Time series saliency maps: Explaining models across multiple domains

Christodoulos Kechris
EPFL
Lausanne, Switzerland
christodoulos.kechris@epfl.ch

Jonathan Dan
EPFL
Lausanne, Switzerland
jonathan.dan@epfl.ch

David Atienza
EPFL
Lausanne, Switzerland
david.atienza@epfl.ch

Abstract

Traditional saliency map methods, popularized in computer vision, highlight individual points (pixels) of the input that contribute the most to the model's output. However, in time-series they offer limited insights as semantically meaningful features are often found in other domains. We introduce Cross-domain Integrated Gradients, a generalization of Integrated Gradients. Our method enables feature attributions on any domain that can be formulated as an invertible, differentiable transformation of the time domain. Crucially, our derivation extends the original Integrated Gradients into the complex domain, enabling frequency-based attributions. We provide the necessary theoretical guarantees, namely, path independence and completeness. Our approach reveals interpretable, problem-specific attributions that time-domain methods cannot capture, on three real-world tasks: wearable sensor heart rate extraction, electroencephalography-based seizure detection, and zero-shot time-series forecasting. We release an open-source Tensorflow/PyTorch library to enable plug-and-play cross-domain explainability for time-series models. These results demonstrate the ability of cross-domain integrated gradients to provide semantically meaningful insights in time series models that are impossible with traditional time-domain saliency.

1 Introduction

Saliency maps are visual tools to explain deep learning models. Popularized in computer vision, they highlight input points that contribute the most to the model's output. For images, the original input domain, pixels, aligns naturally with human perception, since neighboring pixels form coherent objects that are understood by human vision. This makes pixel-level saliency intuitive and semantically meaningful. Similarly, in natural language processing, word-level attributions can be informative, as words inherently bear semantic meaning.

In contrast, in time-series this intuition breaks down. In the time domain, groups of temporally adjacent points - the equivalent of the pixel - do not necessarily form intuitive *concepts*. Rather, such *concepts* are found in intricate interactions between points, linking them to higher-level abstractions such as oscillating frequency patterns or statistically independent formations. As a consequence, highlighting individual time points does not provide meaningful insight into the behavior of the model.

Preprint. Under review.

Signal processing practice has long faced this challenge, where signal interpretation generally relies on the decomposition of the original signal into structured *components*. Through transformations, the original time domain is mapped to the component domain, capturing the higher-level interaction, and linking the input to semantically meaningful concepts. The choice of decomposition and component domain depends on the nature of the signals and the task. For example, the Fourier transform decomposes the original signal into sinusoid oscillations, while the Independent Component Analysis (ICA) decomposes the signal into statistically independent components. Such transformations map the time signals into structured, semantically rich domains, providing more intuitive interpretations of the signal's contents.

Building on this insight, we argue that visual explanations of time-series models should be expressed in interpretable domains, even when the model processes time points. We empirically demonstrate that the explainability power of available saliency-based methods is limited in the time domain. This motivates the need for saliency map tools that can visualize feature importance across multiple domains.

To address this, we develop Cross-domain Integrated Gradients, a novel method to visualize feature importance across multiple domains. We apply our method to real-world time-series models and applications, demonstrating that descriptive domains can be very powerful in understanding model behavior.

In this work, we introduce the following novel contributions:

- We propose a generalization of the Integrated Gradients that enables cross-domain explainability for any invertible transformation, including non-linear ones.
- We derive a generalization of the Integrated Gradients for real-valued functions with a complex domain, enabling the generation of frequency-domain saliency maps.
- We demonstrate how different domains allow for better understanding of model behavior on time-series data.
- We release an open-source Python library, compatible with `tensorflow` and `pytorch`, for cross-domain time series explainability: `https://github.com/esl-epfl/cross-domain-saliency-maps`. The code for reproducing the results of this paper is available here: `https://github.com/esl-epfl/cross-domain-saliency-maps-paper`.

2 Related works

Saliency map interpretation. Saliency maps as a means of interpreting the behavior of the model have been popularized in computer vision. These methods generate an output mapping each individual input pixel to a significance score. Several methods have been proposed for this mapping. Activation-based methods, such as GradCAM Selvaraju et al. [2017] and later variations Chattopadhay et al. [2018], generate saliency based on deep layer activations. Gradient-based methods such as Integrated Gradients (IG) Sundararajan et al. [2017], Kapishnikov et al. [2021] generate significance scores by using the model's output gradients with respect to its inputs. Similarly, Layer-wise Relevance Propagation (LRP) methods Bach et al. [2015] propose rules to propagate the model output backwards by splitting the overall output among individual input features.

Time domain explainability. Saliency map methods have been applied to time series applications, either by direct application of computer vision-derived methods Jahmunah et al. [2022], Tao et al. [2024] or by developing dedicated time series saliency approaches Queen et al. [2023], Liu et al. [2024]. In all cases, these approaches focus on identifying significant regions of the time domain input which contribute the most in the model's output. Such regions of interest are events that trigger the model's output.

Cross domain interpretability. The current time domain saliency methods have limitations, as highlighted parts of the signal may not be directly understood Theissler et al. [2022]. Additionally, Chung et al. [2024] demonstrate that such methods will highlight the same temporal region of interest, even when the underlying structures, e.g. frequency content, in those regions are different. These limitations diminish the explanatory power of the generated saliency map. To address this issue they proposed a perturbation method in the time-frequency domain, attributing model output to time-frequency components. However, frequency perturbations could be unnatural and model output could decrease due to out-of-distribution effects Sundararajan et al. [2017]. In a similar pattern, Vielhaben et al. Vielhaben et al. [2024] proposed the *virtual inspection layer* placed after the model input to transform the saliency map of the time domain to the frequency and time frequency domains, proposing dedicated relevance propagation rules for the frequency transform.

Despite progress in time-series saliency, existing methods (i) operate solely in the time domain, (ii) rely on perturbation-based attributions only in the frequency domain, or (iii) require transform-specific hand-crafted relevance-propagation rules. In contrast, our work provides a principled generalization of Integrated Gradients that supports any invertible, differentiable transform, including complex-valued domains, while preserving axiomatic properties and enabling semantically meaningful attributions across diverse time series applications.

3 Preliminaries

3.1 Problem statement and motivation

We consider a function $f : \mathcal{D}_s \to \mathbb{R}$ representing a deep learning model. The input $\boldsymbol{x} \in \mathcal{D}_s$ is constructed from a continuous-time signal $x(t) \in \mathbb{R}$ after discretizing it at a sampling frequency f_s [Hz] and considering a window of length L seconds: $\boldsymbol{x} = [x_0, ..., x_{n-1}]$, $n = f_s \cdot L$. Now consider a transform $T : \mathcal{D}_S \to \mathcal{D}_T$ that maps the original time domain to a semantically rich explanation target domain $\mathcal{D}_T$. Our task is to construct an informative saliency map that assigns a significance score to each characteristic $z_i = T(\boldsymbol{x})_i$ in the explanation domain.

Saliency maps developed in computer vision applications, and in partigular IG, provide explanations in the same domain as the model's inputs, i.e. $\mathcal{D}_T = \mathcal{D}_S$. Applying these methods to time-series models results in maps expressed in the time domain.

Proposition 1. *The time domain is not always informative in explaining f.*

We motivate Proposition 1 through a synthetic example. We provide additional real-world examples in Section 5 after formally defining our method.

3.2 Time domain explanation limitations

Consider that the input $\boldsymbol{x}$ is sampled from signals $x(t) = cos(2\pi\xi t + \phi)$. In this setup there are two classes of samples depending on the oscillating frequency ξ:

$$y = \begin{cases} 1, & \xi \sim \mathcal{N}(1.0, 0.5) \\ 2, & \xi \sim \mathcal{N}(4.0, 0.5) \end{cases} \tag{1}$$

We design a classifier f to distinguish between these two classes. We opt to manually construct f such that we have full mechanistic understanding of its inner workings. We choose a CNN architecture composed of a single convolutional layer with two channels followed by a ReLU activation and global average pooling $f(\boldsymbol{x}) = AvgPool\ (ReLU(\boldsymbol{w} * \boldsymbol{x}))$. The kernel of the first channel is a lowhigh-pass filter (cutoff at $2.5Hz$), while the second channel kernel is a highlow-pass filter with the same cutoff (see Figure 1).

Ideally, the model should be fully explained by describing its inner mechanism. In this particular scenario, we have designed f for this purpose, and hence a formal detailed explanation is available.

Mechanistic Interpretation 1. *Convolutional channel i allows only frequencies of class i to pass through the output; otherwise, the channel's output is almost zero, not activating. The ReLU and Average Pooling mechanism extract the amplitude of the signal Kechris et al. [2024a]. Hence, channel i of the model's output is only active when samples from class i are processed, leading to the correct classification of the input.*

That depth in model understanding is not easily available in bigger models which have been learned from samples. Hence, saliency maps are often used as a proxy. We provide IG explanations of the model f for samples from both classes expressed in the time and frequency domains (Figure 1). Although time points are periodically highlighted as *more important*, it is not exactly clear how this input tilts the model towards producing its output.

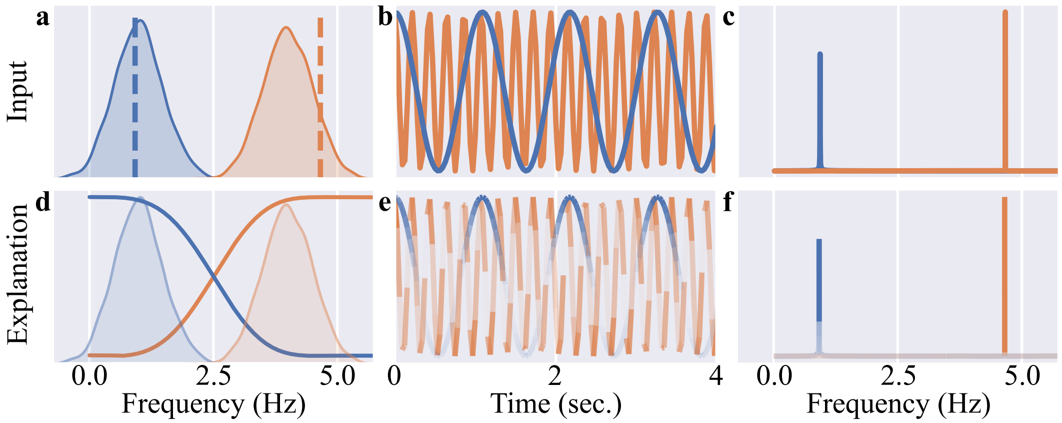

Figure 1: **Mechanistic interpretation along with Time and Frequency domain saliency maps. (a)** Distributions of the main frequency, ξ, for classes **one** and **two**. For producing the saliency maps, we sample one input for each class (vertical dashed lines). **(b)** The sampled inputs presented in the time and **(c)** frequency domains. **(d)** Illustration of the Mechanistic Interpretation1. We plot the frequency response for the **first** and **second** channels of the CNN. The sample distributions (a) are also overlayed. **(e)** Saliency maps expressed in the time and **(f)** frequency domains.

In contrast, a saliency map expressed in the frequency domain, which we introduce in Section 4, highlights the frequency structures that contributed to the final output: for the samples of class 1 only the 1Hz component contributes to the model's output, and accordingly for class 2 the 4Hz component. This saliency map is much more intuitive, provides useful information, and better aligns with the introduced mechanistic understanding (Mechanistic Interpretation 1) of this model. In Section 4 we show analytically that the frequency-sexpressed IG, as in this example, is directly linked to this detailed mechanistic explanation.

3.3 Integrated Gradients

To explain the output of a model f on an input $\boldsymbol{x}$ with a baseline $\hat{\boldsymbol{x}} \in \mathbb{R}^n$, IG generates a saliency map as Sundararajan et al. [2017]:

$$IG_i(\boldsymbol{x}) = (x_i - \hat{x}_i) \int_0^1 \frac{\partial f}{\partial x_i}\bigg|_{\boldsymbol{x}' + t\cdot(\boldsymbol{x}-\hat{\boldsymbol{x}})} dt \tag{2}$$

with each element $IG_i(x)$ of the map corresponding to the significance of the input feature x_i: saliencys is expressed in the same domain as the input. The IG definition relies on two key points from the theory of integrals over differential forms: the line integral definition and Stoke's theorem.

Line integral definition. The IG can be derived from the definition of the integral of the differential form df along the line $\boldsymbol{\gamma}(t) = \hat{\boldsymbol{x}} + t(\boldsymbol{x} - \hat{\boldsymbol{x}})$:

$$\int_\gamma df = \int \boldsymbol{\gamma}^* df = \int_0^1 \sum_{i=0}^{N} \frac{\partial f}{\partial x_i} \gamma_i'(t) dt = \sum_{i=0}^{N} \int_0^1 \frac{\partial f}{\partial x_i} \gamma_i'(t) dt = \sum_{i=0}^{N} (x_i - \hat{x}_i) \int_0^1 \frac{\partial f}{\partial x_i} dt \tag{3}$$

where $\boldsymbol{\gamma}^* df$ is the pullback of df by $\boldsymbol{\gamma}$: $\boldsymbol{\gamma}^* df = \sum_{i=0}^{N} \frac{\partial f}{\partial x_i} \gamma_i'(t) dt$ Do Carmo [1998]. Each individual element of the IG map $IG_i(\boldsymbol{x})$ corresponds to each element of the last sum of eq. 3.

Stoke's Theorem. The *Completeness* axiom of the IG Sundararajan et al. [2017]: $f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}}) = \sum IG_i$ is a consequence of the Stokes' Theorem for the case of integral of 1-form:$\int_\gamma df = \int_{\partial\gamma} f = f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}})$, which guarantees path independence: the value of the integral is only dependent on the first and last points of the path, not the path itself.

4 Methods

We now formally introduce the Cross-domain Integrated Gradients. Let $f : \mathcal{D}_s \rightarrow \mathbb{R}$ a deep neural network, operating on a domain $\mathcal{D}_s \subseteq \mathbb{R}^n$. Also, denote $\boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathcal{D}_s$ the input and baseline samples, respectively, as defined by the IG method. We introduce an invertible, differentiable transformation $T : \mathcal{D}_S \rightarrow \mathcal{D}_T$ and its inverse T^{-1}, also differentiable, with $\boldsymbol{z} = T(\boldsymbol{x})$ and $\boldsymbol{x} = T^{-1}(\boldsymbol{z})$ and $\mathcal{D}_T \subseteq \mathbb{C}^m$. The cross-domain IG generates the saliency map for f, attributing the difference $f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}})$ to the features $\boldsymbol{z}$, expressed in $\mathcal{D}_T$. To define Cross-domain Integrated Gradients, we consider the path integral of model gradients over the transformed feature space:

Definition 4.1 (Cross-domain Integrated Gradients)**.** *Given a model $f : \mathcal{D}_s \rightarrow \mathbb{R}$, a transform $T : \mathcal{D}_S \rightarrow \mathcal{D}_T$ and its inverse T^{-1}, input and baseline samples $\boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathcal{D}_s$ and $\boldsymbol{\gamma}(t)$ the line from $\boldsymbol{z} = T(\boldsymbol{x})$ to $\hat{\boldsymbol{z}} = T(\hat{\boldsymbol{x}})$ the Cross-Domain IG is defined as:*

$$IG_i^{\mathcal{D}_T}(\boldsymbol{z}) = 2\int_0^1 \mathrm{Re}\left\{ \frac{\partial (f \circ T^{-1})}{\partial z_i}\bigg|_{\boldsymbol{\gamma}(t)} \cdot (z_i - \hat{z}_i) \right\} dt \tag{4}$$

Note that the original IG, eq. 2, and $IG^{\mathcal{D}_T}$ explain the exact same functionality since $f(\boldsymbol{x})$ and $(f \circ T^{-1})(\boldsymbol{z})$ are equivalent. However, their output saliency maps are expressed in different domains. We now derive Definition 4.1 from first principles of the original Integrated Gradients method, Section 3.3.

Derivation sketch. The original IG is only defined for real inputs. To enable complex-valued transformations, such as the Fourier transform, we extend IG for real-valued functions g with complex inputs $\boldsymbol{z}$, referred to in this paper as *Complex IG*. Our derivation is based on the two key points introduced in Section 3.1:

1. **Line integral definition.** We begin our derivation by defining a function u that is equivalent to $g(\boldsymbol{z})$. Just like in the case of real inputs, eq. 2, we elaborate on the line integral $\int_\gamma du$. The end goal is to end up with a sum of integrals $\sum_i \int ...dt$ similar to eq. 3. In the final step, each IG element is defined as the corresponding integral term of the final sum, $\int ...dt$.
2. **Stokes' Theorem.** We carefully define u and derive the complex IG such that path independence holds, satisfying the *Completeness axiom*, which is not always guaranteed for functions of several complex variables Lebl [2019].

Lemma 4.1. *Let $g : \mathbb{C}^n \rightarrow \mathbb{R}$, $\boldsymbol{z} = \boldsymbol{p} + j\boldsymbol{q}$, with $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^N$, $\boldsymbol{\gamma}(t) = \hat{\boldsymbol{z}} + t(\boldsymbol{z} - \hat{\boldsymbol{z}}), t \in [0, 1]$ the line from the baseline point $\hat{\boldsymbol{z}}$ to the input point $\boldsymbol{z}$ and $\boldsymbol{n}(t) = \mathrm{Re}\{\boldsymbol{\gamma}(t)\}$ and $\boldsymbol{m}(t) = \mathrm{Im}\{\boldsymbol{\gamma}(t)\}$, $\boldsymbol{n}(t), \boldsymbol{m}(t) \in \mathbb{R}^n$. Then the IG of g in $\boldsymbol{z}$ is given by:*

$$IG_i^{\mathbb{C}^n}(\boldsymbol{z}) = \int_0^1 \left(\frac{\partial g}{\partial p_i} n_i'(t) + \frac{\partial g}{\partial q_i} m_i'(t) \right) dt \tag{5}$$

Proof. Let $u : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ such that $g(\boldsymbol{z}) = u(\boldsymbol{w}), \forall \boldsymbol{z} = \boldsymbol{p} + j\boldsymbol{q}, \boldsymbol{w} = [\boldsymbol{p}, \boldsymbol{q}]$. For the differential form of u:

$$du \coloneqq \sum_{i=0}^{2N} \frac{\partial u}{\partial w_i} dw_i \tag{6}$$

Similarly to the $g(\boldsymbol{z})$–$u(\boldsymbol{w})$ equivalence, we consider the equivalence between $\boldsymbol{\gamma}(t)$ and $\boldsymbol{a}(t) = [\boldsymbol{n}(t), \boldsymbol{m}(t)] \in \mathbb{R}^{2n}$. Then the pullback of du by $\boldsymbol{a}$ is :

$$\boldsymbol{a}^* du \coloneqq \sum_{i=0}^{2N} \frac{\partial u}{\partial w_i} a_i'(t) dt \tag{7}$$

Denoting with a_i' the i-th element of $d\boldsymbol{a}/dt$. The line integral of u along the line defined by $\boldsymbol{a}$ is:

$$\int_\gamma du = \int_\gamma \boldsymbol{a}^* du = \int_0^1 \sum_{i=0}^{2N} \frac{\partial u}{\partial w_i} a_i'(t) dt = \sum_{i=0}^{2N} \int_0^1 \frac{\partial u}{\partial w_i} a_i'(t) dt \tag{8}$$

Due to the equivalence between $\boldsymbol{w}$ and $\boldsymbol{p}, \boldsymbol{q}$ and u and g the latter sum can be formulated as :

$$\int_\gamma du = \sum_{i=0}^{N} \left(\int_0^1 \frac{\partial g}{\partial p_i} n_i'(t)dt + \int_0^1 \frac{\partial g}{\partial q_i} m_i'(t)dt \right) = \sum_{i=0}^{N} \int_0^1 \left(\frac{\partial g}{\partial p_i} n_i'(t) + \frac{\partial g}{\partial q_i} m_i'(t) \right) dt \quad (9)$$

which concludes the derivation. □

From Lemma 4.1 we conclude to Definition 4.1 by considering $g(\boldsymbol{z}) = f\left(T^{-1}(\boldsymbol{z})\right)$ and the complex differential form Range [1998]:

$$dg = \partial g + \overline{\partial} g \quad (10)$$

with $\partial g = \sum \partial g / \partial z_i dz_i$, $\overline{\partial} g = \sum \partial f / \partial \overline{z_i} \overline{dz_i}$ and the complex partial derivatives are defined as Range [1998] $\partial / \partial z_i = 1/2(\partial/\partial p - j\partial/\partial q)$ and $\partial/\partial \overline{z_i} = 1/2(\partial/\partial p + j\partial/\partial q)$. Then the pullback of dg by $\boldsymbol{\gamma}$ is :

$$\boldsymbol{\gamma}^* dg = \sum \frac{\partial g}{\partial z_i} \gamma_i'(t)dt + \sum \frac{\partial g}{\partial \overline{z_i}} \overline{\gamma_i'(t)} dt \quad (11)$$

Since $g \in \mathbb{R}$, $\partial g / \partial \overline{\boldsymbol{z}} = \overline{(\partial g / \partial \boldsymbol{z})}$, thus:

$$\boldsymbol{\gamma}^* dg = 2\,\mathrm{Re}\left\{ \sum \frac{\partial g}{\partial z_i} \gamma_i'(t)dt \right\} \quad (12)$$

Expanding the product into its real and imaginary parts produces the same form as eq. 9:

$$\boldsymbol{\gamma}^* dg = 2\,\mathrm{Re}\left\{ \sum \frac{1}{2} \left(\frac{\partial g}{\partial p_i} - j \frac{\partial g}{\partial q_i} \right) (n_i' + j m_i'(t))\, dt \right\} = \sum \left(\frac{\partial g}{\partial p_i} n_i'(t) + \frac{\partial g}{\partial q_i} m_i'(t) \right) \quad (13)$$

Thus the complex integrated gradient definition can be rewritten as:

$$IG_i^{\mathbb{C}^n} = 2 \int_0^1 \mathrm{Re}\left\{ \frac{\partial g}{\partial z_i} \gamma_i'(t) \right\} dt \quad (14)$$

Notice that:

$$\int_\gamma du = u(\boldsymbol{a}(1)) - u(\boldsymbol{a}(0)) = g(\boldsymbol{z}) - g(\hat{\boldsymbol{z}}) = f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}}) \quad (15)$$

maintaining the *Completeness* property.

If g processes real-valued inputs, then eq. 14 is equivalent to eq. 2: since $g(\boldsymbol{z}) = g(\boldsymbol{p} + j0)$, $\partial g / \partial q = 0$, $\partial g / \partial z = (1/2) \partial g / \partial p$. Thus if $\mathcal{D}_T \subseteq \mathbb{R}^n$ the cross-domain IG can equivalently be expressed as:

$$IG_i^{\mathcal{D}_T}(\boldsymbol{z}) = (z_i - \hat{z}_i) \int_0^1 \left. \frac{\partial (f \circ T^{-1})}{\partial z_i} \right|_{\boldsymbol{z}' + t \cdot (\boldsymbol{z} - \hat{\boldsymbol{z}})} dt \quad (16)$$

Complex IG on a single-layer single-channel CNN. Adebayo et al. [2018] analytically study a minimal single-layer convolutional network, demonstrating that IG can collapse into an *edge detector*, producing misleading saliency maps. Although this exposes a failure mode of the IG in the input domain, we show that Complex-IG faithfully reflects the inner mechanisms of a simple convolutional network in the frequency domain. In direct parallel, we derive a closed-form link between the complex IG saliency map of a CNN and the frequency response of its filters. Building on the example in Section 3.2, we work on a simple CNN and prove that Complex-IG highlights each filter's gain at its corresponding input frequency.

Let f be a convolutional neural network composed of a single convolutional layer (1 channel) followed by a ReLU operation and Global Average Pooling: $f(\boldsymbol{x}) = AvgPool\left(ReLU(\boldsymbol{w} * \boldsymbol{x})\right)$. We begin with the case in which f processes windows sampled from single-component sinusoidal signals $x(t) = a_j \cdot cos(2\pi \xi_j t + \phi),\ a_j > 0$. Then, the output $f(\boldsymbol{x})$ is Kechris et al. [2024a]:

$$f(\boldsymbol{x}) = \frac{a_j b_j}{\pi} \quad (17)$$

with b_i the amplification of the filter $\boldsymbol{w}$ at frequency $\xi_i Hz$: $b_i = \| \sum_n w_n e^{-2\pi \xi_i n} \|$. We employ the Complex IG method on f with baseline input $\hat{\boldsymbol{x}} = \mathbf{0}$, $f(\mathbf{0}) = 0$. This yields $IG_i^{\mathbb{C}^n} = 0, \ \forall i \neq j$ and $\sum_i IG_i^{\mathbb{C}^n} = f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}})$.Thus,

$$IG_j^{\mathbb{C}^n} = f(\boldsymbol{x}) = \frac{a_j b_j}{\pi} \tag{18}$$

This links $IG_j^{\mathbb{C}^n}$ to the output's frequency content $a_j b_j$ and by extension to the convolutional filter's frequency response. An example for the model introduced in Section 3.2 is presented in Figure 2.

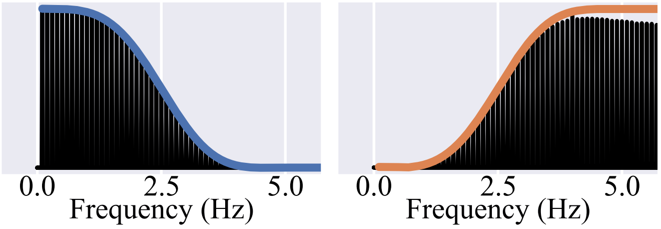

Figure 2: Frequency response (blue - orange) and frequency integrated gradients (black) for the two channels of the model of Section 3.2. We probe the model, performing frequency IG on samples with varying base frequencies.

Implementation. Autograd (pytorch / tensorflow) allows for automatic differentiation with complex variables using Wirtinger calculus Kreutz-Delgado [2009]. Thus, the complex IG can be directly approximated by autograd, using Definition 4.1 or Lemma 4.1, with the detail that Autograd (in both libraries) calculates the conjugate of the complex partial derivative. For the integral calculation, we use a summation approximation similar to Sundararajan et al. [2017]. The algorithms for estimating cross-domain IG for the case of $\mathcal{D}_T \subseteq \mathbb{R}^n$ and the two implementations on $\mathcal{D}_T \subseteq \mathbb{C}^n$ (Lemma 4.1 and Definition 4.1) are presented in Algorithms 1 and 2, 3 in the appendix, respectively.

5 Applications

We deploy cross-domain IG in a range of time series applications and models. For each application, we select an appropriate explanation space, based on domain knowledge.

5.1 Heart rate extraction from physiological signals

We use the KID-PPG Kechris et al. [2024b] model to extract heart rate (HR) from photoplethysmography (PPG) signals collected from a wrist-worn wearable device. We use signals from the PPGDalia dataset Reiss et al. [2019]. For a time window small enough for the HR frequency, ξ_{hr}, to be considered constant, a clean PPG signal can be modeled as Kechris et al. [2024b]:$x(t) = a_1 cos(2\pi \cdot \xi_{hr} \cdot t + \phi) + a_2 cos(2 \cdot \pi(2\xi_{hr}) \cdot t + \phi)$, with $a_1 > a_2$. However, due to sensor limitations, external interference signals are also usually present in PPG recordings Reiss et al. [2019], Kechris et al. [2024b].

Since our understanding in this application is mostly frequency-based, we have selected the frequency domain using the Fourier transform as the explanation target domain. This allows us to investigate whether the HR inference is produced from heart-related components or external interference. An illustration of two PPG inputs and the corresponding frequency-domain IGs are presented in Figure 3. The IG saliency maps allow us to identify samples in which the model infers heart rate from external interference, hence limiting the reliability of the model's output.

5.2 Electroencephalography-based epileptic seizure detection

We use the `zhu-transformer` Zhu and Wang [2023] which performs seizure detection on scalp-electroencephalography (EEG). We analyze a recording from the Physionet Siena Scalp EEG Database v1.0.0 Detti [2020], Detti et al. [2020], Goldberger et al. [2000]. We chose Independent Component Analysis Lee and Lee [1998] (ICA) as the transform to transform the time domain into a basis of statistically independent components. In EEG, different electric components, e.g., epileptic

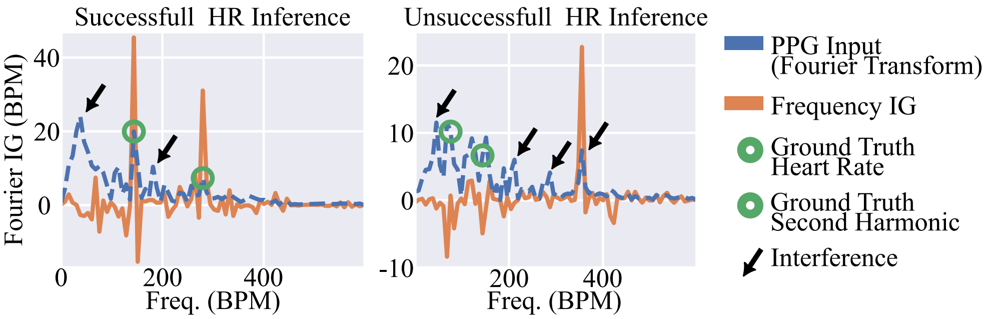

Figure 3: **Frequency-domain IG on heart rate inference model.** The **PPG** signal includes components from the **heart rate** and other components attributed to external interference (→), e.g. motion. **Left:** Sample with a small inference error 0.93 beats-per-minute (BPM). The **IG** highlights the two heart components located at hr and $2 \cdot hr$, with more weight given to the actual heart rate frequency. **Right:** **PPG** sample with high inference error (26.78 BPM). **IG** coefficients highlight frequency components which are not related to the heart.

activity or muscle interference, are spread across multiple channels. ICA isolates each individual activity/component to an activity-specific channel, assuming statistical independence between the components.

The ICA decomposition, along with the corresponding IG map, is presented in Figure 4.

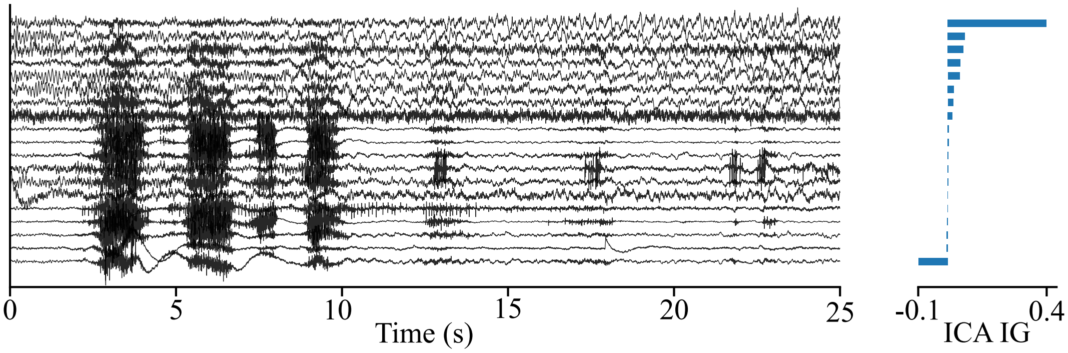

Figure 4: **ICA-domain IG on seizure detection model.** The ICA components are sorted from the component with the highest IG significance (top) to the lowest (bottom). **Left:** 19 output channels calculated from ICA on the original EEG channels. The first channel contains the majority of the epileptic activity, which is visible as an evolving pattern of spike-and-wave discharges at ~ 4.5 Hz. Some epileptic activity can also be found in the second channel. Significant muscle artifacts are isolated in the 9th-19th channels between 4 and 10 seconds. **Right:** IG saliency map calculated on the channel components. The map identifies the first channel as the most significant channel in detecting this sample as epileptic. Some significance, although much less, is also given to the next four channels. The channels corresponding to interference components do not get any significance in the output of the classifier. Finally, the last channel *tends to tilt* the classifier towards a non-epileptic output.

5.3 Foundation model time series forecasting

We use TimesFM Das et al. [2024] time-series foundation model to explain forecasting outputs. To isolate the relevant *concepts* we chose Seasonal-Trend decomposition using LOESS (STL) Cleveland et al. [1990] to decompose the input time series into trend and seasonality components.

We perform zero-shot forecasting, without any fine-tuning, on a time series with exponential trend and seasonal components(Figure 5). This attribution domain allows us to study the model's behavior for long-term forecasting horizons where the forecast error increases: the model underestimates the overall trend, while the seasonal component estimation presents a smaller error.

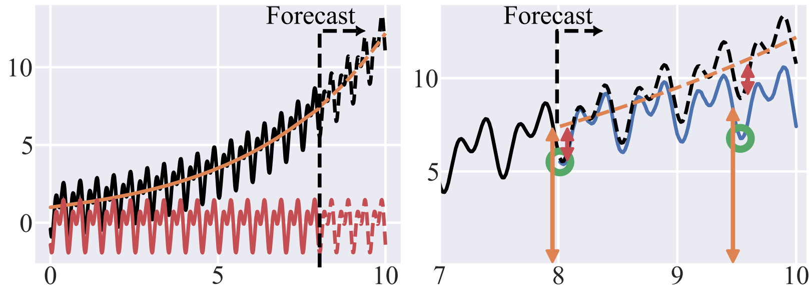

Figure 5: **Seasonal-Trend IG on time series foundation model. Left:** The **input** time series decomposed into **trend** and **seasonality** using STL. **Right:** Zero-shot **forecasting** using Timesfm along with **Seasonal-Trend** IG. For a small horizon, one step ahead prediction (**first circle**), TimesFM forecast error is small. Out of the overall output, 7.5 units are attributed to the trend components (⇕) which agrees with the ground truth trend (dashed orange line) and similarly the seasonal component (⇕) correctly contributes -1.96 units. For a larger forecast horizon (**second circle**) the forecast absolute error increases from 0.2 to 2.14. The majority of the error can be attributed to the model's underestimation of the **trend** (21% relative error), while the **seasonal** effect is correctly captured by the model (5.1% relative error).

6 Conclusions

We introduced a novel generalization of the Integrated Gradients method which enables saliency map generation in any invertible differentiable transform domain, including complex spaces. As transforms capture high-level interactions between input points, our methods enhances model explainability, especially in time-series data where individual time-point features are often uninformative. We demonstrated versatility of Cross-domain Integrated Gradients, applying it on a diverse set of time-series tasks, model architectures and explanation target domains. We release an open-source library to enable broader adoption of cross-domain time-series explainability.

Limitations. Our method requires an invertible, differentiable transform and a carefully selected baseline point. Consequently, we excluded non-invertible transforms and further investigation is needed for approximate-invertible cases. Baseline selection also plays a role in the final saliency map. We focused on the zero-signal as the baseline point - future work should include extensive investigation on the effects of the baseline selection. The current implementation also focuses on a linear integration path, reflecting the original IG. However, other non-linear paths, e.g. Guided IG Kapishnikov et al. [2021], should be explored.

Broader Impact. This work enables time-series model interpretability by generating saliency maps in meaningful domains, such as frequency or independent component bases. Fields where time signals are extensively used, such as healthcare, finance and environmental monitoring, could benefit from domain-specific saliency maps. In particular, with the recent rise of time-series foundation models, our method provides a strong investigation tool for inspecting model behavior.

However, risks may arise if the selected explanation target domain is not appropriate or saliency maps are over-interpreted. It is important to note that the saliency map solely provides feature significance scores. The interpretation of these scores requires domain expertise. We encourage a holistic interpretation approach of integrating domain knowledge with cross-domain saliency maps. We also caution that this method alone cannot function as definitive proof of the model's behavior. Responsible usage of the method should take into consideration model, data and transformation limitations, especially in high-stakes settings, such as in healthcare.

7 Acknowledgements

We thank Nikolaos Tsakanikas for insightful feedback on the methodological formulation and derivation. This research was partially supported by IMEC through a joint PhD grant for ESL-EPFL.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

Hyunseung Chung, Sumin Jo, Yeonsu Kwon, and Edward Choi. Time is not enough: Time-frequency based explanation for time-series black-box models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 394–403, 2024.

Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73, 1990.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Paolo Detti. Siena scalp eeg database v1.0.0. Physionet, 2020.

Paolo Detti, Giampaolo Vatti, and Garazi Zabalo Manrique de Lara. EEG synchronization analysis for seizure prediction: A study on data of noninvasive recordings. *Processes*, 8:846, 2020.

Manfredo P Do Carmo. *Differential forms and applications*. Springer Science & Business Media, 1998.

Ary L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 2000.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, 1 edition, May 2001. ISBN 9780471405405 9780471221319. doi: 10.1002/0471221317. URL `https://onlinelibrary.wiley.com/doi/book/10.1002/0471221317`.

Vicneswary Jahmunah, Eddie YK Ng, Ru-San Tan, Shu Lih Oh, and U Rajendra Acharya. Explainable detection of myocardial infarction using deep learning models with grad-cam technique on ecg signals. *Computers in Biology and Medicine*, 146:105550, 2022.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5050–5058, 2021.

Christodoulos Kechris, Jonathan Dan, Jose Miranda, and David Atienza. Dc is all you need: describing relu from a signal processing standpoint. *arXiv preprint arXiv:2407.16556*, 2024a.

Christodoulos Kechris, Jonathan Dan, Jose Miranda, and David Atienza. Kid-ppg: Knowledge informed deep learning for extracting heart rate from a smartwatch. *IEEE Transactions on Biomedical Engineering*, 2024b.

Marius Klug and Klaus Gramann. Identifying key factors for improving ICA-based decomposition of EEG data in mobile and stationary experiments. *The European Journal of Neuroscience*, 54(12): 8406–8420, December 2021. ISSN 1460-9568. doi: 10.1111/ejn.14992.

Ken Kreutz-Delgado. The complex gradient operator and the cr-calculus. *arXiv preprint arXiv:0906.4835*, 2009.

Jiri Lebl. *Tasty bits of several complex variables*. Lulu. com, 2019.

Te-Won Lee and Te-Won Lee. *Independent component analysis*. Springer, 1998.

Zichuan Liu, Tianchun Wang, Jimeng Shi, Xu Zheng, Zhuomin Chen, Lei Song, Wenqian Dong, Jayantha Obeysekera, Farhad Shirani, and Dongsheng Luo. Timex++: Learning time-series explanations with information bottleneck. *arXiv preprint arXiv:2405.09308*, 2024.

Owen Queen, Tom Hartvigsen, Teddy Koker, Huan He, Theodoros Tsiligkaridis, and Marinka Zitnik. Encoding time-series explanations through self-supervised model behavior consistency. *Advances in Neural Information Processing Systems*, 36:32129–32159, 2023.

R Michael Range. *Holomorphic functions and integral representations in several complex variables*, volume 108. Springer Science & Business Media, 1998.

Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14):3079, 2019.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

Rui Tao, Lin Wang, Yingnan Xiong, and Yu-Rong Zeng. Im-ecg: An interpretable framework for arrhythmia detection using multi-lead ecg. *Expert Systems with Applications*, 237:121497, 2024.

Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. Explainable ai for time series classification: a review, taxonomy and research directions. *Ieee Access*, 10:100700–100724, 2022.

Johanna Vielhaben, Sebastian Lapuschkin, Grégoire Montavon, and Wojciech Samek. Explainable ai for time series via virtual inspection layers. *Pattern Recognition*, 150:110309, 2024.

Irene Winkler, Stefan Haufe, and Michael Tangermann. Automatic Classification of Artifactual ICA-Components for Artifact Removal in EEG Signals. *Behavioral and Brain Functions*, 7(1): 30, August 2011. ISSN 1744-9081. doi: 10.1186/1744-9081-7-30. URL `https://doi.org/10.1186/1744-9081-7-30`.

Yuanda Zhu and May D Wang. Automated seizure detection using transformer models on multichannel eegs. In *2023 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–6. IEEE, 2023.

A Cross-domain IG Algorithms

Algorithm 1 Real Target Domain IG

```
Input: $f(\cdot), x, \hat{x}, n_{iter}$
Output: $IG$
$i \leftarrow 1$
$sum \leftarrow 0$
$tape \leftarrow tensorflow.GradientTape()$
$X' \leftarrow T(\hat{x})$
for $i \leq n_{iter}$ do
    $z \leftarrow T(x)$
    $z \leftarrow \hat{z} + (z - \hat{z}) \cdot i/n_{iter}$
    tape.watch($z$)
    $x_{rec} \leftarrow T^{-1}(z)$
    $y \leftarrow f(x_{rec})$
    $dy \leftarrow tape.gradient(y, z)$
    $sum \leftarrow sum + dy$
    $i \leftarrow i + 1$
end for
$sum \leftarrow sum/n_{iter}$
$IG = (z - \hat{z}) \cdot sum$
```

Algorithm 2 Complex Target Domain IG

```
Input: $f(\cdot), x, \hat{x}, n_{iter}$
Output: $IG$
$i \leftarrow 1$
$sum\_real \leftarrow 0$
$sum\_imag \leftarrow 0$
$tape\_real \leftarrow tensorflow.GradientTape()$
$tape\_imag \leftarrow tensorflow.GradientTape()$
$\hat{z} \leftarrow T(\hat{x})$
for $i \leq n_{iter}$ do
    $X \leftarrow T(x)$
    $z \leftarrow \hat{z} + (z - \hat{z}) \cdot i/n_{iter}$
    $re\_z \leftarrow \mathrm{Re}\{z\}$
    $im\_z \leftarrow \mathrm{Im}\{z\}$
    $tape\_real.watch$(re_z)
    $tape\_imag.watch$(im_z)
    $\hat{z} \leftarrow re\_z + j \cdot im\_z$
    $x_{rec} \leftarrow T^{-1}(\hat{z})$
    $y \leftarrow f(x_{rec})$
    $re\_dy \leftarrow tape\_real.gradient(y, re\_z)$        ▷ Calculate $\frac{\partial g}{\partial p_i}$
    $im\_dy \leftarrow tape\_imag.gradient(y, im\_z)$        ▷ Calculate $\frac{\partial g}{\partial q_i}$
    $sum\_real \leftarrow sum\_real + re\_dy$
    $sum\_imag \leftarrow sum\_imag + im\_dy$
    $i \leftarrow i + 1$
end for
$sum\_real \leftarrow sum\_real/n_{iter}$
$sum\_imag \leftarrow sum\_imag/n_{iter}$
$IG = \mathrm{Re}\{z - \hat{z}\} \cdot sum\_real + \mathrm{Im}\{z - \hat{z}\} \cdot sum\_imag$
```

B EEG and ICA

The raw EEG input is presented in Figure 6.

Algorithm 3 Complex Target Domain IG with complex differential

```
Input: $f(\cdot), x, \hat{x}, n_{iter}$
Output: $IG$
$i \leftarrow 1$
$sum \leftarrow 0$
$tape \leftarrow tensorflow.GradientTape()$
$\hat{z} \leftarrow T(\hat{x})$
for $i \leq n_{iter}$ do
    $z \leftarrow T(z)$
    $z \leftarrow \hat{z} + (z - \hat{z}) \cdot i/n_{iter}$
    tape.watch($z$)
    $x_{rec} \leftarrow T^{-1}(z)$
    $y \leftarrow f(x_{rec})$
    $dy \leftarrow tape.gradient(y, X)$
    $sum \leftarrow sum + \overline{dy}$
    $i \leftarrow i + 1$
end for
$sum \leftarrow sum/n_{iter}$
$IG = 2\,\mathrm{Re}\{(z - \hat{z}) \cdot sum\}$
```

The implementation of the `zhu-transformer` we used can be found here `https://github.com/esl-epfl/zhu_2023`.

The application of ICA in EEG signals is based on the general assumption that the EEG data matrix $X \in \mathbb{R}^{N \times M}$ is a linear mixture of different sources (activities) $S \in \mathbb{R}^{N \times M}$ with a mixing matrix $A \in \mathbb{R}^{N \times N}$ such that $X = AS$, where N is both the number of sources and EEG channels, and M is the number of samples in the dataset. Sources are assumed to be statistically independent and stationary. These assumptions can be leveraged to compute an inverse unmixing matrix $W = A^{-1} (\in \mathbb{R}^{N \times N})$, such that $S = WX$. Finding W is an ill-posed problem without an analytical solution which can be estimated by means of different ICA algorithms Hyvärinen et al. [2001], Klug and Gramann [2021]. ICA is used in EEG to decompose the signal into independent components that separate the signal of interest from various sources of artifacts Winkler et al. [2011]. In this work, for ICA we selected the FastICA algorithm implemented in `sklearn` (`max_iter` $= 3 \cdot 10^4$, `tol` $= 1 \cdot 10^{-8}$).

The independent channels estimated using ICA are presented in Figure 7.

C Generated time series for TimesFM forecasting

We generate a synthetic time series signal, $x(t)$, composed of an exponential trend, $x_{trend}(t)$, and a seasonal component, $x_{seasonal}(t)$:

$$\begin{aligned} x_{trend}(t) &= e^{\frac{t}{4}} \\ x_{seasonal}(t) &= sin(2\pi \cdot 2 \cdot t + \phi) + sin(2\pi \cdot 4 \cdot t + \phi) \\ x(t) &= x_{trend}(t) + x_{seasonal}(t) \end{aligned}$$

A window of 512 time points, starting at $t = 0$, are given as input to TimesFM which generates forecasts up to 128 time points in the future from $t = 512$. The input time series and STL decomposition are presented in more detail in Figure 8.

D Experiments compute resources

All experiments were run on an NVIDIA Tesla V100 with 32GB memory.

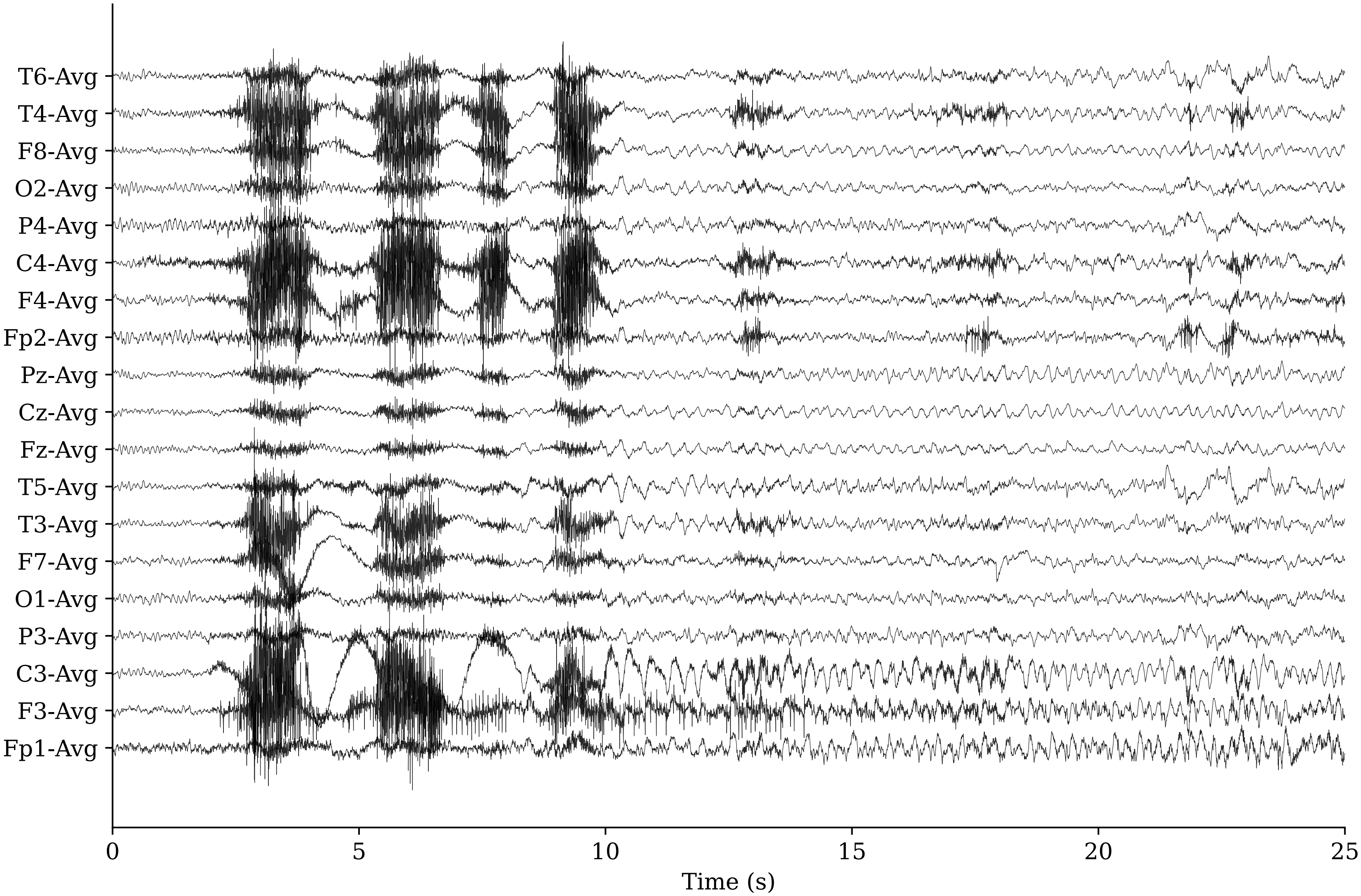

Figure 6: EEG signal in the original channel space.

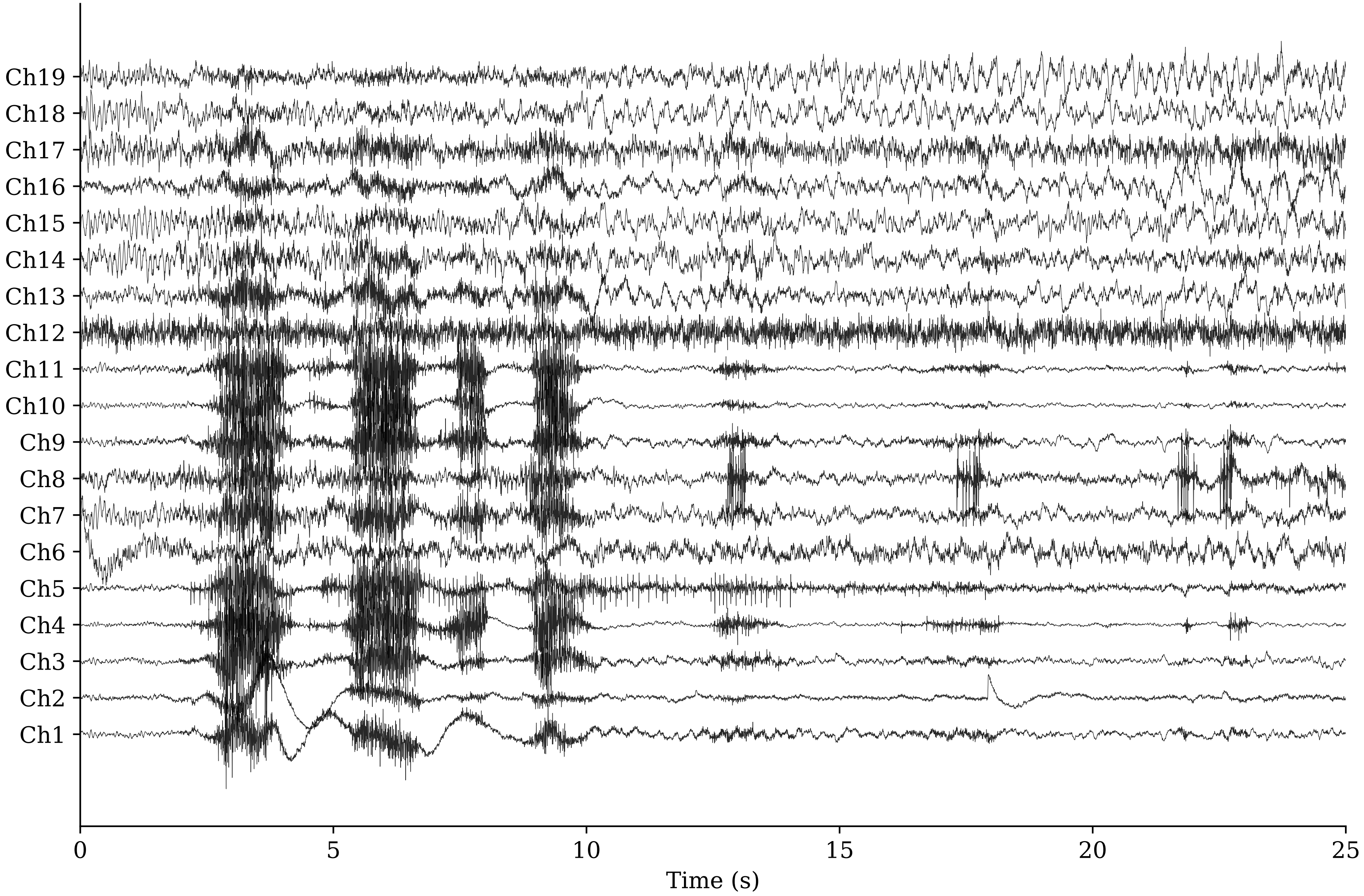

Figure 7: EEG signal in the Independent Component space.

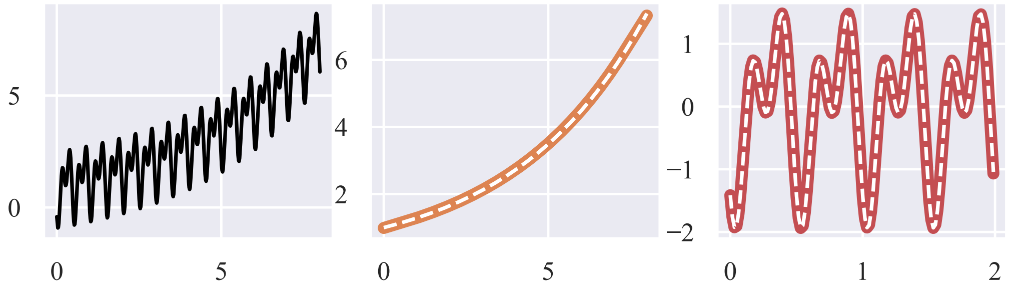

Figure 8: **Input time series for forecasting and successful STL decomposition. Left:** time series with a **trend** and a **seasonal** component. **Center:** The decomposed **trend** component and ground truth trend (white dashed line). **Right:** The decomposed **seasonal** component and ground truth seasonality (white dashed line).