Pattern Recognition Letters 199 (2026) 239–245

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Additive decomposition of one-dimensional signals using Transformers

Samuele Salti [a,*], Andrea Pinto [c,1], Alessandro Lanza [b], Serena Morigi [b]

[a] Department of Computer Science and Engineering, Viale Risorgimento, 2, Bologna 40136, BO, Italy
[b] Department of Mathematics, Piazza di Porta San Donato 5, Bologna, 40126, BO, Italy
[c] Master student, University of Bologna, Italy

ARTICLE INFO

Editor: Shaun Canavan

MSC:
41A05
41A10
65D05
65D17

Keywords:
Additive signal decomposition
Deep learning
Transformer
Non-stationary signal

ABSTRACT

One-dimensional signal decomposition is a well-established and widely used technique across various scientific fields. It serves as a highly valuable pre-processing step for data analysis. While traditional decomposition techniques often rely on mathematical models, recent research suggests that applying the latest deep learning models to this very ill-posed inverse problem represents an exciting, unexplored area with promising potential. This work presents a novel method for the additive decomposition of one-dimensional signals. We leverage the Transformer architecture to decompose signals into their constituent components: piecewise constant, smooth (trend), highly-oscillatory, and noise components. Our model, trained on synthetic data, achieves excellent accuracy in modeling and decomposing input signals from the same distribution, as demonstrated by the experimental results.

1. Introduction

Signal decomposition (SD) approaches aim to decompose non-stationary signals into their constituents. This serves as a key preliminary step in many signal processing workflows providing useful knowledge and insight into the data and the systems that generated it. SD enables deeper insights and more effective decision-making across various domains like environmental science, Chen et al. [1], health monitoring and biomedical signal processing, Lin et al. [2], finance, time-series and speech analysis, Huang et al. [3], Singh et al. [4] and Zhou et al. [5]. SD techniques are used in many other fields where understanding the constituent parts of a complex signal is crucial for further analysis by facilitating tasks such as removing noise or artifacts and extracting key features; see, e.g. [6] and [7].

Quite a lot of work focuses on the decomposition of a multi-component signal into amplitude-modulated (AM) and frequency-modulated (FM) oscillatory components; see [8] for a comprehensive review of the popular SD approaches into its constituent AM–FM components. However, two are the significant challenges for all these methods: noise, which is ubiquitous in most real-life signals, masking the desirable information content in the data; and the presence of components with flat sections linked by abrupt jumps, commonly known as piecewise constant signals.

Hence, to take these challenges into account, the SD problem considered in this paper is based on the hypothesis that an observed discrete signal $f \in \mathbb{R}^M$ can be mathematically represented as an additive mixture of four components,

$$f = c + s + o + n\,, \tag{1}$$

with $c \in \mathbb{R}^M$ representing a piecewise constant (also known as *cartoon*) component, $s \in \mathbb{R}^M$ a smooth (*trend* or *seasonal*) component, $o \in \mathbb{R}^M$ a highly-oscillatory component, and $n \in \mathbb{R}^M$ a noise component which we assume to be Additive White Gaussian Noise (AWGN), i.e., it contains the realization of a Gaussian-distributed M-variate random vector with zero-mean and covariance matrix equal to a scaled identity.

Although this specific SD problem has been recently tackled with variational approaches [9–11], they suffer from some shortcomings, such as low computational efficiency and high sensitivity to hyper-parameters, which are usually tuned to each test signal. Data-driven methods, and in particular deep learning models, hold the potential to overcome these limitations, since they are usually very fast and can seamlessly generalize to all signals sampled from the data-generating distribution seen at training time, without any test-time parameters tuning.

* Corresponding author.
E-mail addresses: samuele.salti@unibo.it (S. Salti), pintoandrea097@gmail.com (A. Pinto), alessandro.lanza2@unibo.it (A. Lanza), serena.morigi@unibo.it (S. Morigi).

[1] Andrea Pinto was a master student of the University of Bologna at the time the study was carried out. Currently, he is not affiliated with a research institution.

https://doi.org/10.1016/j.patrec.2025.11.002
Received 9 June 2025; Received in revised form 22 October 2025; Accepted 2 November 2025
Available online 17 November 2025

Our research showcases one of the earliest applications of learning techniques to the SD problem (1) in the case of more general and realistic mixtures of signals which include noise, piecewise constant, and oscillatory parts. In particular, the neural architecture chosen to solve the problem is the Transformer [12], first introduced in natural language processing due to its ability to capture long-range dependencies. We speculate that such dependencies can also be important in the decomposition of a signal, since the value of the components at each time step can be estimated more effectively by having a global view on the input signal. We refer to the proposed Trasformer-based approach for solving the SD problem with the TSD acronym.

The contributions of this paper are as follows:
- a neural network architecture for the SD problem based on Transformers;
- a large synthetic dataset of signals with their ground-truth components, that we make publicly available to foster research on this important topic[2];
- an experimental comparison against a state-of-the-art variational method, that highlights pros and cons of the data-driven approach to SD problems.

2. Related work

In the field of SD, key techniques are Fourier Decomposition Method (FDM) [4], and Empirical Mode Decomposition (EMD) introduced in [3]. Both break down a signal into a small number of its basic oscillatory components, Intrinsic Mode Functions and Fourier intrinsic band functions, respectively. Later, other methods gained popularity, like synchrosqueezing [13], which is similar to EMD but uses the Wavelet Transform for improved time–frequency analysis and reconstruction of individual oscillatory modes. The most recent advances involve Variational Mode Decomposition (VMD) [14], which decomposes a signal into distinct band-limited components. VMD is entirely non-recursive and more resistant to noise. Variational methods, which have been classically used for the decomposition of images and, more generally, of functions defined on surfaces – see, e.g., [15,16], respectively – have more recently also been proposed for the decomposition of 1D signals. For instance, the two-steps JOT method in [9] separates signals into Jump, Oscillatory and Trend components, rather than just oscillatory and non-oscillatory parts. The work in [10] defines masked proximal operators of each of the component losses and obtains the decomposition by minimizing the weighted sum of all component losses. In general, variational methods can obtain good decomposition results but require an expensive per-signal tuning since they are highly sensitive to change in the free hyper-parameters. An automatic selection of the free parameters based on whiteness and autocorrelation is introduced in the Predictor-Corrector SD approach proposed in [11]. Another limitation of variational methods is their low computational efficiency.

From the perspective of a deep learning approach to SD, to our knowledge, there is only a recent proposal in [17] to compute non-stationary decomposition of signals. The authors present a convolutional neural network with a residual structure. However, their additive decompositions focus only on the simpler problem of separating noise and simple oscillatory components.

Transformer networks have become incredibly popular in signal analysis, and their attention mechanism is indeed the core reason for this success [18]. Despite their capabilities, Transformers remain largely unexplored for SD. This is a missed opportunity, as decomposing time series into trend and seasonal elements can greatly enhance the accuracy of tasks such as time series forecasting. Autoformer in place of the Transformer is proposed in [19] for long-term time series forecasting; it introduces an auto-correlation mechanism in place of self-attention (SA). In Frequency Enhanced Decomposed Transformer (FEDformer) [20], a frequency enhanced method is introduced to compute attention. However, in all these methods, while the machine learning model is used to predict future samples in the series based on the decomposed components, the decomposition itself is not tackled with a data-driven approach. Contrary to what we investigate in this paper, decomposition in these methods still relies on model-based approaches and hand-crafted rules.

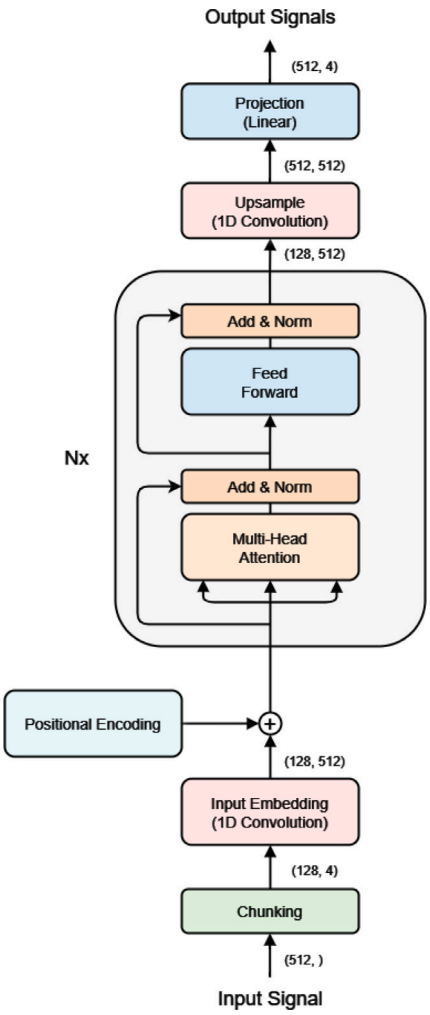

Fig. 1. Neural architecture of the proposed TSD approach.

3. Proposed TSD method

The neural architecture that we investigate in the context of additive SD is the Transformer [12], first introduced in natural language processing because of its ability to capture long-range dependencies. While initially developed for natural language processing, the underlying principles of the transformer architecture, particularly attention, are highly versatile. This has allowed them to be successfully adapted to various data analysis applications beyond text, including image processing, Dosovitskiy et al. [21], audio and speech processing, Zaman et al. [22], etc..

The ability to compare different parts of the input signal seems conducive to SD, where the presence and profile of a component is suggested also to a human observer by comparisons among sections of the input. While the original proposal features both an encoder and a decoder, it has been quite common to use only one of the two in subsequent works, and we have opted to use only the encoder component of the original Transformer architecture. Its ability to transform an input

[2] To be published upon acceptance.

sequence into a processed sequence of the same length seems a good fit to model the SD problem. We only need to insert simple layers before and after the Transformer encoder to adapt it to the SD task, which are described below. We note here that also the decoder part of the architecture could be used here, to cast the SD problem as a generative task. We leave the exploration of such research path to future work.

The model takes as input a sequence of scalars $f_i,\ i = 1, \dots, M$ and produces four sequences of the same length c_i, s_i, o_i, n_i, corresponding to the piecewise constant, smooth, oscillatory and noise components of the input sequence. The Transformer encoder takes as input a sequence of vectors $y_j \in \mathbb{R}^D, j = 1, \dots, L$ called tokens and produces a sequence of processed tokens of the same length and dimensionality $z_j \in \mathbb{R}^D, j = 1, \dots, L$. The encoder is a stack of N identical layers. Each layer consists of two parts: multi-head self-attention (SA) and a fully connected feed-forward network. Each part is surrounded by a skip connection and followed by layer normalization [23]. If we stack the input tokens as rows of a matrix $Y \in \mathbb{R}^{L \times D}$, the SA operator of Y is

$$\mathrm{SA}(Y) = \mathrm{softmax}\left(\frac{YQK^TY^T}{\sqrt{D}}\right)(YV), \tag{2}$$

where $Q, K \in \mathbb{R}^{D \times D_k}, V \in \mathbb{R}^{D \times D_v}$ are learnable projection matrices that compute the so called queries, keys, and values out of the input matrix Y, the softmax operation is applied row-wise, and D_k, D_v are free hyperparameter, which set the length of queries/keys and values, respectively. Multi-head SA uses h SA modules, without sharing parameters between them and setting $D_k = D_v = D/h$, then concatenates their output along rows, and applies a final learnable linear projection $O \in \mathbb{R}^{D \times D}$

$$\mathrm{MHSA}(Y) = \left[SA_1(Y)|SA_2(Y)|\dots|SA_h(Y)\right] O. \tag{3}$$

The feed-forward network is a 2-layer fully connected network with ReLU activation function,

$$\mathrm{FF}(Y) = \mathrm{ReLU}(YW_1^T + b_1)W_2^T + b_2, \tag{4}$$

where $W_1 \in \mathbb{R}^{4D \times D}, W_2 \in \mathbb{R}^{D \times 4D}, b_1 \in \mathbb{R}^{4D}, b_2 \in \mathbb{R}^D$ are learnable parameters, and where the symbols + in (4) are realized by replicating the arrays b_1 and b_2 so that they are added to each row of the matrix they are summed to.

In our experiments, L is always less than or equal to M, since both the time and space complexity of the transformer blocks is quadratic in L. Therefore, to feed our input into the transformer encoder, we have to adjust the depth of the signal to let it become D, as well as to downsample the signal to create less input tokens when $L < M$. To downsample, we divide the signal into non-overlapping chunks of size $S = M/L$. In particular, as we will illustrate in Section 5.1, we have evaluated the following alternatives for splitting the signal into chunks and projecting the input to the desired dimension D:

no chunks i.e., $L = M$. We project input scalars to dimension D with a 1D convolution with kernel size 3 and D output channels.

sum We project input scalars to dimension D with a 1D convolution with kernel size 3 and D output channels. We then sum the D-dimensional embeddings of the S samples in a chunk to compute the L input tokens.

cat We project input scalars to dimension D/S with a 1D convolution with kernel size 3 and D/S output channels. We then concatenate the (D/S)-dimensional embeddings of the S samples in a chunk to create the L input tokens with dimension D.

conv We reshape the input signal to have L samples and S channels and project it to dimension D with a 1D convolution with kernel size 3, S input and D output channels.

To map the output tokens computed by the transformer encoder back into four signals with M scalars each, we treat the sequence of tokens as a signal with D samples and L channels and use a 1D convolution with M output channels. Then, we apply a shared linear layer on each of the M samples to compute 4 output channels, i.e., the 4 output signals of the decomposition, from the D channels computed by the network. Fig. 1 describes our architecture with the 1D conv input variant. In particular, the figure shows the architecture with $M = 512$, $D = 512$, and $L = 128$, hence $S = 4$.

Since Transformers are equivariant with respect to the order of the input tokens, but the SD problem is not, we add sinusoidal positional encondings [12] immediately after having performed downsampling and having projected the input samples to D channels, to let the network be aware of the absolute position of individual tokens within the input sequence. To regularize the model, the output of this operation as well as the output of every multi-head SA and of every linear layer in the feed-forward modules are subject to dropout [24] at training time.

To train the network, we minimize the sum of the Mean Square Error (MSE) for each component of a training signal, i.e. the loss for the generic signal $f \in \mathbb{R}^M$ that can be decomposed as $f = c + s + o + n$ is given by

$$\mathcal{L}(c, s, o, n, \hat{c}, \hat{s}, \hat{o}, \hat{n}) = \sum_{x \in \{c,s,o,n\}} \mathrm{MSE}(x, \hat{x}), \tag{5}$$

$$\text{with} \quad \mathrm{MSE}(x, \hat{x}) := \frac{1}{M}\|x - \hat{x}\|_2^2, \tag{6}$$

with x and $\hat{x}$ the true and predicted components, respectively.

4. Dataset generation

Experiments have been performed on synthetically generated datasets of signals. More precisely, each observation $f \in \mathbb{R}^M$ is obtained as formalized in (1), in particular

$$f = c + s + o + n = b_c\bar{c} + b_s\bar{s} + b_o\bar{o} + n = \bar{f} + n, \tag{7}$$

where both the normalized (zero-mean and unitary-variance) versions $\bar{c}, \bar{s}, \bar{o}$ of the cartoon/smooth/oscillatory components c, s, o, are generated pseudo-randomly and their *blending factors* $b_c, b_s, b_o \in [0, 1]$ are chosen from a predefined list.

The smooth and oscillatory components $\bar{s}$ and $\bar{o}$ are generated in a similar way based on their discrete Fourier transform

$$x_i = \sum_{k \in \mathcal{K}} (\alpha_k \cos(2\pi\omega_k i) + \beta_k \sin(2\pi\omega_k i)), \quad i = 0, \dots, M-1, \tag{8}$$

$$\mathcal{K} = \left\{ k_1, \dots, k_n \in \left[k_{min}, k_{max}\right],\ n \in \left[n_{min}, n_{max}\right] \right\}, \tag{9}$$

with frequencies $\omega_k = k/M$ and where n, k_i are generated randomly from uniform discrete distributions, whereas α_k, β_k from a uniform pdf in $[0, 1]$. In particular, the Fourier coefficients of $\bar{s}$ and $\bar{o}$ are generated pseudo-randomly for different frequency-bands (low frequencies for $\bar{s}$, high-frequencies for $\bar{o}$).

The piecewise constant component $\bar{c}$ is characterized in terms of the location and amplitude of its jumps (discontinuities). In particular, the component is generated, from left to right, by selecting for each new jump its distance $d \in [d_{min}, d_{max}]$ from the previous one (or from the first sample, for the first jump) and its amplitude $a = \mathrm{sign}(a)|a|$, $|a| \in [a_{min}, a_{max}]$ with $a_{min} > 0$, $\mathrm{sign}(a) \in \{-1, +1\}$. All variables d, $|a|$ and $\mathrm{sign}(a)$ are uniformly sampled in their domains.

Finally, the AWGN component $n \in \mathbb{R}^M$ is generated by sampling from a Gaussian pdf of M variables with zero mean and diagonal covariance matrix $\sigma_n \mathrm{I}_M$ with standard deviation σ_n that produces a prescribed value of the Signal-to-Noise Ratio (SNR) of the noisy observation $f = \bar{f} + n$, given by

$$\mathrm{SNR}\left(f, \bar{f}\right) = 20 \log_{10} \frac{\left\|\bar{f} - \mathbb{E}\left[\bar{f}\right]\right\|_2}{\|n\|_2}. \tag{10}$$

Table 1
Ablation study. Performed on the reduced dataset with 2000 training samples, 500 validation samples and 500 test samples.

idx	Chunking	Chunk size	Zero-init	lr scheduler	Averaged RMSE ($\times 10^3$) on the test set				
					$\hat{c}$	$\hat{s}$	$\hat{o}$	$\hat{n}$	Avg
1	no	1			5.996	5.933	1.840	2.329	4.024
2	sum	4			6.654	5.900	1.211	4.338	4.526
3	cat	4			7.096	6.164	1.314	3.667	4.561
4	conv	4			6.821	5.572	1.061	3.546	4.250
5	conv	2			6.938	6.133	1.321	2.838	4.308
6	conv	8			6.901	5.994	1.042	4.364	4.575
7	conv	4	✓		6.913	5.722	1.124	3.577	4.334
8	conv	4	✓	✓	7.266	5.809	1.158	3.702	4.484
9	no	1	✓		6.077	5.890	1.526	2.369	3.966
10	no	1	✓	✓	6.215	5.796	1.701	2.531	4.061

Table 2
Experimental results. The first part of the table refers to a reduced test set of 13 signals. The last two lines report the performance of our method on the full test set of 4000 signals.

Method	Test set	Averaged RMSE ($\times 10^3$) on the test set				
		$\hat{c}$	$\hat{s}$	$\hat{o}$	$\hat{n}$	Avg
VSD	13 signals	7.859	5.493	2.666	3.100	4.780
TSD chunks	13 signals	4.248	3.853	0.879	3.020	2.999
TSD no chunks	13 signals	2.983	2.873	0.997	1.762	2.153
TSD chunks	4000 signals	4.107	3.757	0.686	2.924	2.869
TSD no chunks	4000 signals	3.230	3.092	0.843	1.811	2.244

It follows that by corrupting $\bar{f}$ with the realization n of AWGN of variance σ_n^2, then $\|n\|_2^2 \simeq M\sigma_n^2$, hence choosing

$$\sigma_n = 10^{-\mathrm{SNR}/20} \frac{\left\| \bar{f} - \mathbb{E}[\bar{f}] \right\|_2}{M}, \tag{11}$$

leads to an observation f characterized by the prescribed SNR.

5. Experiments

In our experiments, the input size is $M = 512$. Based on our ablation study reported in Section 5.1, we consider two variants of our architecture:

(i) we set the number of input tokens $L = M = 512$, a variant that we refer to as *TSD no chunks*;

(ii) we set $L = 128$, i.e. the chunk size $S = 4$, and use the conv input layer to create the input sequence, a variant that we refer to as *TSD chunks*.

The encoder stack is always composed of $N = 4$ layers. We use the standard values for the inner dimensions $D = 512$, the number of heads $h = 8$, and dropout probability $p = 0.1$. We train our model with ADAM [25] with default hyperparameters for $15k$ epochs, with learning rate 0.0001 and batch size 64. We run all our experiments on a machine equipped with an Nvidia GeForce GTX 1080 with 11 GB of vRAM GPU.

In all our experiments, we generated signals according to the following setup. For $\bar{s}$ and $\bar{o}$ components, in formulas (8)–(9) we used $[n_{min}, n_{max}] = [1, 3]$ for both, whereas $[k_{min}, k_{max}] = [2, 7]$ for s and $[k_{min}, k_{max}] = [70, 80]$ for $\bar{o}$. For $\bar{c}$, we used $[d_{min}, d_{max}] = [40, 50]$ and $[a_{min}, a_{max}] = [0.5, 1]$. The blending factors in (7) for the data set of the 13 signals used to compare the variational method VSD with the proposed TSD approach have been generated to represent a broad range of component mixtures. Specifically, $[b_c, b_s, b_o] =$ [1 0 0; 0 1 0; 0 0 1; 1/3 2/3 0; 2/3 1/3 0; 0 1/3 2/3; 0 2/3 1/3; 1/3 0 2/3; 2/3 0 1/3; 1/3 1/3 1/3; 0.2 0.2 0.6; 0.6 0.2 0.2; 0.2 0.6 0.2]. Finally, for what regards the noise component n, we used SNR = 20 in (11).

We measure the quality of a decomposed component $\hat{x} \in \mathbb{R}^M$ by means of the Root Mean Square Error (RMSE), given by the square root of the MSE defined in (6).

5.1. Ablation study

To understand the impact of our design decisions and define the best training recipe for our model, we performed an ablation study on smaller models with $N = 2$ layers in the encoder, by using a reduced dataset composed of 2000 training samples, 500 validation samples and 500 test samples. While training, we save the model with the lowest average RMSE across components on the validation set, whose results on the unseen test set are then reported in Table 1.

The first four rows compare the different strategies to create chunks of the input signal, i.e. no chunks, sum, cat and conv, while keeping everything else constant. We can see how avoiding to downsample the signals gives the better performance on average. Interestingly, it is also the best at predicting the c, s and n component, while it is the worst at o, the highly-oscillatory one. The best on o, and the second best configuration on average is the conv strategy, with $S = 4$. Since it also enjoys the additional advantage of requiring significantly less memory to perform a forward pass due to the quadratic complexity of Transformers with respect to the sequence length, we consider both variants in the reminder of the ablation.

Rows 5–6 study the effect of the chunk size S for the conv variant. 4 turns out to be the best value.

Finally, rows 7–10 study the effect of two tentative improvements in the training recipe. In our data, in several signals one or more components are missing. Therefore the network has to sometimes output very small values for all samples of a component (ideally, all 0s). To facilitate this task, we have tried to initialize all parameters of the last linear layer responsible for computing the components to 0. This turns out to be slightly better for the no chunks variant (row 1 versus row 9), but it is not for the other one (row 4 versus 7). Hence, we adopt it in subsequent experiments only for the first one. We have also tried to add a lr scheduler [26] to the training recipe, which however was not effective in our experiments (rows 8 and 10).

5.2. Comparison with variational method

We evaluate the accuracy and efficiency performance of the proposed TSD approach and we compare it against the state-of-the-art variational method presented in [11], labeled VSD. For this experiment,

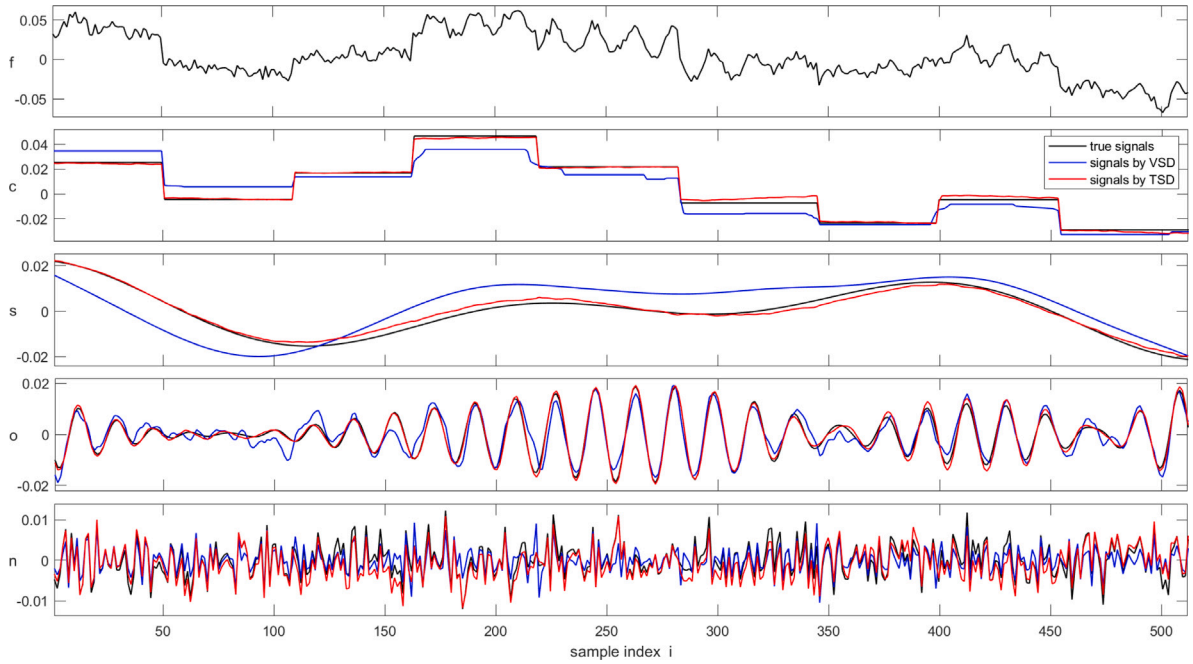

Fig. 2. Visual decomposition results for a signal f from the test set, shown in the top graph: ground truth components (in black) and components estimated by the VSD (in blue) and the proposed TSD data-driven approach (in red). The quality metrics RMSE $\times 10^{-3}$ associated with the estimated components $\hat{c}$, $\hat{s}$, $\hat{o}$, $\hat{n}$ by VSD and the proposed TSD approach are $7.615, 7.466, 3.082, 2.395$ and $1.852, 1.558, 0.963, 2.013$, respectively.

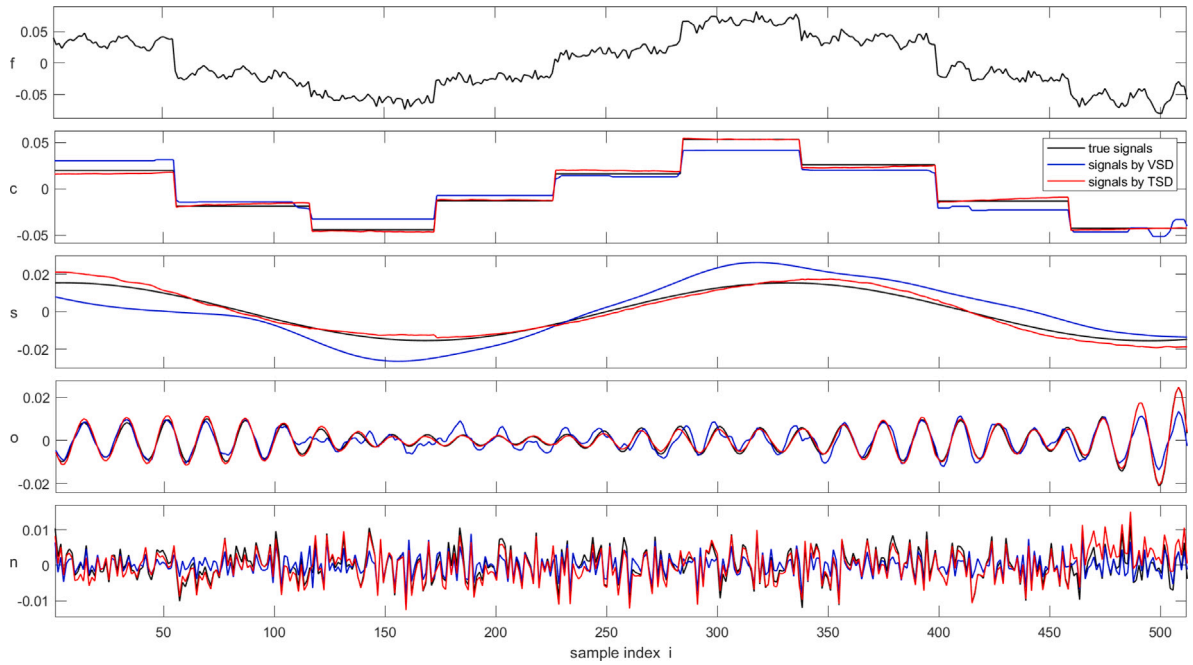

Fig. 3. Visual decomposition results for a signal f from the test set, shown in the top graph: ground truth components (in black) and components estimated by the VSD (in blue) and the proposed TSD data-driven approach (in red). The quality metrics RMSE $\times 10^{-3}$ associated with the estimated components $\hat{c}$, $\hat{s}$, $\hat{o}$, $\hat{n}$ by VSD and the proposed TSD approach are $8.026, 7.588, 2.899, 2.431$ and $2.317, 2.264, 0.872, 1.887$, respectively.

we use a larger dataset, comprising 18 000 samples, which we split into 12 000 samples for training, 2000 for validation, and 4000 for testing.

First, training the TSD network over 12 000 samples took about 48 h for the chunks version, 72 h for the no chunks version. Then, the comparison against VSD has been performed on a reduced test set of 13 samples, namely, one test signal for each blending factor used to generate the dataset. Averaged accuracy (RMSE) results for the four components are reported in Table 2, with average of the four averages reported in the last column labeled Avg. The small size of the reduced test set is due to the long, exhaustive tuning of the hyperparameters required by the VSD approach, which takes more than 30 min for each signal. In fact, in general, variational methods require solving large-size optimization problems by means of iterative numerical methods for each configuration of the hyperparameters. Even if a single run of the optimization algorithm can take only a few seconds for 1D signals with hundreds of samples, the number of hyperparameter configurations to test can be very high in order to obtain optimal results. In the specific case of the compared VSD approach, the computational cost is

dominated by its first stage, aimed to decompose the three components c, s and $o+n$. The two hyperparameters γ_1, γ_2 in the first stage model are 'optimally' set by numerically solving the optimization problem 3500 times for as many different pairs (γ_1, γ_2) and then selecting the pair yielding the minimally cross-correlated decomposed components. This takes, on average, about 30 min. On the other hand, data-driven methods are very fast, that is, our network in the chunks variant requires about 144 ms to process a batch of 128 signals on GPU and 1.6 s on CPU, and does not require to adjust hyperparameters to process different input signals.

Another important practical advantage of the data-driven solution is that it is able to automatically predict the absence of a component, by returning very low values for all its samples. With variational methods, users have to manually specify the components to be estimated, typically after a visual inspection of the input signal.

The results demonstrate the effectiveness of data-driven methods, and in particular of the proposed architecture, for the SD problem considered. The decompositions produced by our approach are more accurate than the output of VSD in all components, while being orders of magnitude faster to run and requiring no per-signal tuning. Indeed, we can seamlessly test our method on a large test set of 4000 in-distribution signals (full set, last row), obtaining similar results, which demonstrate the generalization ability of the model. The best variant in terms of raw performance is *TSD no chunks*, which is however more costly to run in terms of memory and time consumption and worse in estimating the o component, as seen in the ablation study. Indeed, due to the quadratic complexity of attention, the version with no chunks requires about 4 GB of VRAM to process a batch of 128 signals, while the version with chunk size 4 requires about 1.5 GB to process the same mini-batch. Processing time is 400 ms without chunks and 144 ms with chunks. Therefore, both variants can be useful in different application scenarios. To get the best performance regardless of resource consumption, we can also create an ensemble of the two, by using the predicted $\hat{o}$ from the chunked version and the other components from the other network.

Finally, in Figs. 2 and 3 we show some visual results, namely the decompositions obtained on two signals from the test set. Both figures show in the top row the signal f to decompose, whereas the ground truth components and the components estimated by the VSD and the proposed TSD approaches are depicted in the second-to-last rows. Associated RMSE values are reported in the captions. These visual and quantitative results are coherent with those reported in Table 2, and reflect the superiority of the proposed TSD method.

6. Conclusions

In this paper, we investigated the feasibility of applying advanced deep learning techniques to additively decompose a mixture of one-dimensional non-stationary signals. We have shown that a data-driven solution can be very effective, while enjoying practical advantages like no need to tune hyperparameters and the ability to automatically detect which components are present in the input signal. We hope our results serve as a promising foundation for future investigations into SD using transformer architectures. Potential avenues for further explorations include employing more complex and real-world datasets, extending the approach to multidimensional signals, and investigating alternative attention mechanisms to improve efficiency when processing large-scale signals.

CRediT authorship contribution statement

Samuele Salti: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Andrea Pinto:** Software, Methodology. **Alessandro Lanza:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Methodology, Investigation, Data curation, Conceptualization. **Serena Morigi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Investigation, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research of SM has been funded by PNRR CN-HPC, under the NextGeneration EU program CUP J33C22001170001, PNRR MICS, and supported by INDAM-GNCS 2025 projects, and PRIN2022_PNRR_CRESCENTINI CUP J53D23014080001.

The research of AL and SM has been funded by PRIN2022_MORIGI, ANC8HL - CUP J53D23003670006.

Data availability

Data will be made available on request.

References

[1] Y. Chen, H. Zhang, Y. You, J. Zhang, L. Tang, A hybrid deep learning model based on signal decomposition and dynamic feature selection for forecasting the influent parameters of wastewater treatment plants, Environ. Res. 266 (2025) 120615.

[2] Y.-D. Lin, Y.K. Tan, B. Tian, A novel approach for decomposition of biomedical signals in different applications based on data-adaptive gaussian average filtering, Biomed. Signal Process. Control. 71 (2022) 103104.

[3] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. Math. Phys. Eng. Sci. 454 (1998) 903–995.

[4] P. Singh, S.D. Joshi, R.K. Patney, K. Saha, The Fourier decomposition method for nonlinear and non-stationary time series analysis, Proc. R. Soc. A: Math. Phys. Eng. Sci. 473 (2017) 20160871.

[5] W. Zhou, Z. Feng, Y. Xu, X. Wang, H. Lv, Empirical Fourier decomposition: An accurate signal decomposition method for nonlinear and non-stationary time series analysis, Mech. Syst. Signal Process. 163 (2022) 108155.

[6] G. Cai, I.W. Selesnick, S. Wang, W. Dai, Z. Zhu, Sparsity-enhanced signal decomposition via generalized minimax-concave penalty for gearbox fault diagnosis, J. Sound Vib. 432 (2018) 213–234.

[7] Y.-X. Li, S.-B. Jiao, X. Gao, A novel signal feature extraction technology based on empirical wavelet transform and reverse dispersion entropy, Def. Technol. 17 (2021) 1625–1635.

[8] T. Eriksen, N. Rehman, Data-driven nonstationary signal decomposition approaches: a comparative analysis, Sci Rep (1798) (2023).

[9] A. Cicone, M. Huska, S.-H. Kang, S. Morigi, JOT: A variational signal decomposition into jump, oscillation and trend, IEEE Trans. Signal Process. 70 (2022) 772–784.

[10] B.E. Meyers, S.P. Boyd, Signal decomposition using masked proximal operators, Found. Trends Signal Process. (2023) Now Publishers.

[11] L. Girometti, M. Huska, A. Lanza, S. Morigi, Convex predictor–nonconvex corrector optimization strategy with application to signal decomposition, J. Optim. Theory Appl. 202 (2024) 1286–1325.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc, Red Hook, NY, USA, 2017, pp. 6000–6010.

[13] I. Daubechies, J. Lu, H.-T. Wu, Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool, Appl. Comput. Harmon. Anal. 30 (2011) 243–261.

[14] K. Dragomiretskiy, D. Zosso, Variational mode decomposition, IEEE Trans. Signal Process. 62 (2014) 531–544.

[15] M. Huska, S.H. Kang, A. Lanza, S. Morigi, A variational approach to additive image decomposition into structure, harmonic, and oscillatory components, SIAM J. Imaging Sci. 14 (2021) 1749–1789.

[16] M. Huska, A. Lanza, S. Morigi, I. Selesnick, A convex-nonconvex variational method for the additive decomposition of functions on surfaces, Inverse Problems 35 (2019) 124008.
[17] F. Zhou, A. Cicone, H. Zhou, IRCNN+: An enhanced iterative residual convolutional neural network for non-stationary signal decomposition, Pattern Recognit. 155 (2024) 110670.
[18] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, L. Sun, Transformers in time series: A survey, in: E. Elkind (Ed.), Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23, International Joint Conferences on Artificial Intelligence Organization., 2023, pp. 6778–6786, Survey Track.
[19] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: decomposition transformers with auto-correlation for long-term series forecasting, in: Proceedings of the 35th International Conference on Neural Information Processing Systems NIPS '21, Curran Associates Inc, Red Hook, NY, USA, 2021.
[20] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022, arXiv preprint arXiv:2201.12740.
[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, ICLR, 2020.
[22] K. Zaman, K. Li, M. Sah, C. Direkoglu, S. Okada, M. Unoki, Transformers and audio detection tasks: An overview, Digit. Signal Process. 158 (2025) 104956.
[23] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.
[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[25] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, ICLR, San Diega, CA, USA, 2015.
[26] L.N. Smith, N. Topin, Super-convergence: Very fast training of residual networks using large learning rates, 2017, arXiv:1708.07120.