Large Kitch Appicants dataset with varying number of samples generated by the generator.

	$N = 1$	$N = 2$	$N = 5$	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
STD	0.0685	0.0662	0.0656	0.0649	0.0647	0.0644	0.0644	0.0642

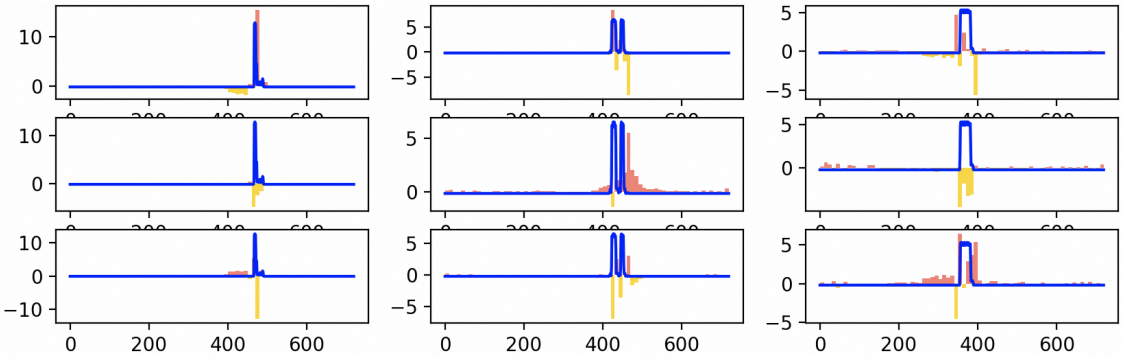

Figure 6: The explanation generated by FDTempExplainer. The plot at ith row and jth columns explains why the ith sample is classified or not classified as the jth class. The blue lines is the original signal, and the red and yellow stems represent the positive and negative contributions, respectively.

Table 8: Performance of FDTempExplainer on explaining y_1 and y_2.

y_1	y_1	y_2	y_2
ACC	AUPRC	ACC	AUPRC
1	1	0.9967	0.9968

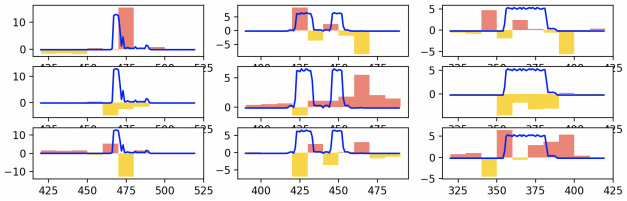

Figure 7: Detailed View of Fig. 6

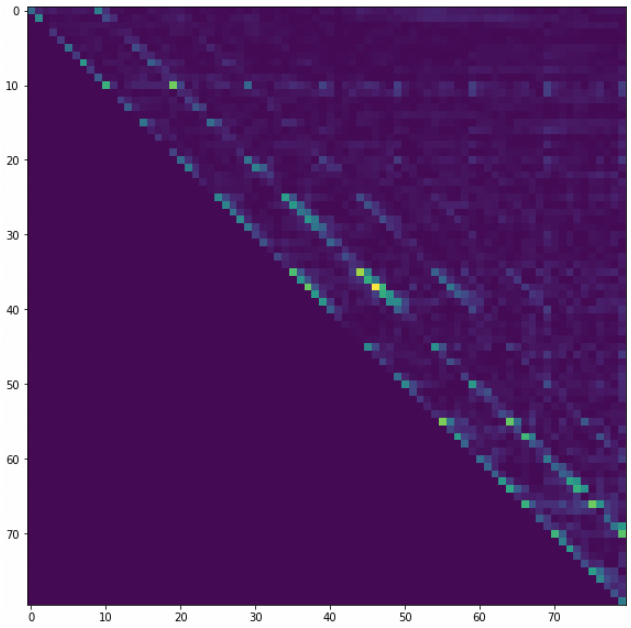

Figure 8: Absolute value of importance score generated by FDTempExplainer. Each pixel at the intersection of the ith row and jth column represents the importance of the ith time step in predicting the jth output. The lighter the pixel, the greater the value.