TIMEX++: Learning Time-Series Explanations with Information Bottleneck

Zichuan Liu [1 2 *] Tianchun Wang [3 *] Jimeng Shi [4] Xu Zheng [4] Zhuomin Chen [4] Lei Song [2]
Wenqian Dong [4] Jayantha Obeysekera [4] Farhad Shirani [4] Dongsheng Luo [4]

Abstract

Explaining deep learning models operating on time series data is crucial in various applications of interest which require interpretable and transparent insights from time series signals. In this work, we investigate this problem from an information theoretic perspective and show that most existing measures of explainability may suffer from trivial solutions and distributional shift issues. To address these issues, we introduce a simple yet practical objective function for time series explainable learning. The design of the objective function builds upon the principle of information bottleneck (IB), and modifies the IB objective function to avoid trivial solutions and distributional shift issues. We further present TIMEX++, a novel explanation framework that leverages a parametric network to produce explanation-embedded instances that are both in-distributed and label-preserving. We evaluate TIMEX++ on both synthetic and real-world datasets comparing its performance against leading baselines, and validate its practical efficacy through case studies in a real-world environmental application. Quantitative and qualitative evaluations show that TIMEX++ outperforms baselines across all datasets, demonstrating a substantial improvement in explanation quality for time series data. The source code is available at https://github.com/zichuan-liu/TimeXplusplus.

1. Introduction

Deep learning has become a cornerstone technology in analyzing time series data, prevalent in scenarios such as finance (Bento et al., 2021), healthcare (Kaushik et al., 2020), and environmental science (Adebayo et al., 2021). Despite its successes, a critical limitation of deep learning in time series analysis is the lack of explainability, which is crucial for gaining trust and actionable insights in these sensitive and impactful domains. For instance, in environmental science, the ability to explain deep learning predictions is vital for understanding complex ecological dynamics and making informed, impactful policy decisions (Reichstein et al., 2019; Razavi, 2021; Adebayo et al., 2021).

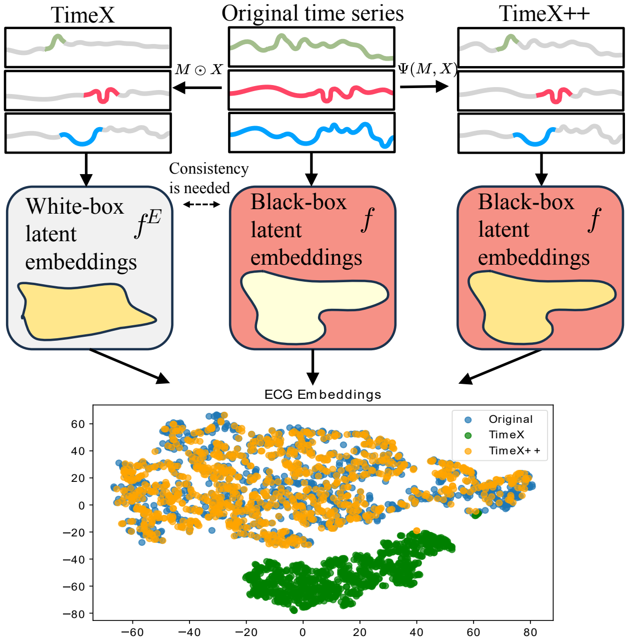


Figure 1. A comparison of our model and the reference model, where the latent embeddings of explanations are learned from the ECG dataset. Our explanations are within the original distribution, whereas explanations of the reference model are not.

Current efforts in enhancing explainability in deep learning for time series analysis primarily focus on pinpointing significant locations of time series signals that dominate the model's prediction in a post-hoc sense. For example, Shi et al. (2023a) explained their trained models for water level prediction using LIME, a local interpretable model-agnostic explanation technique (Ribeiro et al., 2016). On top of this intuitive principle, perturbation-based methods,

[*] Equal contribution [1] Nanjing University [2] Microsoft Research Asia [3] Pennsylvania State University [4] Florida International University. Correspondence to: Farhad Shirani <fshirani@fiu.edu>, Dongsheng Luo <dluo@fiu.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

including Dynamask (Crabbé & Van Der Schaar, 2021) and Extrmask (Enguehard, 2023), offer insights by altering non-salient features to assess their impact on model output. However, without a theoretical foundation, these ad-hoc designed objectives are often specific to a single domain and do not generalize to wider scenarios.

Recently, the principle of information bottleneck (IB) has been used for explainable learning (Tishby & Zaslavsky, 2015). Specifically, given a time series instance X and its label Y, the IB principle finds an *explanation* sub-instance X' which optimizes a tradeoff between compactness and informativeness, where compactness is achieved by minimizing the mutual information between the original instance X and the sub-instance X', and informativeness is achieved by maximizing the mutual information between the sub-instance X' and instance label Y. That is, X' minimizes $I(X;X') - \alpha I(X';Y)$, where the hyperparameter $\alpha > 0$ captures the trade-off between compactness and informativeness of X' (Miao et al., 2022). Thus, IB has been used extensively in domains including computer vision (Luo et al., 2019), natural language (Mahabadi et al., 2021; Liu et al., 2024a), and graph-structured data (Miao et al., 2022).

The application of the IB principle in the context of time series explainability faces several limitations. First, due to the potentially large size of the time series data X, accurate estimation of the mutual information term $I(X;X')$ may require significant amounts of training data (e.g., (McAllester & Stratos, 2020)). Second, as shown in previous studies (Huang et al., 2024), the IB loss function is sometimes minimized by sub-instances that do not align with the intuitive notion of explainability, leading to perceptually unrealistic explanations. This latter issue, which is referred to as the *signaling* issue, arises when an explanation rule produces sub-instances X' which signal the value of Y, e.g., by taking very large values when $Y = 1$ and small values when $Y = 0$, without necessarily aligning with the notion of explainability. Such sub-instances have high $I(X';Y)$ due to the high correlation between X' and Y and low $I(X;X')$ due to the fact that X' only signals the value of Y, however, they do not provide an intuitively valid explanation despite optimizing the IB loss function.

An alternative approach to evaluating the suitability of explanations is to apply the classifier $f(\cdot)$ directly to the sub-instance X' and evaluate a cross-entropy term $\mathrm{CE}(Y;Y')$, where $Y' = f(X')$. This resolves the aforementioned signaling issue and is used by existing methods (Enguehard, 2023; Liu et al., 2024b; Queen et al., 2023) hereinafter called reference models. However, X' is often out-of-distribution for $f(\cdot)$ thus leading to inaccurate predictions for Y' (Zhao et al., 2022). To elaborate, as shown in Figure 1, let us take TIMEX (Queen et al., 2023) as an example (other reference models face a similar problem), where the distribution of original instances and their sub-instances is substantially different from each other. Then, since $f(\cdot)$ is trained on the original instances, it would produce inaccurate results when applied to the explanation sub-instances. Consequently, TIMEX retrains an additional white-box model $f^E(\cdot)$ for model consistency, while explanation instances (generated by the reference models) input to $f(\cdot)$ remain unreliable. Such replication requires knowledge of the to-be-explained model structure, and consistency in the behavior of the white-box model may not equal consistency in explanation.

To address these challenges, we develop a practical objective function for time series explainable learning. Specifically, we replace the compactness quantifier $I(X;X')$ in IB with a traceable variational upper-bound, which consists of a minimality constraint and a discrete constraint. To avoid the signaling issue, we replace the informativeness quantifier $I(X';Y)$ with a measure of label consistency between Y' and Y, where Y' is the label of sub-instance X'. We propose a novel explainable framework TIMEX++ that generates in-distributed and label-preserving time series instances for both quantifiers. TIMEX++ maintains the new instances consistent with the original distribution (see Figure 1) and preserves the uninformative areas provided by the reference model. We summarize our contributions as follows

- We investigate the limitations of existing explanation models for time series learning from the perspective of information theory and propose a practical objective function.
- We propose a novel explanation framework TIMEX++, which addresses the distribution shifting issue by generating in-distributed and explanation-embedded instances.
- We achieve state-of-the-art performances compared to other explainers on eight synthetic and real-world time series datasets, and verify the effectiveness in a real-world application from environmental science.

2. Notations and Preliminary

2.1. Notations and Problem Formulation

This work focuses on explainability in time series classification. A time series instance $X \in \mathcal{X} = \mathbb{R}^{T \times D}$ is represented by a $T \times D$ real-valued matrix, where T is the length of the time series, and D is the feature dimension. A multivariate time series is one for which $D > 1$, otherwise, the time series is called univariate. The value of the feature indexed d at time t is denoted by $X[t, d]$. A training set $\mathcal{T} = \{(X_i, Y_i)|i \in [N]\}$ consists of N time series instances X_i along with their associated labels Y_i, where $Y_i \in \mathcal{C}$ and $\mathcal{C} = \{1, 2, \cdots, |\mathcal{C}|\}$ is the set of all possible labels. We use the shorthand $\boldsymbol{X} = \{X_i\}_{i=1}^{N}$ to refer to the instances without labels. A time series classifier $f(\cdot)$ takes an instance $X \in \mathbb{R}^{T \times D}$ as input, and outputs a label $f(X) \in \mathcal{C}$.

In order to develop a generally applicable model for explainability, we consider explanation methods which are task-agnostic and treat the to-be-explained model $f(\cdot)$ as a black box, i.e., the so-called *post-hoc, instance-level* explanation methods (Zhang et al., 2021). In this context, an explanation refers to a sub-instance of the input time series, extracted using a saliency mask, which is a 'sufficient statistic' of the input with respect to its label. Furthermore, we consider model explainability, rather than task explainability. That is, the extracted sub-instance must be a sufficient statistic with respect to the model output, rather than the ground-truth label (Faber et al., 2021; Liu et al., 2024b). The high-level problem statement is given as follows.

Problem 1 (Post-hoc Instance-level Time Series Explanation)**.** *Given a trained model f and input $X \in \mathcal{X} = \mathbb{R}^{T\times D}$, the objective in post-hoc instance-level time series explanation is to find a sub-instance X' that 'explains' the prediction of f on X. The sub-instance X' is obtained by applying a binary mask $M \in \mathcal{M} = \{0,1\}^{T\times D}$ on X, i.e., $X' = X \odot M$, where $\odot$ is the element-wise multiplication.*

As can be observed in the problem statement, in order to find good post-hoc instance-level time series explanations, given an observed instance X, one needs to optimize the choice of binary mask $M \in \{0,1\}^{T\times D}$ with respect to an underlying objective function, e.g., the information bottleneck objective function discussed in the subsequent sections. In this work, we transform this discrete optimization problem into a continuous one, and consider stochastic masks. That is, we define an explanation extractor $g(\cdot)$ as a function that takes the instance X as input, and outputs a matrix $\boldsymbol{\pi} = [\pi_{t,d}]_{t\in[T],d\in[D]} \in [0,1]^{T\times D}$. Then, the binary mask is generated by producing each $M[t,d]$ independently and according to a Bernoulli distribution with parameter $\pi_{t,d}$.

2.2. The Information Bottleneck Principle

The IB principle (Tishby et al., 1999) has been widely used in the explainability literature. Formally, given an input instance X with label Y, the IB principle obtains a compact and informative sub-instance X' of X using the following optimization:

$$\max_{\substack{g:\mathcal{X}\mapsto[0,1]^{T\times D}\\ M[t,d]\sim \mathrm{Bern}(\pi_{t,d})}} I(Y;X') - \alpha I(X;X'), \tag{1}$$

where $X' = X \odot M$, $[M_{t,d}]_{t\in[T],d\in[D]}$ are generated independently and according to a Bernoulli distribution with parameter $\pi_{t,d}$, $g(X) = \boldsymbol{\pi} = [\pi_{t,d}]_{t\in[T],d\in[D]}$, and α is a hyperparameter capturing the trade-off between informativeness quantified by $I(Y;X')$ and compactness quantified by $I(X;X')$. The IB principle has been used to obtain several explanation mechanisms (Mahabadi et al., 2021; Wu et al., 2020). However, it has been pointed out that IB-based approaches suffer from a *signaling* issue which may lead to explanation outputs that do not align with the notion of explainability. That is, the value of X' can *signal* the value of Y, yielding a large $I(X;Y)$ and small $I(X;X')$, without necessarily aligning with the intuitive notion of explainability. The interested reader is referred to Appendix B for an illustrating example of the signaling issue. To avoid this issue, the IB optimization is modified as follows:

$$\min_{\substack{g:\mathcal{X}\mapsto[0,1]^{T\times D}\\ M[t,d]\sim \mathrm{Bern}(\pi_{t,d})}} -\mathrm{LC}(Y;Y') + \alpha I(X;X'), \tag{2}$$

where $\mathrm{LC}(Y;Y')$ denotes the label consistency (LC) between Y and the label of X', denoted by Y', which is discussed in detail in the subsequent sections. It can be noted that the signaling issue, described through an example in Appendix B, is resolved by this modification.

3. Explaining Time Series Learning via Information Bottleneck

As discussed in the prequel, the application of the IB principle to explainability problems suffers from the signaling issue. Additionally, direct application of the modified IB principle in Eq. (2) in time series explanation problems is challenging due to the complex and high-dimensional nature of the data (Goldfeld & Polyanskiy, 2020). Specifically, the temporal dependencies and varying feature dynamics inherent in time series complicate the accurate estimation of mutual information, a key component of the IB framework, thus rendering the direct application of IB computationally intractable in this context. In the following, we identify several additional issues in applying the IB principle, and derive a tractable objective to address these issues.

Modifying the Compactness Quantifier $I(X;X')$. As mentioned in the previous sections, the term $I(X;X')$ is included in IB to ensure the compactness of the resulting explanation sub-instance. That is, to ensure that the explanation contains as few features and time-instances as needed. However, in the context of time-series classification, the mutual information term $I(X;X')$ does not necessarily align with this objective, as it sometimes gives preference to multiple low-entropy sub-instances of X' as opposed to a single high-entropy sub-instance, thus violating compactness. An illustrating example of this issue is provided in Appendix C. To address this, we modify the IB optimization further, and consider the following objective function:

$$\min_{\substack{g:\mathcal{X}\mapsto[0,1]^{T\times D}\\ M[t,d]\sim \mathrm{Bern}(\pi_{t,d})}} -\mathrm{LC}(Y;Y') + \mathbb{E}_X[\alpha \sum_{t,d} H(M[t,d]) + \gamma|M|], \tag{3}$$

where $\alpha, \gamma > 0$ are hyperparameters and $H(\cdot)$ is the entropy, We provide the following justification for this modified objective function. The term $I(X;X')$ was included in the original IB formulation to ensure the compactness of

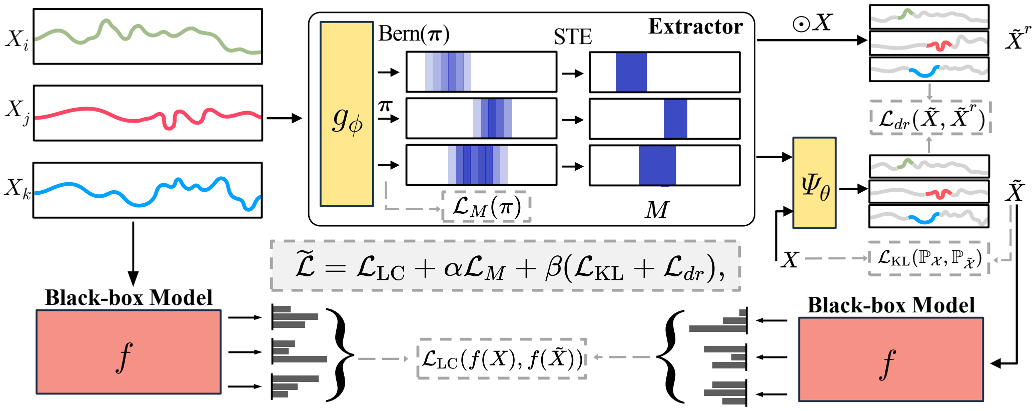

X_i
X_j
X_k
g_ϕ
Bern(π)
π
STE
Extractor
$\mathcal{L}_M(\pi)$
M
$\odot X$
$\widetilde{X}^r$
$\mathcal{L}_{dr}(\tilde{X}, \widetilde{X}^r)$
Ψ_θ
$\tilde{X}$
X
$\mathcal{L}_{\mathrm{KL}}(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\tilde{\mathcal{X}}})$
$\widetilde{\mathcal{L}} = \mathcal{L}_{\mathrm{LC}} + \alpha\mathcal{L}_M + \beta(\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{dr}),$
Black-box Model
f
$\mathcal{L}_{\mathrm{LC}}(f(X), f(\tilde{X}))$
Black-box Model
f

Figure 2. The overall architecture of TIMEX++, which consists of the explanation extractor and the conditioner generating new instances.

the resulting sub-instance explanation. Minimizing $\mathbb{E}(|M|)$ serves a similar purpose, by limiting the average number of non-zero elements in the mask M, while avoiding the aforementioned problem which arises due to high-entropy components of the time series sequence in optimizing the original IB objective. The inclusion of the term $\sum_{t,d} H(M[t,d])$ ensures that the mask is 'almost' deterministic, i.e., $\pi_{t,d} \approx 0$ or $\pi_{t,d} \approx 1$.

The objective function in Eq. (3) can be further simplified in practice. Particularly let us take a hyperparameter $r \in [0,1]$ and define $\mathbb{Q}(M[t,d]) \sim \mathrm{Bern}(r)$ and $\mathbb{Q}(M) = \prod_{t,d} \mathbb{Q}(M[t,d])$. Then, as shown in Appendix D, Eq. (3) can be reformulated as:

$$\begin{aligned} \min_{\substack{g:\mathcal{X}\mapsto[0,1]^{T\times D} \\ M[t,d]\sim\mathrm{Bern}(\pi_{t,d})}} \quad & -\mathrm{LC}(Y;Y') \\ & + \alpha\mathbb{E}_X\left[D_{\mathrm{KL}}(\mathbb{P}(M|X)\|\mathbb{Q}(M))\right], \end{aligned} \tag{4}$$

where we have defined $\mathbb{P}(M|X) = \prod_{t,d} \mathbb{P}(M[t,d]|X)$, and for a given instance X, $\mathbb{P}(M[t,d]|X)$ is a Bernoulli distribution with parameter $\pi_{t,d}$.

The Informativeness Quantifier $\mathrm{LC}(Y;Y')$**.** In the modified IB formulation in Eq. (4), the infromativeness of explanations is quantified by $\mathrm{LC}(Y;Y')$ which measures the cross entropy between the original label Y and the label Y' generated based on the explanation sub-instance X'. Ideally, one would produce Y' by directly applying $f(\cdot)$ on X'. However, this approach suffers from the out-of-distribution (OOD) issue as described in the following. The sub-instance X' is not necessarily in-distribution for $f(\cdot)$ and hence directly applying $f(\cdot)$ to X' may not yield an accurate estimate of the correct label. That is, while X' is a sub-instance of X, and it can be observed as part of X in the training data, it may not be observed in isolation as an element of the training set. Thus, $f(\cdot)$ is not trained to classify X' directly, and will not necessarily yield accurate predictions if applied to this input.

A routinely adopted method for producing Y' is to 'pad' the sub-instance X' using additional Gaussian noise variables to form a time series instance $\widetilde{X}^r$, and then setting $Y' = f(\widetilde{X}^r)$ (e.g., Fong & Vedaldi (2017); Enguehard (2023)). Formally, we consider an approximate baseline distribution: $\mathbb{B}_{\mathcal{X}} = \Pi_{t,d}\mathcal{N}(\mu_{t,d}, \sigma^2_{t,d})$, where $\mu_{t,d}, \sigma^2_{t,d}$ are the mean and variance over the whole time series samples $\boldsymbol{X}$. Then, the 'padded' instance is given by:

$$\widetilde{X}^r := M \odot X + (1-M) \odot b, \text{ where } b \sim \mathbb{B}_{\mathcal{X}}. \tag{5}$$

The justification is that while X' is OOD with respect to $f(\cdot)$, it might be the case that using an appropriate choice of padded input parameters $\mu_{t,d}, \sigma^2_{t,d}$, $\widetilde{X}^r$ may be made in-distribution for $f(\cdot)$.

In practice, although transforming X' to $\widetilde{X}^r$ using the padding approach alleviates the OOD issue to some extent, it does not completely mitigate the problem. An alternative method has been introduced recently to address the OOD issue in TIMEX (Queen et al., 2023). In that approach, first, a copy of the classifier $f(\cdot)$, denoted by $f^E(\cdot)$ is constructed. Then, to address the OOD challenge, TIMEX fine-tunes the parameters of $f^E(\cdot)$ with a consistent loss between $f(X)$ and $f^E(\widetilde{X}^r)$, so that f^E is trained on instances of $\widetilde{X}^r$ and $f(\cdot)$ on instances of X. However, this state-of-the-art method suffers from two main drawbacks. First, TIMEX has to treat the to-be-explained model as a white box whose architecture and model parameters are accessible (see Figure 1). However, in a wide range of real-world scenarios, the classifier model is given in a black-box manner. Second, similar to the original IB, a cross-entropy loss defined between Y, the ground-truth label, and Y' the output of $f^E(\widetilde{X}^r)$ suffers from the signaling issue. That is, $\widetilde{X}^r$ can signal the value of Y and f^E may be trained to detect the signal, thus yielding explanations that do not align with the intuitive notion of explainability (please refer to the example in Appendix B which explains the signaling problem in detail).

To tackle these limitations, we propose an additional step in generating in-distributed instances from X'. That is, we first generate $\widetilde{X}^r$ using the standard padding technique described previously. Then, we take $\widetilde{X}^r$ as a *reference instance* and generate a new *explanation-embedded instance* $\widetilde{X} \in \widetilde{\mathcal{X}} = \mathbb{R}^{T \times D}$ by minimizing two loss functions: i) a loss function $\mathcal{L}_{\mathrm{KL}}(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\widetilde{\mathcal{X}}})$, quantifying the distribution shift between $\widetilde{X}$ and X, and ii) a loss function $\mathcal{L}_{dr}(\widetilde{X}, \widetilde{X}^r)$, quantifying Euclidean distance between $\widetilde{X}^r$ and $\widetilde{X}$. The exact formulation of each of these loss functions is discussed in detail in the following sections. The first loss function ensures that $\widetilde{X}$ is in-distribution for $f(\cdot)$, while the second loss function builds upon the previously adopted methods mentioned in the prequel, and by forcing $\widetilde{X}$ to have a distribution that is 'close' to a Gaussian distribution, ensures a general solution without overfitting. Thus, to maximize $\mathrm{LC}(Y; Y')$, the total loss function for the informativeness of the explanation is equal to:

$$\mathcal{L}_{\mathrm{LC}}(f(X), f(\widetilde{X})) + \beta(\mathcal{L}_{\mathrm{KL}}(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\widetilde{\mathcal{X}}}) + \mathcal{L}_{dr}(\widetilde{X}, \widetilde{X}^r)). \tag{6}$$

4. TIMEX++ Method

This section describes the TIMEX++ approach, which builds upon the ideas set forth in the previous sections. The architecture is depicted in Figure 2. Specifically, we introduce an explanation extractor g_ϕ to learn based on the compactness quantifier of Eq. (4). Next, an explanation conditioner Ψ_θ controls the generation of explanation-embedded instances, which must generate in-distribution instances $\widetilde{X}$ using the loss in Eq. (6). TIMEX++ learns label consistency for reliable approximation of $\mathrm{LC}(Y; Y')$. Each of these components is detailed in the following.

Explanation Extractor g_ϕ. The extractor $g_\phi : \mathbb{R}^{T \times D} \mapsto [0, 1]^{T \times D}$ encodes the input X into the stochastic mask $\boldsymbol{\pi}$. We implement the compactness quantifier in Eq. (4) by parameterizing $g_\phi(\cdot)$ via an encoder-decoder transformer, in which $\mathbb{P}(M|X)$ is represented by $g_\phi(\cdot)$. This parameterization can be understood as a way to assign attribution scores such that a low $\pi_{t,d}$ has a low probability of being masked-in. Furthermore, to penalize irregular non-contiguous shapes in a time series sample, TIMEX++ also optimizes a connective loss $\mathcal{L}_{\mathrm{con}}$ for the predicted distributions:

$$\mathcal{L}_{\mathrm{con}} = \lambda_{\mathrm{con}} \frac{1}{T \times D} \sum_{d=1}^{D} \sum_{t=1}^{T-1} \sqrt{(\pi_{t,d} - \pi_{t+1,d})^2}, \tag{7}$$

where λ_{con} is the penalization strength. Further, as discussed in Queen et al. (2023), the explanation generating stochastic mask should be 'hardened' via mapping to a deterministic binary mask. Thus, we adopt a straight-through estimator (STE) (Jang et al., 2017) to obtain a binary mask $M \in \{0, 1\}^{T \times D}$, i.e, $M := \mathrm{STE}(M)$. As discussed in Appendix D, the loss function of masks binding Eq. (7) when training $g_\phi(\cdot)$ can be written as:

$$\begin{aligned} \mathcal{L}_M = & \sum_{t,d} [\pi_{t,d} \log(\frac{\pi_{t,d}}{r}) \\ & + (1 - \pi_{t,d}) \log(\frac{1 - \pi_{t,d}}{1 - r})] + \mathcal{L}_{\mathrm{con}}. \end{aligned} \tag{8}$$

Explanation Conditioner Ψ_θ. A key idea in the design of TIMEX++ is to use a two-step process to generate the instance $\widetilde{X}$ which in turn is used to quantify informativeness. That is, to first generate the reference instance $\widetilde{X}^r$ using the conventional Gaussian padding method described in the prequel, and then to generate $\widetilde{X}$ to mitigate the OOD issue. We directly $\widetilde{X}$ learn via a parameterized conditioner Ψ_θ so that:

$$\widetilde{X} = \Psi_\theta(M, X), \tag{9}$$

where Ψ_θ is an MLP that maps the concatenation $[M, X]$ into $\widetilde{X} \in \widetilde{\mathcal{X}} = \mathbb{R}^{T \times D}$. To ensure the in-distribution property, we minimize the KL divergence:

$$\mathcal{L}_{\mathrm{KL}}(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\tilde{\mathcal{X}}}) = \mathbb{E}\left[D_{\mathrm{KL}}\left(\mathbb{P}_{\mathcal{X}}(X) \| \mathbb{P}_{\tilde{\mathcal{X}}}(\tilde{X})\right)\right]. \tag{10}$$

As mentioned previously, to ensure generalizability and avoid overfitting, we minimize the Euclidean distance between $\widetilde{X}$ and the Gaussian padded reference $\widetilde{X}^r$ using:

$$\mathcal{L}_{dr}(\widetilde{X}, \widetilde{X}^r) = \frac{1}{T \times D} \sum_{d=1}^{D} \sum_{t=1}^{T} ||\widetilde{X}[t, d] - \widetilde{X}^r[t, d]||^2. \tag{11}$$

Label Consistency. To maintain a similar amount of label in $\mathrm{LC}(Y; Y')$, we use the preservation game (Fong & Vedaldi, 2017) that minimizes the deviation of predictions from the original ones by the explained black-box $f(\cdot)$. We use the same Jensen-Shannon (JS) divergence as TIMEX for the label consistency loss $\mathcal{L}_{\mathrm{LC}}$ in Eq. (6) by:

$$\mathcal{L}_{\mathrm{LC}}(f(X), f(\widetilde{X})) = \mathbb{E}\left[D_{\mathrm{JS}}(f(X) \| f(\widetilde{X}))\right]. \tag{12}$$

Overall Learning Objective. The learning objective is to train the whole framework by minimizing the total loss:

$$\widetilde{\mathcal{L}} = \mathcal{L}_{\mathrm{LC}} + \alpha \mathcal{L}_M + \beta(\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{dr}), \tag{13}$$

where $\{\alpha, \beta\} \in \mathbb{R}$ are hyperparameters adjusting the weight of losses. TIMEX++ is optimized end-to-end, requiring little hyperparameter tuning. The choice of r for regularized masks is still crucial and proved to be stable (Miao et al., 2022). Hence, we set $r = 0.5$ to remain consistent throughout experiments, which is analyzed in Appendix J. We summarize the pseudo-code of TIMEX++ in Appendix E.

Table 1. Attribution explanation performance on univariate and multivariate synthetic datasets.

	FREQSHAPES			SEQCOMB-UV		
METHOD	AUPRC	AUP	AUR	AUPRC	AUP	AUR
IG	0.7516±0.0032	0.6912±0.0028	0.5975±0.0020	0.5760±0.0022	0.8157±0.0023	0.2868±0.0023
DYNAMASK	0.2201±0.0013	0.2952±0.0037	0.5037±0.0015	0.4421±0.0016	0.8782±0.0039	0.1029±0.0007
WINIT	0.5071±0.0021	0.5546±0.0026	0.4557±0.0016	0.4568±0.0017	0.7872±0.0027	0.2253±0.0016
CORTX	0.6978±0.0156	0.4938±0.0004	0.3261±0.0012	0.5643±0.0024	0.8241±0.0025	0.1749±0.0007
SGT + GRAD	0.5312±0.0019	0.4138±0.0011	0.3931±0.0015	0.5731±0.0021	0.7828±0.0013	0.2136±0.0008
TIMEX	0.8324±0.0034	0.7219±0.0031	0.6381±0.0022	0.7124±0.0017	**0.9411**±0.0006	0.3380±0.0014
TIMEX++	**0.8905**±0.0018	**0.7805**±0.0014	**0.6618**±0.0019	**0.8468**±0.0014	0.9069±0.0003	**0.4064**±0.0011

	SEQCOMB-MV			LOWVAR		
METHOD	AUPRC	AUP	AUR	AUPRC	AUP	AUR
IG	0.3298±0.0015	0.7483±0.0027	0.2581±0.0028	0.8691±0.0035	0.4827±0.0029	0.8165±0.0016
DYNAMASK	0.3136±0.0019	0.5481±0.0053	0.1953±0.0025	0.1391±0.0012	0.1640±0.0028	0.2106±0.0018
WINIT	0.2809±0.0018	0.7594±0.0024	0.2077±0.0021	0.1667±0.0015	0.1140±0.0022	0.3842±0.0017
CORTX	0.3629±0.0021	0.5625±0.0006	0.3457±0.0017	0.4983±0.0014	0.3281±0.0027	0.4711±0.0013
SGT + GRAD	0.4893±0.0005	0.4970±0.0005	**0.4289**±0.0018	0.3449±0.0010	0.2133±0.0029	0.3528±0.0015
TIMEX	0.6878±0.0021	0.8326±0.0008	0.3872±0.0015	0.8673±0.0033	0.5451±0.0028	**0.9004**±0.0024
TIMEX++	**0.7589**±0.0014	**0.8783**±0.0007	0.3906±0.0011	**0.9466**±0.0015	**0.8057**±0.0016	0.8332±0.0016

Table 2. Difference between the distribution of different explanation instances and the distribution of original data.

	FREQSHAPES			SEQCOMB-UV		
METHOD	KDE ↑	KL-DIVERGENCE ↓	MMD	KDE ↑	KL-DIVERGENCE ↓	MMD ↓
ZERO	-36.6705± 0.2747	0.1440±0.0069	0.0769±0.0044	-90.0931± 0.3115	0.3985±0.0048	0.1668±0.0032
MEAN	-36.5394±0.1942	0.1465±0.0034	0.0777±0.0063	-83.0381±0.2708	0.3249±0.0089	0.1112±0.0047
$b \sim \mathbb{B}_{\mathcal{X}}$	-53.8552±0.5086	0.2888±0.0032	0.0240±0.0004	-100.0496±0.6577	0.4981±0.0045	**0.0085**±0.0004
OURS	**-28.7566**±2.6582	**0.0964**±0.0239	**0.0162**±0.0037	**-57.9323**±1.6837	**0.0598**±0.0176	0.0777±0.0186

	SEQCOMB-MV			LOWVAR		
METHOD	KDE ↑	KL-DIVERGENCE ↓	MMD ↓	KDE ↑	KL-DIVERGENCE ↓	MMD ↓
ZERO	-257.0395±1.6558	0.7394±0.0290	0.2476±0.0063	-431.9932±1.1306	0.5972±0.0116	0.1049±0.0029
MEAN	-230.2502±1.3131	0.4549±0.0192	0.0741±0.0089	-429.3312±1.2488	0.5710±0.0118	0.0953±0.0035
$b \sim \mathbb{B}_{\mathcal{X}}$	-260.1821±1.5705	0.7213±0.0188	0.0261±0.0002	-474.4819±1.3187	0.9763±0.0166	0.1041±0.0001
OURS	**-191.2647**±0.8897	**0.0377**±0.0099	**0.0141**±0.0116	**-426.4307**±2.7648	**0.5380**±0.0251	**0.0725**±0.0172

5. Experiments

In this section, we evaluate the quality of our explanations on four synthetic datasets and six real-world datasets. We employ a Transformer (Vaswani et al., 2017) classifier as the black-box model f to explain, where the hyperparameters are optimized to ensure model performance. In each evaluation metric, we mark **bold** as the best and underline as the second best. All reported results for our method, baselines, and ablations are presented as mean ± std from 5 fold cross-validation. Additionally, we evaluate the feasibility of our approach by conducting a case study involving real-world flood prediction. The details of each dataset and black-box model are provided in Appendix F.

5.1. Feature Importance for Synthetic Datasets

Datasets and Benchmarks. We first conduct experiments on four datasets with known ground-truth explanations: **FreqShapes**, **SeqComb-UV**, **SeqComb-MV**, and **LowVar**. These datasets are meticulously curated to encapsulate a wide array of temporal dynamics within both univariate and multivariate settings. In this context, it is crucial to highlight the importance of identifying key features at different time steps. We compare our method with six explainability benchmarks, including integrated gradients (IG) (Sundararajan et al., 2017), Dynamask (Crabbé & Van Der Schaar, 2021), WinIT (Leung et al., 2023), CoRTX (Chuang et al., 2023), SGT + GRAD (Ismail et al., 2021), and TIMEX (Queen et al., 2023). The implementation details of all algorithms are available in Appendix G.

Metrics. Given that the precise salient features are known, we utilize them as the ground truth for evaluating explanations. At each time step, features causing prediction label changes are attributed an explanation of 1, whereas those that do not effect such changes are 0. Following Crabbé & Van Der Schaar (2021), we evaluate the quality of explanations with area under precision (AUP) and area under recall (AUR). We also employ AUPRC for consistency with Queen et al. (2023), which combines the results from both AUP and AUR. For all evaluated metrics higher values are better, and their detailed definitions are referred to the Appendix F.3.

Results. Table 1 summarizes the performance results of the above explainers on univariate and multivariate datasets. TIMEX++ outperforms other explainers on 9 out of 12 cases (across 3 metrics on four datasets), achieving an average improvement in explanation AUPRC of 11.01%, AUP of 10.87%, and AUR of 1.25% when compared to the strongest baseline TIMEX. Note that under all these datasets, AUP is more valuable than AUR because the predicted signal has re-

Table 3. (*Left*) Attribution explanation performance on the ECG dataset. (*Right*) Results of ablation analysis.

METHOD	ECG AUPRC	AUP	AUR	TIMEX++ ABLATIONS	ECG AUPRC	AUP	AUR
IG	0.4182±0.0014	0.5949±0.0023	0.3204±0.0012	FULL	**0.6599**±0.0009	0.7260±0.0010	0.4595±0.0007
DYNAMASK	0.3280±0.0011	0.5249±0.0030	0.1082±0.0080	W/O STE	0.6152±0.0007	**0.7468**±0.0008	0.4023±0.0012
WINIT	0.3049±0.0011	0.4431±0.0026	0.3474±0.0011	W/O $\mathcal{L}_{\text{LC}}$	0.6209±0.0019	0.6417±0.0020	0.4287±0.0015
CORTX	0.3735±0.0008	0.4968±0.0021	0.3031±0.0009	W/O $\mathcal{L}_{\text{KL}}$	0.6417±0.0019	0.6979±0.0009	0.4424±0.0007
SGT + GRAD	0.3144±0.0010	0.4241±0.0024	0.2639±0.0013	W/O $\mathcal{L}_{dr}$	0.1516±0.0003	0.1405±0.0003	**0.6313**±0.0006
TIMEX	0.4721±0.0018	0.5663±0.0025	0.4457±0.0018	W/O $\mathcal{L}_{\text{con}}$	0.6072±0.0008	0.6921±0.0010	0.4387±0.0007
TIMEX++	**0.6599**±0.0009	**0.7260**±0.0010	**0.4595**±0.0007				

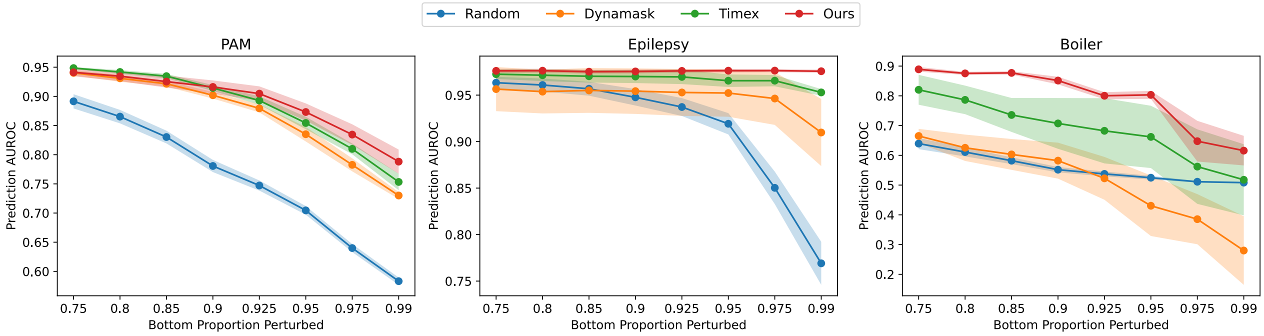


Figure 3. Occlusion experiments on real-world datasets. Higher values indicate better performance.

dundant information. We analyze the statistical significance of multiple methods using the Friedman Test, resulting in the statistic $F_F = 51.32$ and $p < 0.001$ for all methods on 12 cases. This suggests that there are significant differences in the results among the different methods. Specifically, when considering the global metric AUPRC, TIMEX++ significantly improves ground-truth explanation 6.97% on FreqShapes, 18.86% on SeqComb-UV, 10.33% on SeqComb-MV, and 8.91% on LowVar over strongest baselines. It matches our claim that our method provides explanation instances without hurting the predictions trained via the IB principle. We also provide visualizations of salient subsequences in Appendix L.

Distribution Analysis of Perturbations. To verify that the explanation instances $\widetilde{X}$ and $\widetilde{X}^r$ are within the distribution of original datasets, we utilize kernel density estimation[1] (KDE) to assess the log-likelihood of each explanation instance under the original distribution, approximating zero indicates a higher likelihood of explanation instances originating from the original distribution. We also quantify the KL divergence and maximum mean discrepancy (MMD) between the distribution of original instances and explanation instances, where a smaller value means a greater similarity between the two distributions. For $\widetilde{X}^r$, we pick three baselines in Eq. (5): including b is Zero, b is Mean of all $\boldsymbol{X}$, and $b \sim \mathbb{B}_{\mathcal{X}}$. We perform experiments on synthetic datasets and provide the results in Table 2. The findings reveal that the explanation instance by our TIMEX++ more closely resembles the original distribution. It significantly reduces the probability of OOD relative to the strongest baseline, which finding is consistent with the visualization depicted in Figure 1. It indicates that our reliable approximation creates in-distributed and explanation-embedded instances. We also present the distribution of explanation embeddings over several datasets, which are shown in Appendix H.

5.2. Feature Importance for Real-world Datasets

Datasets and Benchmarks. We similarly evaluate our method on four datasets from real-world time series classification tasks: **ECG**, **PAM**, **Epilepsy**, and **Boiler**. Extracting wave intervals by Queen et al. (2023), the ground-truth explanations of ECG are defined as the QRS interval, where arrhythmias can be detected. Thus it can be used as a real-world evaluation, while other datasets without ground-truth explanations use occlusion experiments (Tonekaboni et al., 2020; Crabbé & Van Der Schaar, 2021; Liu et al., 2024b). Besides, we further expand the occlusion experiments with two datasets **Water** and **Freezer**, which are typical time series data in the UCR archive (Dau et al., 2019). For benchmarks and metrics, we maintain consistency with the synthetic experiments previously employed.

Results and Ablation Study on ECG Data. The performance results on ECG arrhythmia detection are presented in Tabel 3 left. We can see that TIMEX++ outperforms the leading baselines by 39.78% AUPRC, 22.04% AUP, and 3.10% AUR, suggesting that its performance in finding relevant QRS intervals drives the arrhythmia diagnosis. We

[1] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KernelDensity

Table 4. Performance report on different real-world datasets by masking the top 10% of salient features. The masked portion is substituted with an average of this feature or with zeros.

Method	Substitution	PAM	Epilepsy	Boiler	Wafer	Freezer	Rank
Random	Mean	0.9791±0.0018	0.9394±0.0045	0.8142±0.0555	0.9925±0.0023	0.7716±0.0932	7.2
	Zero	0.9794±0.0018	0.9395±0.0043	0.8950±0.0120	0.9920±0.0021	0.7747±0.0930	7.8
Dynamask	Mean	0.7776±0.0154	0.8155±0.0194	0.8080±0.0364	0.4878±0.2672	0.4515±0.1288	4.6
	Zero	0.7765±0.0154	0.3612±0.0725	0.5553±0.1459	0.4882±0.2671	**0.3550**±0.1100	3.6
TimeX	Mean	0.7494±0.0486	0.8518±0.0183	0.5630±0.0413	0.4633±0.1452	0.4693±0.0605	4.4
	Zero	0.7543±0.0493	0.4544±0.0771	0.4163±0.0476	0.4646±0.0574	0.4646±0.0574	3.6
TimeX++	Mean	**0.7172**±0.0194	0.8451±0.0291	0.3851±0.1229	0.4095±0.1502	0.3786±0.0738	2.8
	Zero	0.7751±0.0232	**0.3553**±0.1459	**0.3612**±0.0725	**0.3998**±0.0613	0.3771±0.0740	**2.0**

further explore ablation studies of our method in Table 3 right, where w/o means no related components. First, we show that the STE improves 7.27% in AUPRC and 14.22% in AUR, demonstrating the effectiveness of the STE. We also select the label consistency loss $\mathcal{L}_{\text{LC}}$, the maintenance loss $\mathcal{L}_{\text{KL}}$, the loss of the reference distance $\mathcal{L}_{dr}$, and the connective loss $\mathcal{L}_{\text{con}}$. Among them, $\mathcal{L}_{dr}$ failed to assess the explanation, resulting in low AUPRC and AUP. Moreover, the lack of other loss functions likewise produces poorer explanations compared to the base model. Our TimeX++ produces high-quality explanations, showing the value in including more intermediate states for optimizing the overall objective. For detailed ablation experiments and choice of parameters on other datasets, see Appendix J.

Occlusion Experiments on Real-world Datasets. Due to the absence of ground-truth explanations on real-world datasets, we occlude the bottom k-percentile of salient features to measure the change in prediction AUROC, as is done in Queen et al. (2023). Besides the above baselines, we also include a random explainer reference to control for potential misinterpretations. The explanation results by occluding salient features are presented in Figure 3. Our results show that TimeX++ outperforms others across both univariate (Epilepsy) and multivariate (PAM and Boiler) time series. Specifically, TimeX++ maintains non-decreasing performance on Epilepsy due to the retention of only salient features, and outperforms baselines at any threshold k on PAM and Boiler datasets. Moreover, our method maintains excellent stability compared to the strongest baseline TimeX, where the error bars of our method are narrower than it. This is because TimeX++ avoids the label leakage caused by re-optimizing a white-box predictive model. We also delete the top 10% of salient features and substitute them with an average of this feature or with zero perturbations (Liu et al., 2024b), and further expand our experiments to five real-world time series datasets. The results and average ranks of conducted experiments are shown in Table 4. As can be seen, the proposed method consistently outperforms existing explainers under different perturbations. Among them, substitution with zeros may be more applicable to the classification of black-boxes in these datasets, and therefore it has the highest average rank. We have similarly analyzed the significance of multiple methods using the Friedman Test, which yields that the statistic $F_F = 25.80$ and $p < 0.001$ for all methods on two substitutions. This shows that there is a significant difference between different methods, while TimeX++ performs the best. Overall, our approach generates effective explanations through information theory.

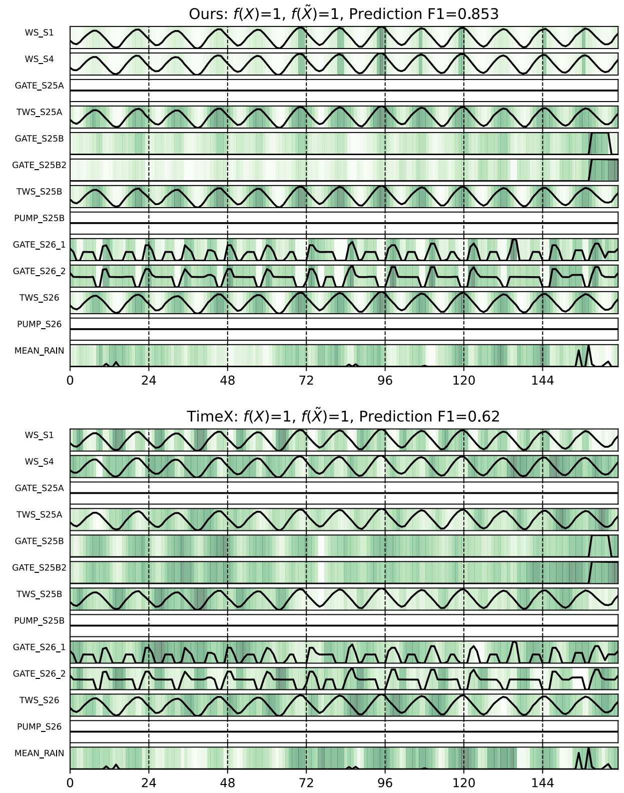


Figure 4. A visual comparison of the explainability of our method with the leading baseline on the Florida water dataset.

5.3. Case Study

Florida Water Dataset. To assess the efficacy of our approach, we conduct a real-world case study focusing on flood prediction in a Florida coastal system (Shi et al., 2023b). The detailed descriptions and a schematic diagram of the study area are illustrated in Appendix F.1. We choose to concentrate on one specific station (i.e., S1) to predict the probability of flooding in order to simplify the problem. We use a transformer-based predictor as a black-box model. Concurrently, we compare our model's explainability against the strongest baseline TIMEX, specifically examining feature importance at each time step.

Results. As shown in Figure 4, the first summary worth pointing out is that the F1 performance of our method outperforms that of the baseline, where the F1 metric on the top figure represents the performance of two models' explanation-embedded instance $\widetilde{X}$ in the black box. We break down our analysis of model explainability as follows: (i) Our TIMEX++ demonstrates the ability to accurately identify critical time points associated with elevated water levels, which often precede flooding. This is evident in the significance of features such as WS_S1, WS_S4, TWS_S25A, TWS_S25B, and TWS_S26. (ii) When it comes to hydraulic structures like gates and pumps situated along the river's course, which control water flow from upstream to the focal point S1, our model identifies the opening of these structures as a factor that heightens the risk of flooding at S1. Notably, TIMEX++ places greater emphasis on instances of higher openings while paying less attention to lower openings. This observation can be seen in features starting with 'GATE'. (iii) Additionally, our model effectively monitors rainfall events, recognizing their substantial influence on water levels within the river. In the context of the analysis above, the baseline TIMEX, exhibits sporadic performance, occasionally failing to capture significant time points or focusing on unnecessary ones.

6. Conclusion

In this work, we theoretically investigate an information-theoretic guided objective for learning time series explanations that ensure the compactness and informativeness of explained sub-instances. We propose a novel approach TIMEX++ based on the IB principle, which allows a traceability computation to produce in-distributed and label-preserving time series explanations. Comparative studies on synthetic and real-world datasets have confirmed that TIMEX++ surpasses existing explanatory tools in performance, with further proven practicality in environmental applications like flood prediction. Its effectiveness shows that TIMEX++'s capability to reflect complex behaviors of pre-trained time series classifiers accurately. However, generating sub-instances may involve some hyperparameters in the learning objective to control the quantifiers of the explanation, especially when dealing with different datasets. We minimally tuned the hyperparameters to obtain comparable results. Hence, it will be interesting to explore the salient areas by adopting a parameter-efficient tuning strategy.

Acknowledgements

This project was partially supported by NSF grants IIS-2331908 and CCF-2241057. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

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 of which we feel must be specifically highlighted here.

References

Adebayo, T. S., Awosusi, A. A., Kirikkaleli, D., Akinsola, G. D., and Mwamba, M. N. Can CO2 emissions and energy consumption determine the economic performance of south korea? A time series analysis. *Environmental Science and Pollution Research*, pp. 38969–38984, 2021.

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. Neural additive models: Interpretable machine learning with neural nets. In *NeurIPS*, pp. 4699–4711, 2021.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *ICLR*, 2017.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. E. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, pp. 061907, 2001.

Bento, J., Saleiro, P., Cruz, A. F., Figueiredo, M. A., and Bizarro, P. Timeshap: Explaining recurrent models through sequence perturbations. In *SIGKDD*, pp. 2565–2573, 2021.

Buhrmester, V., Münch, D., and Arens, M. Analysis of explainers of black box deep neural networks for computer vision: A survey. *Machine Learning and Knowledge Extraction*, 3(4):966–989, 2021.

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *NeurIPS*, pp. 3504–3512, 2016.

Chuang, Y.-N., Wang, G., Yang, F., Zhou, Q., Tripathi, P., Cai, X., and Hu, X. CoRTX: Contrastive framework for real-time explanation. In *ICLR*, pp. 1–23, 2023.

Crabbé, J. and Van Der Schaar, M. Explaining time series predictions with dynamic masks. In *ICML*, pp. 2166–2177, 2021.

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., and Sen, P. A survey of the state of explainable ai for natural language processing. In *IJCNLP-AACL*, pp. 447–459, 2020.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Enguehard, J. Learning perturbations to explain time series predictions. In *ICML*, pp. 9329–9342, 2023.

Faber, L., K. Moghaddam, A., and Wattenhofer, R. When comparing to ground truth is wrong: On evaluating gnn explanation methods. In *SIGKDD*, pp. 332–341, 2021.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *CVPR*, pp. 3429–3437, 2017.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Goldfeld, Z. and Polyanskiy, Y. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 19–38, 2020.

Guidotti, R. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pp. 1–55, 2022.

Huang, R., Shirani, F., and Luo, D. Factorized explainer for graph neural networks. In *AAAI*, 2024.

Ismail, A. A., Corrada Bravo, H., and Feizi, S. Improving deep learning interpretability by saliency guided training. In *NeurIPS*, pp. 26726–26739, 2021.

Jacovi, A. and Goldberg, Y. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *ACL*, pp. 4198–4205, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR*, pp. 1–12, 2017.

Kaushik, S., Choudhury, A., Sheron, P. K., Dasgupta, N., Natarajan, S., Pickett, L. A., and Dutt, V. AI in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in Big Data*, 3:4, 2020.

Leung, K. K., Rooke, C., Smith, J., Zuberi, S., and Volkovs, M. Temporal dependencies in feature importance for time series prediction. In *ICLR*, pp. 1–18, 2023.

Lin, H., Bai, R., Jia, W., Yang, X., and You, Y. Preserving dynamic attention for long-term spatial-temporal prediction. In *SIGKDD*, pp. 36–46, 2020.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

Liu, Z., Zhu, Y., and Chen, C. NA^{2}Q: Neural attention additive model for interpretable multi-agent Q-learning. In *ICML*, pp. 22539–22558, 2023.

Liu, Z., Wang, Z., Xu, L., Wang, J., Song, L., Wang, T., Chen, C., Cheng, W., and Bian, J. Protecting your llms with information bottleneck. *arXiv preprint arXiv:2404.13968*, 2024a.

Liu, Z., Zhang, Y., Wang, T., Wang, Z., Luo, D., Du, M., Wu, M., Wang, Y., Chen, C., Fan, L., and Wen, Q. Explaining time series via contrastive and locally sparse perturbations. In *ICLR*, pp. 1–21, 2024b.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *NeurIPS*, pp. 4765–4774, 2017.

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. Parameterized explainer for graph neural network. In *NeurIPS*, pp. 19620–19631, 2020.

Luo, D., Zhao, T., Cheng, W., Xu, D., Han, F., Yu, W., Liu, X., Chen, H., and Zhang, X. Towards inductive and efficient explanations for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Luo, Y., Liu, P., Guan, T., Yu, J., and Yang, Y. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *ICCV*, pp. 6778–6787, 2019.

Mahabadi, R. K., Belinkov, Y., and Henderson, J. Variational information bottleneck for effective low-resource fine-tuning. In *ICLR*, pp. 1–13, 2021.

McAllester, D. and Stratos, K. Formal limitations on the measurement of mutual information. In *AISTATS*, pp. 875–884, 2020.

Miao, S., Liu, M., and Li, P. Interpretable and generalizable graph learning via stochastic attention mechanism. In *ICML*, pp. 15524–15543, 2022.

Moody, G. B. and Mark, R. G. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20:45–50, 2001.

Queen, O., Hartvigsen, T., Koker, T., He, H., Tsiligkaridis, T., and Zitnik, M. Encoding time-series explanations through self-supervised model behavior consistency. In *NeurIPS*, 2023.

Razavi, S. Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling. *Environmental Modelling & Software*, 144:105159, 2021.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat, f. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

Reiss, A. and Stricker, D. Introducing a new benchmarked dataset for activity monitoring. In *ISWC*, pp. 108–109, 2012.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. In *SIGKDD*, pp. 1135–1144, 2016.

Shi, J., Stebliankin, V., and Narasimhan, G. The power of explainability in forecast-informed deep learning models for flood mitigation. *arXiv preprint arXiv:2310.19166*, 2023a.

Shi, J., Yin, Z., Myana, R., Ishtiaq, K., John, A., Obeysekera, J., Leon, A., and Narasimhan, G. Deep learning models for water stage predictions in south florida. *arXiv preprint arXiv:2306.15907*, 2023b.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *ICML*, pp. 3145–3153, 2017.

Silva, A., Gombolay, M., Killian, T., Jimenez, I., and Son, S.-H. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *AISTATS*, pp. 1855–1865, 2020.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *ICML*, pp. 3319–3328, 2017.

Suresh, H., Hunt, N., Johnson, A., Celi, L. A., Szolovits, P., and Ghassemi, M. Clinical intervention prediction and understanding with deep neural networks. In *MLHC*, pp. 322–337, 2017.

Theissler, A., Spinnato, F., Schlegel, U., and Guidotti, R. Explainable ai for time series classification: a review, taxonomy and research directions. *IEEE Access*, 10: 100700–100724, 2022.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop*, pp. 1–5, 2015.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. In *Allerton Conference on Communicationn, Control, and Computing*, pp. 368–377, 1999.

Tonekaboni, S., Joshi, S., Campbell, K., Duvenaud, D. K., and Goldenberg, A. What went wrong and when? Instance-wise feature importance for time-series blackbox models. In *NeurIPS*, pp. 799–809, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

West, P., Holtzman, A., Buys, J., and Choi, Y. Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *EMNLP-IJCNLP*, pp. 3752–3761, 2019.

Wu, T., Ren, H., Li, P., and Leskovec, J. Graph information bottleneck. In *NeurIPS*, pp. 20437–20448, 2020.

Zhang, Y., Tiňo, P., Leonardis, A., and Tang, K. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5): 726–742, 2021.

Zhao, B., Yu, S., Ma, W., Yu, M., Mei, S., Wang, A., He, J., Yuille, A., and Kortylewski, A. OOD-CV: A benchmark for robustness to out-of-distribution shifts of individual nuisances in natural images. In *ECCV*, pp. 163–180, 2022.

A. Related Works

Explainable Artificial Intelligence. With the ongoing advancement of neural networks, there is a burgeoning necessity to facilitate user comprehension of their operational mechanisms (Danilevsky et al., 2020). Predominantly, the corpus of research within explainable artificial intelligence (XAI) has been concentrated on the disciplines of computer vision (Linardatos et al., 2020; Buhrmester et al., 2021) and natural language processing (Danilevsky et al., 2020; Jacovi & Goldberg, 2020). With their success, recent studies have been extended to explain various data modalities, including reinforcement learning (Liu et al., 2023), graphs (Luo et al., 2020; Miao et al., 2022; Luo et al., 2024), and time series (Bento et al., 2021; Enguehard, 2023). The literature primarily focuses on *in-hoc* models (Agarwal et al., 2021; Silva et al., 2020) that learn intrinsic interpretability with some white-box models; and *post-hoc* explainability (Lundberg & Lee, 2017; Sundararajan et al., 2017) where explanations are provided for a trained model. Despite these efforts, the adoption of methods for time series analysis remains relatively limited as the importance and interrelation of features shift across multivariate sequences.

Information Bottleneck. IB (Tishby et al., 1999; Tishby & Zaslavsky, 2015), originally proposed for signal processing, has recently been adapted in different areas as it reduces length while preserving maximum information. On this foundation, Alemi et al. (2017) proposed a variational information bottleneck, which for the first time bridges the gap between deep learning and IB. Thus, it has been wildly employed in downstream applications like semantic segmentation (Luo et al., 2019), summarization (Mahabadi et al., 2021; West et al., 2019), defense against jailbreaks (Liu et al., 2024a), and graph learning (Miao et al., 2022; Huang et al., 2024). However, the IB principle is less researched on multivariate time series due to the intractability of mutual information.

Time-series Explainability. Recent literature (Enguehard, 2023; Bento et al., 2021) has probed into the domain of XAI with respect to multivariate time series. Within this scope, attention-based approaches (Lin et al., 2020; Choi et al., 2016) utilize attention mechanisms to generate significance scores that are inherently linked to the coefficients within the model. Gradient-based techniques (Shrikumar et al., 2017; Sundararajan et al., 2017) have employed the influence of localized modifications in input on the salience of features. Furthermore, perturbation-based methods, which are notably prevalent in time series analysis, typically modify the data via a baseline (Suresh et al., 2017), generative models (Tonekaboni et al., 2020; Leung et al., 2023), or by diminishing the informativeness of the data (Crabbé & Van Der Schaar, 2021; Liu et al., 2024b). Moreover, counterfactual-based explanations (Theissler et al., 2022; Guidotti, 2022) present minimal changes that would lead to a different classification by the model. Despite these advances, none of these methods have explored the problem of data shifting in generating unlabeled sub-instances. Different from them, our model generates in-distributed and label-preserving time series instances.

B. Example Illustrating the Signaling Problem in Applying the IB Princple

Let us consider a binary classification problem on a univariate time-series, where X_i are independent random variables taking values from $\{-1, 1\}$ with equal probability, and let $Y = \mathbb{1}(X_n > 0)$ be the indicator of the event that the n-th value in the time-series is positive, for some fixed $n \in \mathbb{N}$. Clearly, an intuitively 'good' explainer must output X_n as the explanation. However, this is not necessarily the output of the IB optimization as described in the following. Let us consider an explainer that outputs the maximum value of the instance X if $Y = 0$ and the minimum value if $Y = 1$, that is:

$$M[t,d] = \begin{cases} 1 & \text{if } (t,d) = \arg\max_{t',d'} X[t',d'] \text{ and } Y = 0 \\ 1 & \text{if } (t,d) = \arg\min_{t',d'} X[t',d'] \text{ and } Y = 1 \\ 0 & \text{otherwise} \end{cases} . \tag{14}$$

That is, the explainer 'signals' the value of Y by outputting large values (equal to 1 in this case) for X' if $Y = 0$ and small values (equal to -1 in this case), otherwise. Then, $I(X; X') \approx I(X; Y) = 1$, and this choice of explainer is an optimal solution for the IB optimization as $T \to \infty$. However, the explainer does not align with the intuitive notion of a 'good' explanation. The phenomenon was investigated in detail in (Huang et al., 2024) in the context of the explainability of graph neural networks.

C. Example Illustrating the Compactness Problem in Applying the IB Principle

Let us consider a univariate time-series classification scenario, where for some fixed even-valued integer n, we have:

$$X_i = \begin{cases} U_i, & \text{if } i < n \\ U_1 + U_2 + \cdots + U_i + N_i & \text{if } i \geq n \end{cases},$$

where U_i are independent binary symmetric random variables, i.e. $P(U_i = 1) = P(U_i = 0) = \frac{1}{2}$, and N_i are independent Gaussian, zero-mean, unit-variance random variables. Let $Y = \mathbb{1}(X_n > \frac{n}{2})$ be the indicator of the event that X_n has a value greater than $\frac{n}{2}$. Note that $P(Y = 0) = P(Y = 1) = \frac{1}{2}$. Clearly, a 'good' informative and compact explainer outputs $X' = X_n$ as the explanation, since Y is a function of X_n and is completely determined by its value. However, $I(X; X_n) = \infty$ since X_n is a continuous random variable and a function of X. So, the IB optimization yields $X_1, X_2, \cdots, X_{n-1}$ as the explanation rather than X_n, which is not a compact explanation. Generally, if a time instance $X[n, 1 : D]$ is highly informative, but has high entropy, it may not be chosen by the IB principle, since it yields large $I(X; X')$, although it is a compact and informative explanation.

D. Simplifying The Objective Function in Equation (3)

First, let us note that by linearity of expectation, we have:

$$\mathbb{E}_X \left[|M|\right] = \sum_{t,d} P(M[t,d] = 1) = \sum_{t,d} \mathbb{E}_X \left[\pi_{t,d}\right].$$

In our implementation, we set the value of $\frac{1}{T \times D}\mathbb{E}_X(|M|) = p$, and choose p as a hyperparameter, which controls the sparsity of the explanation. Consequently, the objective function becomes:

$$\min_{\substack{g:\mathbb{R}^{T\times D} \mapsto [0,1]^{T \times D} \\ M[t,d] \sim \text{Bern}(\pi_{t,d})}} -\text{LC}(Y; Y') + \alpha \mathbb{E}_X \sum_{t,d} H(M[t,d]) + \gamma T D p. \tag{15}$$

Let us define $r \in [0, 1]$ such that:

$$r \triangleq \frac{1 - \sqrt{1 - 2^{-\frac{\gamma p}{\alpha}+2}}}{2},$$

so that:

$$\frac{1}{\alpha}\gamma p = -\log r - \log 1 - r,$$

where we have taken γ such that $\frac{\gamma p}{\alpha} \geq 2$. Then,

$$\begin{aligned} &\mathbb{E}_X[\alpha \sum_{t,d} H(M[t,d]) + \gamma |M|] \\ &= \mathbb{E}_X \left[\alpha \sum_{t,d} \left(\pi_{t,d} \log \frac{\pi_{t,d}}{r} + (1 - \pi_{t,d}) \log \frac{1 - \pi_{t,d}}{1 - r} \right) \right]. \end{aligned} \tag{16}$$

Note that this resembles a KL-divergence term. To see this, let us define $\mathbb{Q}(M)$ as the Bernoulli distribution with parameter r, and $\mathbb{P}(M|X) = \prod_{t,d} \mathbb{P}(M[t,d]|X)$. Then, the modified objective function can be written as:

$$\min_{\substack{g:\mathbb{R}^{T\times D} \mapsto [0,1]^{T \times D} \\ M[t,d] \sim \text{Bern}(\pi_{t,d})}} -\text{LC}(Y; Y') + \alpha \mathbb{E}_X \left[D_{\text{KL}}(\mathbb{P}(M|X) \| \mathbb{Q}(M)) \right]. \tag{17}$$

E. Pseudo Code

Algorithm 1 The pseudo-code of TIMEX++

```
**Input:** A time series dataset $\mathcal{T} = \{(X_i, Y_i) | i \in [N]\}$, a trained black-box predictor $f : \mathcal{X} \mapsto \mathcal{C}$, adjusting hyperparameters $\{\alpha, \beta, r, \lambda_{\text{con}}\}$, total training epochs $E$, learning rate $\eta$
**Output:** Mask $\boldsymbol{M} = \{M_i\}_{i=1}^{N} \in \mathcal{M}$ to explain
**Training:**
Initialize a baseline distribution $\mathbb{B}_{\mathcal{X}} = \Pi_{t,d} \mathcal{N}(\mu_{t,d}, \sigma_{t,d}^2)$, where $\mu_{t,d}, \sigma_{t,d}^2$ are the mean and variance over $\mathcal{T}$
Initialize an explanation generator $g_\phi : \mathcal{X} \mapsto [0,1]^{T \times D}$, an explanation conditioner $\Psi_\theta : \{\mathcal{M}, \mathcal{X}\} \mapsto \widetilde{\mathcal{X}}$
**for** $e \leftarrow 1$ to $E$ **do**
  **for** $i \leftarrow 1$ to $N$ **do**
    Get $\boldsymbol{\pi} = g_\phi(X_i)$ and sample a mask $M_i \sim \mathbb{P}(M_i \mid X_i) = \prod_{t,d} \text{Bern}(\pi_{t,d})$
    Use a straight-through estimator STE to obtain the discrete mask $M_i \leftarrow \text{STE}(M_i)$
    Compute the reference instance $\widetilde{X}_i^r = M_i \odot X_i + (1 - M_i) \odot b$, where $b \sim \mathbb{B}_{\mathcal{X}}$
    Compute the explanation-embedded instance $\widetilde{X}_i = \Psi_\theta(M_i, X_i)$
    Reparameterize the original distribution $\mathbb{P}_{\mathcal{X}}(X_i)$ and the explanation-embedded distribution $\mathbb{P}_{\widetilde{\mathcal{X}}}(\widetilde{X}_i)$
    Get the output predictions $f(X_i)$ and $f(\widetilde{X}_i)$, respectively
  **end for**
  Regularize $\boldsymbol{\pi}$ via $\mathcal{L}_M = \mathbb{E}\left[D_{\text{KL}}\left(\mathbb{P}_\phi\left(M \mid X\right) \| \mathbb{Q}\left(M\right)\right)\right] + \lambda_{\text{con}} \frac{1}{T \times D} \sum_{d=1}^{D} \sum_{t=1}^{T-1} \sqrt{\left(\pi_{t,d} - \pi_{t+1,d}\right)^2}$
  Providing uninformative areas through $\mathcal{L}_{dr}(\widetilde{X}, \widetilde{X}^r) = \frac{1}{T \times D} \sum_{d=1}^{D} \sum_{t=1}^{T} ||\widetilde{X}[t,d] - \widetilde{X}^r[t,d]||^2$
  Make embedded instances maintain the original distribution via $\mathcal{L}_{\text{KL}}(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\widetilde{\mathcal{X}}}) = \mathbb{E}\left[D_{\text{KL}}\left(\mathbb{P}_{\mathcal{X}}(X) \| \mathbb{P}_{\widetilde{\mathcal{X}}}(\widetilde{X})\right)\right]$
  Make label consistency between the logits of both predictors through $\mathcal{L}_{\text{LC}}(X, \widetilde{X}) = \mathbb{E}\left[D_{\text{JS}}(f(X) \| f(\widetilde{X}))\right]$
  Construct the total loss function as $\widetilde{\mathcal{L}} = \mathcal{L}_{\text{LC}} + \alpha \mathcal{L}_M + \beta(\mathcal{L}_{\text{KL}} + \mathcal{L}_{dr})$
  Update $\phi \leftarrow \phi - \eta \nabla_\phi \widetilde{\mathcal{L}}$ and $\theta \leftarrow \theta - \eta \nabla_\theta \widetilde{\mathcal{L}}$
**end for**
**Inference:**
**for** $i \leftarrow 1$ to $N$ **do**
  Get $\boldsymbol{\pi} = g_\phi(X_i)$ and sample final masks $M_i \sim \mathbb{P}(M_i \mid X_i)$
**end for**
**Return:** Mask $\{M_i\}_{i=1}^{N}$
```

F. Experimental Details

F.1. Description of Datasets

Following Queen et al. (2023), we conduct experiments and empirical analyses utilizing both synthetic and real-world datasets containing a total of eleven different data. Then we delineate the constitution of each dataset category, inclusive of the methodology for the derivation of ground-truth explanations where relevant.

Synthetic Datasets. Our research incorporates the use of four synthetic datasets, which are crafted with inherent ground-truth explanations. These datasets include **FreqShapes**, **SeqComb-UV**, **SeqComb-MV**, and **LowVar**, respectively, and are generated in the same way described in Queen et al. (2023). The creation of these datasets adheres to established methodologies that guard against the susceptibility to heuristic learning, as articulated by Geirhos et al. (2020) and further elaborated by (Faber et al., 2021) within the context of graph structures. Each time series dataset is established using a non-autoregressive moving average (NARMA) model to introduce noise to suit the generation of synthetic time series data. For each synthetic dataset, we have generated 5,000 training samples, 1,000 test samples, and 100 validation samples. The description of each dataset is systematically summarized in Table 5.

Real-world Datasets. We first employ four datasets derived from real-world time series classification tasks: **ECG** (Moody & Mark, 2001) for the detection of electrocardiogram arrhythmias; **PAM** (Reiss & Stricker, 2012) for the recognition of human activities; **Epilepsy** (Andrzejak et al., 2001) for the identification of electroencephalogram seizure episodes;

Table 5. The description of synthetic and real-world datasets.

DATASET	# OF SAMPLES	LENGTH	DIMENSION	CLASSES	TASK
FREQSHAPES	6,100	50	1	4	MULTI-CLASSIFICATION
SEQCOMB-UV	6,100	200	1	4	MULTI-CLASSIFICATION
SEQCOMB-MV	6,100	200	4	4	MULTI-CLASSIFICATION
LOWVAR	6,100	200	2	4	MULTI-CLASSIFICATION
ECG	92,511	360	1	5	ECG CLASSIFICATION
PAM	5,333	600	17	8	ACTION RECOGNITION
EPILEPSY	11,500	178	1	2	EEG CLASSIFICATION
BOILER	160,719	36	20	2	MECHANICAL FAULT DETECTION
WAFER	7,164	152	1	2	SENSOR CLASSIFICATION
FREEZERREGULAR	3,000	301	1	2	SENSOR CLASSIFICATION
WATER	573	168	13	2	BINARY CLASSIFICATION

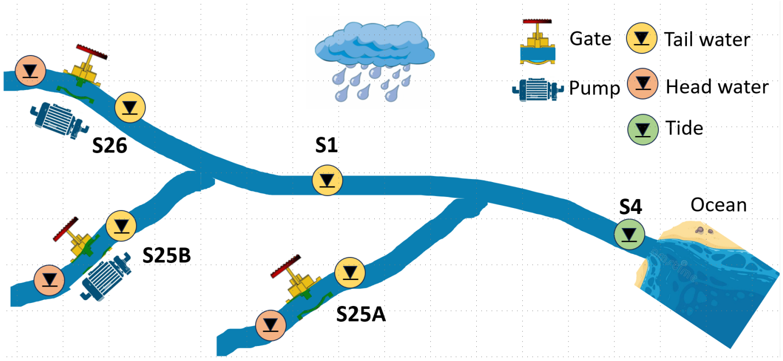


Figure 5. Diagram of the downstream of Miami River (Shi et al., 2023b).

and **Boiler** [2] for the automated identification of mechanical faults. These datasets are described in detail by Queen et al. (2023) and the process of building them is provided, which can be accessed through this url[3]. Note that only ECG has true explanatory labels, the other three datasets are only available for occlusion experiments. We also select two representative datasets **Wafer** and **FreezerRegular** in the UCR archive (Dau et al., 2019) to conduct occlusion experiments, which are commonly used time series classification to explore the effects of models.

Florida Water Data. We also conduct a real-world case study focusing on flood prediction in a South Florida coastal system. Following Shi et al. (2023b), we downloaded the pertinent data, including water level measurements from multiple stations, control schedules for various hydraulic structures (such as gates and pumps) along the river, tide information, and rainfall data. This dataset spans 11 years, ranging from 2010 to 2020, with hourly measurements. We start with constructing sliding windows, each spanning one week or 168 hours. These windows are tagged with labels based on the water levels observed at station S1. A label of 1 is assigned to a sliding window if, at any time within that window, water levels exceed the 95-th percentile, indicating a flood warning or the presence of flooding. Conversely, a label of 0 is assigned if no such conditions are met, signifying the absence of flooding. The primary feature description and the schematic diagram of the study domain are presented in Figure 5. We select 13 important features for water level alarm classification, as shown in Table 6. Thus each sample length is 168 and contains a 13-dimensional feature space. Further details can be referred to Shi et al. (2023b).

F.2. Black-box Hyperparameters and Performance

For black-box in all datasets, we employ a vanilla Transformer (Vaswani et al., 2017) as the classification black-box model. Faber et al. (2021) suggests that A good classification performance is necessary for explainability evaluation. Thus, we pick different hyperparameters for those black-box models as shown in Table 7, which are the same as in Queen et al. (2023) to ensure the best performance and fairness with baselines. The results of the classification performance can be found in

[2] https://dx.doi.org/10.21227/awav-bn36

[3] https://doi.org/10.7910/DVN/B0DEQJ

Table 6. Feature description of Florida water dataset.

FEATURES	NAME
WATER LEVEL	WS_S1, WS_S4, TWS_S25A, TWS_S25B, TWS_S26
STRUCTURE	GATE_S25A, GATE_S25B, GATE_S25B2, PUMP_S25B, GATE_S26_1,GATE_S26_2, PUMP_S26
PRECIPITATION	MEAN_RAIN

Table 8 in all datasets. All results guarantee the best performance as suggested by the original authors to ensure that the black-box models are strong predictors.

Table 7. Training parameters for transformer-based predictors across all ground-truth and real-world datasets.

PARAMETER	FREQSHAPE	SEQCOMB-UV	SEQCOMB-MV	LOWVAR	ECG	PAM	EPILEPSY	BOILER	WAFER	FREEZERREGULAR	WATER
LEARNING RATE	0.001	0.001	0.0005	0.001	0.002	0.001	0.0001	0.001	0.0001	0.0001	0.002
WEIGHT DECAY	0.1	0.01	0.001	0.01	0.001	0.01	0.001	0.001	0.001	0.001	0.001
EPOCHS	100	200	1000	120	500	100	300	500	200	300	500
NUM. LAYERS	1	2	2	1	1	1	1	1	1	1	1
d_h	16	64	128	32	64	72	16	32	16	16	64
DROPOUT	0.1	0.25	0.25	0.25	0.1	0.25	0.1	0.25	0.1	0.1	0.1
NORM. EMBEDDING	NO	NO	NO	YES	YES	NO	NO	YES	NO	NO	YES

Table 8. The performance of transformer-based predictors for time series classification. Throughout the experimental analyses conducted in this study, these models are consistently treated as time series black-boxes.

DATASET	F1	AUPRC	AUROC
FREQSHAPES	0.9920±0.0026	0.9995±0.0003	0.9998±0.0001
SEQCOMB-UV	0.9434±0.0104	0.9817±0.0061	0.9936±0.0021
SEQCOMB-MV	0.9745±0.0063	0.9951±0.0013	0.9983±0.0005
LOWVAR	0.9718±0.0033	0.9975±0.0006	0.9991±0.0002
ECG	0.9072±0.0228	0.9345±0.0247	0.9509±0.0232
PAM	0.8925±0.0073	0.9294±0.0042	0.9797±0.0015
EPILEPSY	0.9190±0.0040	0.9220±0.0074	0.9368±0.0064
BOILER	0.8396±0.0107	0.8129±0.0176	0.8982±0.0132
WAFER	0.9928±0.0012	0.9970±0.0009	0.9995±0.0001
FREEZERREGULAR	0.9793±0.0156	0.9888±0.0074	0.9902±0.0070
WATER	0.9121±0.0104	0.9490±0.0125	0.9469±0.0130

F.3. Details of Our Metrics

Following Crabbé & Van Der Schaar (2021), we use AUP and AUR as metrics to evaluate the efficacy of salient features, framing it as a binary classification task. For an explainer, we have an obtained mask $M \in [0,1]^{T\times D}$ as its explanation. Let $Q \in \{0,1\}^{T\times D}$ be a ground-truth matrix whose elements indicate the true saliency of the inputs contained in $X \in \mathbb{R}^{T\times D}$, where $Q[t,d]=1$ if the feature $X[t,d]$ is salient, otherwise it is 0. Let $\tau \in (0,1)$ be the detection threshold for $M[t,d]$ to indicate that the feature $X[t,d]$ is salient. This allows to convert the mask into an estimator $\hat{Q}[t,d](\tau)$ by:

$$\hat{Q}[t,d](\tau) = \begin{cases} 1 & \text{if } M[t,d] \geq \tau \\ 0 & \text{else.} \end{cases}$$

Consider truly salient index sets and index sets selected by the saliency method:

$$\begin{aligned} A &= \{(t,d) \in [1:T] \times [1:D] \mid Q[t,d]=1\}, \\ \hat{A}(\tau) &= \left\{(t,d) \in [1:T] \times [1:D] \mid \hat{Q}[t,d](\tau)=1\right\}. \end{aligned}$$

The precision and recall curves that map each threshold to a precision and recall score as:

$$\begin{aligned} \mathrm{P}: (0,1) \longrightarrow [0,1] : &\tau \longmapsto \frac{|A \cap \hat{A}(\tau)|}{|\hat{A}(\tau)|}, \\ \mathrm{R}: (0,1) \longrightarrow [0,1] : &\tau \longmapsto \frac{|A \cap \hat{A}(\tau)|}{|A|}. \end{aligned}$$

Table 9. Training parameters for TIMEX++ across all ground-truth and real-world experiments.

PARAMETER	FREQSHAPE	SEQCOMB-UV	SEQCOMB-MV	LOWVAR	ECG	PAM	EPILEPSY	BOILER	WAFER	FREEZERREGULAR	WATER
LEARNING RATE	0.001	0.001	0.002	0.005	0.0005	0.0005	0.0005	0.0001	0.0001	0.0005	0.001
BATCH SIZE	64	64	64	64	16	32	32	32	64	64	64
WEIGHT DECAY	0.001	0.001	0.001	0.0001	0.0001	0.001	0.001	0.001	0.001	0.001	0.01
SCHEDULER?	YES	YES	NO	NO	NO	NO	YES	YES	YES	YES	YES
EPOCHS	50	50	100	100	5	100	50	50	100	50	100
r	0.5	0.5	0.5	0.5	0.5	0.1	0.5	0.5	0.5	0.5	0.5
α	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
β	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
λ_{con}	1.0	2.0	2.0	2.0	2.0	0.0	2.0	2.0	2.0	2.0	2.0

Thus, the AUP and AUR scores can be derived by:

$$\begin{aligned}\text{AUP} &= \int_0^1 \text{P}(\tau)d\tau,\\ \text{AUR} &= \int_0^1 \text{R}(\tau)d\tau.\end{aligned}$$

G. Implementation Details

G.1. Details of Baseline Methods

We compare our TIMEX++ against six popular baselines. Integrated gradients (IG) (Sundararajan et al., 2017) is used as a general explainer; Dynamask (Crabbé & Van Der Schaar, 2021) and WinIT (Leung et al., 2023) are used as recent time-series specific explainers; CoRTX (Chuang et al., 2023) is used for applied contrastive learning; SGT + GRAD (Ismail et al., 2021) is demonstrated as an *in-hoc* explainer for time series; and TIMEX (Queen et al., 2023) is the strongest baseline explained by keeping the model behavior consistent. All hyperparameters follow the code provided by their authors, and the implementation of baselines is based on open source codes Dynamask[4] and TIMEX[5].

G.2. Details of Our Method

We opt for minimal tuning of hyperparameters, ensuring compatibility with parameters similar to those of TIMEX. When sparser explanations are required, the value of α needs to be increased to ensure the explanation extractor has compactness, β is used to control the generation of instances within the distribution, and λ regulates the continuity of explanations that is a smoothing constraint. For the above nine datasets, we list hyperparameters for each experiment performed in Table 9. For the explanation extractor g_ϕ, we build a transformer encoder-decoder structure which is the same as TIMEX, where an autoregressive transformer decoder with a 32-dimensional feed-forward, one attention head, and a sigmoid activation to out probabilities for each instance to generate $\boldsymbol{\pi}$. In the explanation conditioner $\Psi_\theta : \{\mathcal{X}, \mathcal{M}\} \mapsto \widetilde{\mathcal{X}}$, we use a single-layer MLP with an ELU activation to embedd the concatenation $[M, X]$, where $M \in \mathcal{M}, X \in \mathcal{X}$ and the number of hidden layers of MLP is 32. See our code for more details: https://github.com/zichuan-liu/TimeXplusplus.

H. Further Distribution Analysis

To visualize the OOD problem, we analyze the distribution between the explanation-embedded instances and the original instances produced by the different methods on three datasets: Freqshape (univariate), SeqComb-MV (multivariate), and ECG (real-world). We plot the shape of the dataset $\boldsymbol{X}$, where we make TSNE reduction of the time and feature level dimensions to 2 dimensions, as shown in Figure 6. It is clear that there are distributional biases in the reference instances (e.g. based on TIMEX), whereas the distributions in our generated explanation-embedded instances by TIMEX++ remain consistent with those between the original instances.

[4] https://github.com/JonathanCrabbe/Dynamask

[5] https://github.com/mims-harvard/TimeX

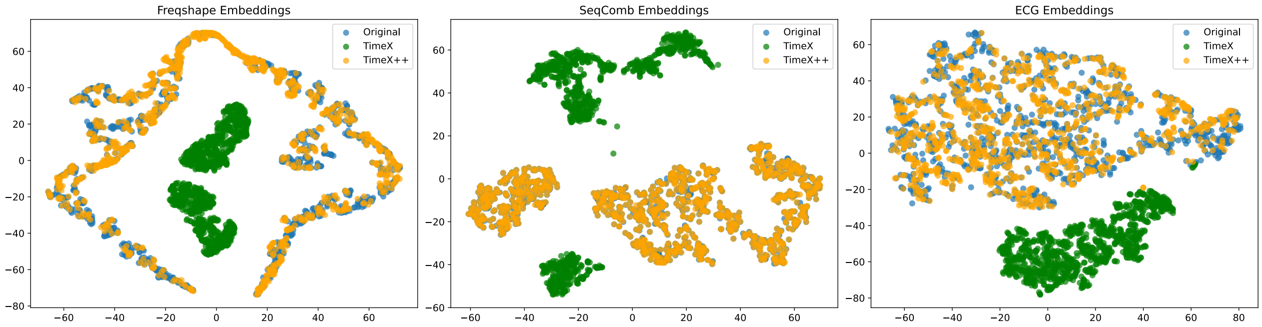


Figure 6. The distribution of explanations over Freqshape (univariate), SeqComb-MV (multivariate), and ECG (real-world) datasets.

I. Computing resources and Runtime

For computational resources, our experiments are performed on an NVIDIA 80GB Tesla A100 GPU. On average, the training runtime for each experiment in this study approximated 3-15 minutes per fold. The training time depends on the size of the data volume, e.g. the ECG dataset is large and requires a longer training time. Compared to TIMEX, which requires training both a white-box model for model consistency and the explanation masks, we directly perturb the black-box to train in less time. We also conduct an inference runtime experiment to compare the performance of TIMEX++ with that of baseline explainers. We select all real-world datasets of varying sizes for this comparison. Table 10 presents the inference time of the test data in five folds, which shows the time in seconds. In all datasets, TIMEX++ emerges as the most expedient model during the inference phase. Such an outcome aligns with expectations given that both Dynamask and IG necessitate recursive operations for individual samples during inference, in contrast to TIMEX++, which requires merely a single forward propagation. For the comparison with TIMEX, our method inference is slightly faster. This is since TIMEX forward passes through a trained white-box model $f^E(\cdot)$ that requires Landmark calculations. Whereas TIMEX++ is perturbing directly on the original black-box model. Since the masks of generators have the same structure and both use transformers, the difference in training/inference time between the two is not very significant. In summary, TIMEX++ enables efficient inference within a second response.

Table 10. Inference runtime of occlusion experiments for IG, Dynamask, TIMEX, and TIMEX++ on all real-world datasets.

METHOD	PAM	EPILEPSY	BOILER	WAFER	FREEZERREGULAR
IG	5.2314±0.2474	15.4665±0.3634	123.5558±0.8070	15.3950±0.3047	6.7452±0.5017
DYNAMASK	105.1664±0.6095	371.092±4.2567	3780.4326±468.1442	403.3997±31.6637	189.2849±1.7308
TIMEX	0.1638±0.2193	0.2042±0.2072	0.9176±0.2029	0.1634±0.0222	**0.2690**±0.4109
TIMEX++	**0.1611**±0.2179	**0.2009**±0.2077	**0.9096**±0.1767	**0.1262**±0.0021	0.2703±0.4169

J. Further Ablation Experiments

In this section, we conduct a comprehensive analysis of ablation studies on the TIMEX++ model across three datasets: including FreqShape (univariate), SeqComb-MV (multivariate), and ECG (real-world). This investigation serves to augment the preliminary ablation studies previously detailed for the ECG dataset in Section 5.2.

Effect of STE. We first conduct an experiment examining the effectiveness of using the STE for training TIMEX++. The results of our method with/without STE are shown in Table 11. The use of STE provided an average of 8.878% increase in explanation performance compared to no STE for all datasets in AUPRC. Moreover, the results indicate that the STE consistently enhances AUR for every dataset, while the AUP is only better for ECG without STE than with STE. In summation, the adoption of STE unequivocally demonstrates a substantial improvement in comparison to the continuous masking strategy, thereby providing tangible support for hard masks.

Effect of Different Losses. We now examine the effectiveness of different losses in the model components. We select the label consistency loss $\mathcal{L}_{\text{LC}}$, the maintenance loss $\mathcal{L}_{\text{KL}}$, and the loss of the reference distance $\mathcal{L}_{dr}$. The results of our

Table 11. Ablation of TIMEX++ with STE versus without STE, where if no STE denotes generating a continuous mask.

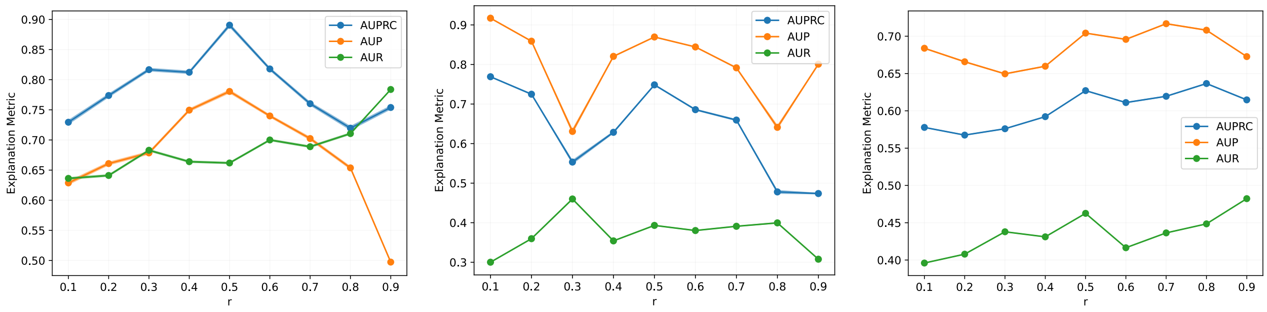
METHOD		WITHOUT STE			WITH STE	
	AUPRC	AUP	AUR	AUPRC	AUP	AUR
FREQSHAPES	0.7789±0.0033	0.7324±0.0033	0.6561±0.0020	**0.8905**±0.0018	**0.7805**±0.0014	**0.6618**±0.0019
SEQCOMB-MV	0.7269±0.0033	0.8727±0.0009	0.3478±0.0022	**0.7589**±0.0014	**0.8783**±0.0007	**0.3906**±0.0011
ECG	0.6152±0.0007	**0.7468**±0.0008	0.4023±0.0012	**0.6599**±0.0009	0.7260±0.0010	**0.4595**±0.0007

Table 12. Ablation of TIMEX++ considering whether there are different losses in our component.

DATASET	ABLATION	AUPRC	AUP	AUR
FREQSHAPES	FULL	**0.8905**±0.0018	**0.7805**±0.0014	0.6618±0.0019
	W/O $\mathcal{L}_{\mathrm{LC}}$	0.2251±0.0014	0.1959±0.0008	**0.7675**±0.0015
	W/O $\mathcal{L}_{\mathrm{KL}}$	0.8303±0.0027	0.6463±0.0027	0.7034±0.0022
	W/O $\mathcal{L}_{dr}$	0.1943±0.0017	0.1390±0.0015	0.4147±0.0019
SEQCOMB-MV	FULL	**0.7589**±0.0014	**0.8783**±0.0007	0.3906±0.0011
	W/O $\mathcal{L}_{\mathrm{LC}}$	0.0950±0.0033	0.0566±0.0018	0.3301±0.0080
	W/O $\mathcal{L}_{\mathrm{KL}}$	0.7484±0.0016	0.8694±0.0008	0.3531±0.0013
	W/O $\mathcal{L}_{dr}$	0.0688±0.0015	0.0532±0.0008	**0.6279**±0.0075
ECG	FULL	**0.6599**±0.0009	**0.7260**±0.0010	**0.4595**±0.0007
	W/O $\mathcal{L}_{\mathrm{LC}}$	0.6209±0.0019	0.6417±0.0020	0.4287±0.0015
	W/O $\mathcal{L}_{\mathrm{KL}}$	0.6417±0.0019	0.6979±0.0009	0.4424±0.0007
	W/O $\mathcal{L}_{dr}$	0.1516±0.0003	0.1405±0.0003	0.6313±0.0006

TIMEX++ with/without these losses are shown in Table 12, where containing all the losses of our method shows the best explanation performance. Specifically, the absence of $\mathcal{L}_{\mathrm{KL}}$ drops some of the performance, as the gap between the distribution of X and the distribution of $\widetilde{X}$ becomes larger, and it is not enough to rely on $\widetilde{X}^r$ alone to do the perturbation. Failure to predict an explanation when there is no $\mathcal{L}_{dr}$ is because there is not a reference explanation to generate, which is common sense. The simulation datasets (FreqShape and SeqComb-MV) likewise failed to explain when there was no consistency labeling $\mathcal{L}_{\mathrm{LC}}$, while the ECG was able to be somewhat normal. It is because TIMEX++ generates $\widetilde{X}$ without label leakage while within the sample distribution. This explains why the alone loss results observed individually perform poorly, while together these losses provide a powerful objective to obtain explanatory performance.

Choosing the Parameter r. One of the most significant parameters in training TIMEX++ is r, which governs the sparsity of the masks that are learned during the process. We conduct experiments on the above tree datasets to scrutinize the impact of varying r on the quality of explanations generated by the model, where we hold all hyperparameters constant while varying r. The outcomes of this variation are graphically represented in Figure 7, facilitating a visual interpretation of the relationship between the sparsity parameter r and the explanation quality. Lower values of the parameter are associated with a decrease in the performance of the explainer, as evidenced by the decline in AUR. It is worth noting that the performance stabilizes when the value is between 0.4 and 0.7, suggesting that the effectiveness of the interpreter is relatively insensitive to the exact choice of r in this range. Consequently, a value proximate to 0.5 is recommended for our experimental applications, a guideline that was adhered to in the course of this work.



(a) Freqshape (b) SeqComb-MV (c) ECG

Figure 7. Experiment on different datasets varying the r parameter.

K. Different Black-box Classifications

To explore the flexibility of TIMEX++, we study different time series classifiers and explore their explanatory role. We replace the original transformer-based black-box f to a long-short term memory (LSTM) or a convolutional neural network (CNN) as the underlying classifiers with the following hyperparameters: (i) For LSTM, we use 3 layer bidirectional LSTM and an MLP on the mean of last hidden states. (ii) For CNN, we use 3 layer CNN and an MLP on meanpool.

We compare TIMEX++ against strong baselines: IG, Dynamask, WinIT, and TIMEX. The results of the LSTM predictor are shown in Table 13-14. TIMEX++ retains the best AUPRC prediction on those datasets and is also slightly ahead for AUP and AUR overall, while TIMEX fails to predict on SeqComb-MV due to non-convergence. Tables 15-16 show the results of our method against strong baselines with a CNN predictor. Our method performs very well for both SeqComb-MV and ECG datasets, achieving the highest AUPRC and AUP for both datasets. However, the performance for FreqShapes AUPRC has high values for both ours and IG, making the comparison more difficult. Overall, our TIMEX++ maintains a relatively strong explanation performance among other black-box classifier architectures.

Table 13. Explainer results with LSTM predictor on FreqShapes and SeqComb-MV synthetic datasets.

	FREQSHAPES			SEQCOMB-MV		
METHOD	AUPRC	AUP	AUR	AUPRC	AUP	AUR
IG	0.9282±0.0016	0.7775±0.0010	0.6926±0.0017	0.2369±0.0020	0.5150±0.0048	0.3211±0.0032
DYNAMASK	0.2290±0.0012	0.3422±0.0037	0.5170±0.0013	0.2836±0.0021	0.6369±0.0047	0.1816±0.0015
WINIT	0.4171±0.0016	0.5106±0.0026	0.3909±0.0017	0.3515±0.0014	0.6547±0.0026	0.3423±0.0021
TIMEX	0.9903±0.0002	**0.7887**±0.0008	0.7963±0.0013	0.1298±0.0017	0.1307±0.0022	**0.4751**±0.0015
TIMEX++	**0.9939**±0.0002	0.7413±0.0009	**0.8428**±0.0008	**0.4052**±0.0038	**0.6804**±0.0052	0.3519±0.0021

Table 14. Explainer results with LSTM predictor on ECG dataset.

	ECG		
METHOD	AUPRC	AUP	AUR
IG	0.5037±0.0018	0.6129±0.0026	0.4026±0.0015
DYNAMASK	0.3730±0.0012	0.6299±0.0030	0.1102±0.0007
WINIT	0.3628±0.0013	0.3805±0.0022	0.4055±0.0009
TIMEX	0.6057±0.0018	0.6416±0.0024	0.4436±0.0017
TIMEX++	**0.6512**±0.0011	**0.7432**±0.0011	**0.4451**±0.0008

Table 15. Explainer results with CNN predictor on FreqShapes and SeqComb-MV synthetic datasets.

	FREQSHAPES			SEQCOMB-MV		
METHOD	AUPRC	AUP	AUR	AUPRC	AUP	AUR
IG	**0.9905**±0.0007	**0.8777**±0.0009	0.7056±0.0017	0.5979±0.0027	0.8858±0.0014	0.2294±0.0013
DYNAMASK	0.2574±0.0008	0.4432±0.0032	0.5257±0.0015	0.4550±0.0016	0.7308±0.0025	0.3135±0.0019
WINIT	0.5321±0.0018	0.6020±0.0025	0.3966±0.0017	0.5334±0.0011	0.8324±0.0020	0.2259±0.0020
TIMEX	0.7489±0.0046	0.4966±0.0033	0.7916±0.0021	0.7016±0.0019	0.7670±0.0012	**0.4689**±0.0016
TIMEX++	0.9134±0.0014	0.6066±0.0011	**0.7952**±0.0014	**0.7822**±0.0012	**0.8896**±0.0005	0.3434±0.0012

Table 16. Explainer results with CNN predictor on ECG dataset.

	ECG		
METHOD	AUPRC	AUP	AUR
IG	0.4949±0.0010	0.5374±0.0012	**0.5306**±0.0010
DYNAMASK	0.4598±0.0010	0.7216±0.0027	0.1314±0.0008
WINIT	0.3963±0.0011	0.3292±0.0020	0.3518±0.0012
TIMEX	0.6401±0.0010	0.7458±0.0011	0.4161±0.0008
TIMEX++	**0.6726**±0.0010	**0.7570**±0.0011	0.4319±0.0012

L. Visualization of FreqShape Dataset

Saliency maps are indeed a potent tool for visualizing the significance of features, particularly in multivariate time series analysis (Crabbé & Van Der Schaar, 2021; Liu et al., 2024b). Thus, we demonstrate the saliency maps of the benchmarks and TIMEX++ on the FreqShapes dataset, where increasing and decreasing subsequences determine the class label. Figure 8 illustrates these explainers and the corresponding ground-truth in a sample. IG identifies large areas as important and may prove to be untenable when applied to larger datasets replete with noise, where the pertinent signal may be less discernible. Dynamask seems to ignore several key sub-instances, often detecting only one or two salient segments with erroneous values. TIMEX is unable to describe important sub-instances with certainty, possibly due to constructed classifiers that produce data distribution bias. In stark contrast, our method focuses on the peaks for matching to ground-truth explanations.

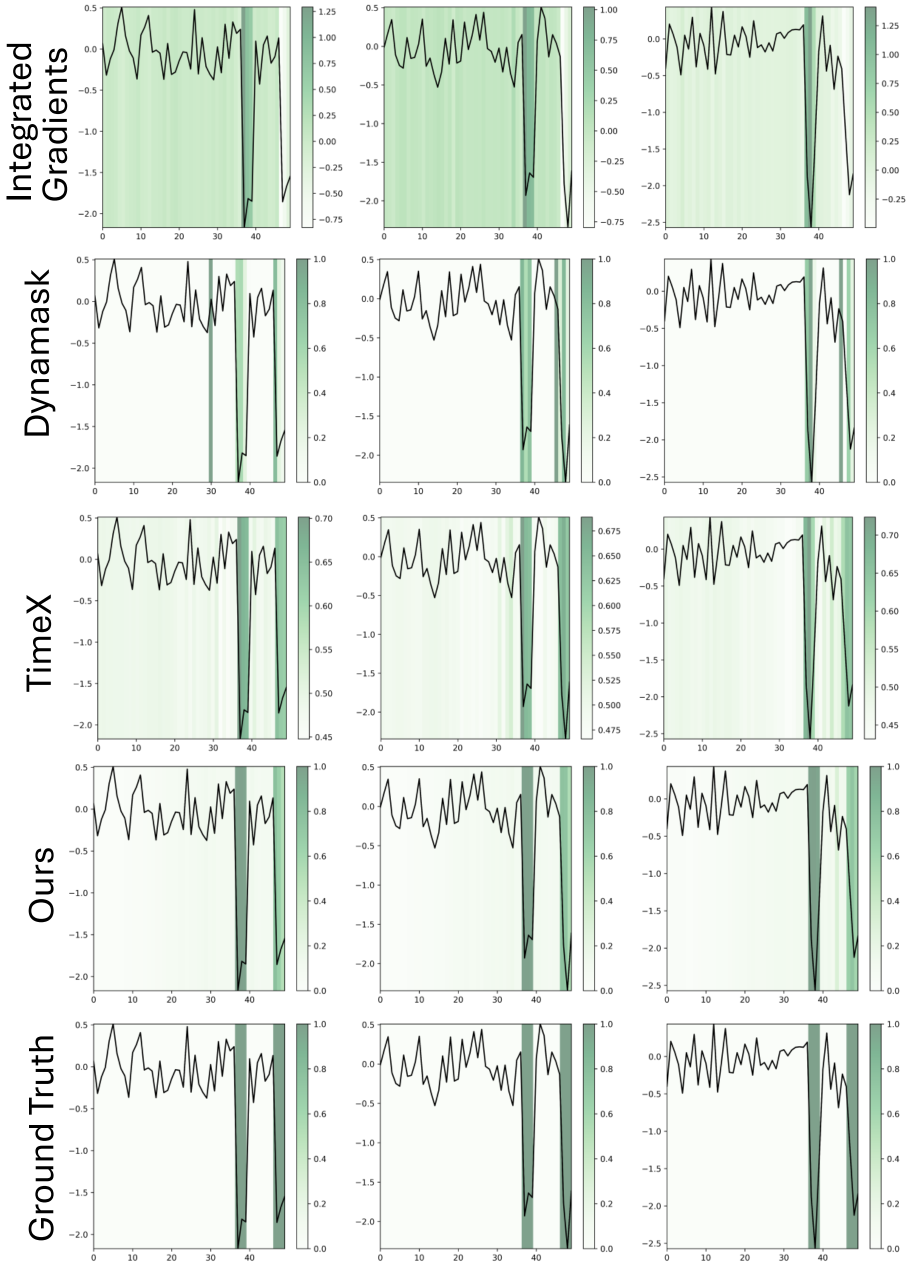


Figure 8. Visualization of all explainers on the FreqShapes dataset. Each column corresponds to a unique sample. For each row, the method used to generate the corresponding explanation figure is indicated, with the ground truth explanations presented in the bottom row.