SDD: Shape-aware Data-driven Attention Mechanism for Time Series Analysis

Yanyun Cao
Hong Kong Baptist University
Hong Kong
24478474@life.hkbu.edu.hk

Rundong Zuo
Hong Kong Baptist University
Hong Kong
csrdzuo@comp.hkbu.edu.hk

Rui Cao
Hong Kong Baptist University
Hong Kong
csrcao@comp.hkbu.edu.hk

Byron Choi
Hong Kong Baptist University
Hong Kong
bchoi@comp.hkbu.edu.hk

Jianliang Xu
Hong Kong Baptist University
Hong Kong
xujl@comp.hkbu.edu.hk

Sourav S Bhowmick
Nanyang Technological University
Singapore
assourav@ntu.edu.sg

Abstract

Multivariate time series (MTS) analysis have extensive applications in various areas such as human activity recognition, healthcare, and economics, among others. Recently, Transformer approaches have been specifically designed for MTS and have consistently reported superior performance. In this paper, we demonstrate a software system for a recent efficient shape-aware Transformer (SDD), where time-series subsequences (a.k.a shapes) are made available to users for investigation. First, a time-series Transformer, called SVP-T, takes shapes, together with their variable position information (VP information) as input to the training of a Transformer model. These shapes are computed from different variables and time intervals, enabling the Transformer model to learn dependencies simultaneously across both time and variables. Second, a data-driven kernel-based attention mechanism, called DARKER, reduces the time complexity of training Transformer models from $O(N^2)$ to $O(N)$, where N is the number of inputs. As a result, the training process by using DARKER offers about 3x-4x speedup over vanilla Transformers'. In this demo, we present the first system (SDD) that integrates SVP-T and DARKER. In particular, SDD visualizes the SVP-T's attention matrix and allows users to explore key shapes that have high attention weights. Furthermore, users can use SDD to decide the shape input to train a new model, to further balance between efficiency and accuracy.

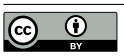
CCS Concepts

• **Human-centered computing** → **Visualization toolkits**.

Keywords

Time Series Analysis, Efficient Transformers, Human-in-the-loop

ACM Reference Format:
Yanyun Cao, Rundong Zuo, Rui Cao, Byron Choi, Jianliang Xu, and Sourav S Bhowmick. 2025. SDD: Shape-aware Data-driven Attention Mechanism for Time Series Analysis. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3746252.3761483



CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761483

1 Introduction

Multivariate time series (MTS) analysis has extensive applications in various areas such as human activity recognition, healthcare, and economics, among others [2, 8, 9]. With the popularity of Transformers in various domains, such as NLP and CV, they have also been applied to MTS. Recently, the time-series Transformer approaches to MTS (*e.g.*, [3, 7, 10, 11]) have exhibited superior performance.

Most Transformer models face two challenges when dealing with MTS. First, they should learn not only the temporal dependencies but also the dependencies between variables. Second, the computational complexity of the Transformer models is high, specifically $O(N^2)$, where N is the length of the input sequences. In particular, directly inputting long time series to the Transformer leads to low efficiency.

Our recent works, named SVP-T [11] and DARKER [10], have been proposed to solve the above challenges. First, SVP-T is proposed to take shapes (such as representative time series subsequences) [3, 10, 11], together with their variable position information (VP information) as input to the Transformer model. These shapes (as opposed to the raw long time series) are derived from different variables and time intervals, enabling the Transformer model to simultaneously learn dependencies across both time and variables. Second, DARKER further reduces the time complexity of Transformer models from $O(N^2)$ to $O(N)$ by proposing a data-driven kernel-based attention mechanism. As a result, the training process of DARKER is about 3x-4x faster than vanilla Transformers.

This demo is the first tool for the aforementioned work. It uses the important downstream task, namely the multivariate time series classification (MTSC), to facilitate a concrete presentation. In particular, the aforementioned works use shapes as inputs; it is natural to visualize the shapes, together with their VP information in a UI, which may shed some insights on the classification results. Furthermore, the demo visualizes the attention weights of the input shapes, with zooming and panning functions, when training a Transformer. Users can further explore and decide to use only the important shapes for training to improve efficiency.

More specifically, we present a software called **S**hape-aware **D**ata-**D**riven attention mechanism for time series analysis (SDD), as shown in Figure 1. In Section 2, we give a system overview of

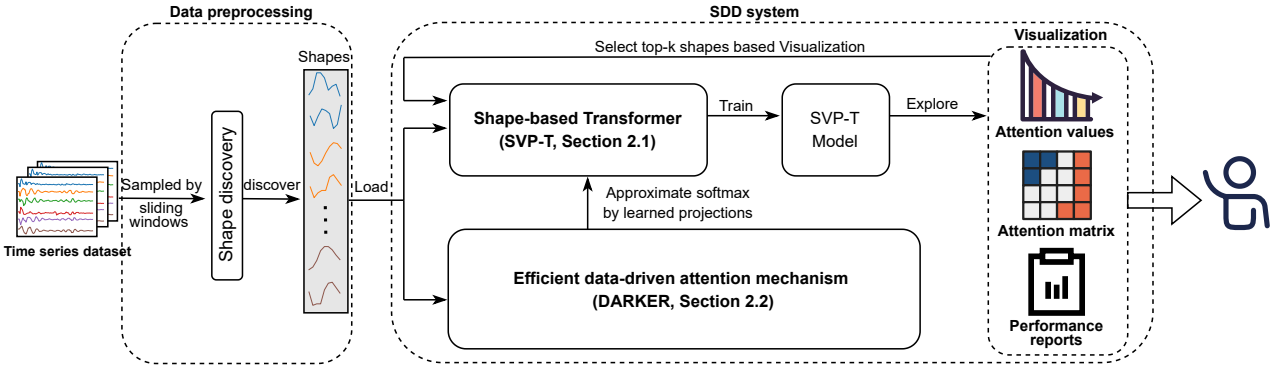


Figure 1: SDD system overview

SDD. Section 3 presents the details and scenarios of demonstration. Finally, Section 4 shows the experiment of choosing top-k shapes to balance the efficiency and accuracy based on SDD.

2 System Overview

Figure 1 presents the system overview of the demonstration software. Foremost, we briefly introduce data preprocessing, in which we apply sliding windows on a time series dataset to extract ample time series subsequences of various lengths, as shown in the LHS of Figure 1. Any shape discovery method can be readily adopted to generate many time series subsequences. Then SDD takes the generated shapes as input to train a shape-based Transformer (Section 2.1) and to learn projections used to approximate softmax (Section 2.2). A series of visualizations are generated, as shown in the RHS of Figure 1. As a result, users can explore the visualizations to examine the shapes that have the highest attention values and the heatmap of the attention matrix. The top shapes, together with the learned projections in DARKER, can be used in the SVP-T model, having higher efficiency and similar accuracies. The performance reports are then provided. Some essential details of SVP-T and DARKER are highlighted in Section 2.1 and 2.2, respectively.

2.1 Highlights of SVP-T

The overview of SVP-T are shown in Figure 2. We follow the same shape discovery in SVP-T to extract shapes from time series. The vanilla Transformer [6] applies sinusoidal position encoding or fully learnable encoding to capture the order of the input sequence. In SVP-T, the shapes' corresponding variable and time interval of the shape (*i.e.*, VP information) are also inputs. Thus, a VP layer is designed to utilize the VP information of time series. The VP information of a specific shape is defined as follows:

$$P_i = \left(\frac{v_i}{V}, \frac{t_{i,\text{start}}}{T}, \frac{t_{i,\text{end}}}{T}\right) \tag{1}$$

where v_i, $t_{i,\text{start}}$ and $t_{i,\text{end}}$ are the variable, the first timestamp, and the last timestamp of S_i, respectively. V is the total number of variables and T is the length of the time series. Then, SVP-T uses a linear layer to learn the VP information when training (shown in Figure 2: Variable-position encoding). Finally, for two shapes from different variables and overlapping in time, they may be important for MTSC and their attention weights should be higher. Thus, SVP-T

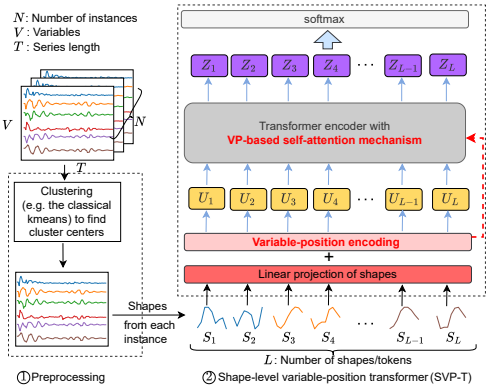


Figure 2: The overview of SVP-T [11]

also allows user to tune the importance of this overlapping behavior in training a model. The overlapping of two shapes S_i and S_j is defined as:

$$O_{\text{lap}}(S_i, S_j) = \begin{cases} \max(\min(t_{i,\text{end}}, t_{j,\text{end}}) - \\ \max(t_{i,\text{start}}, t_{j,\text{start}}), 0) & v_i \neq v_j \\ 0 & v_i = v_j \end{cases} \tag{2}$$

The overlapping of shapes $O_{\text{lap}}(S_i, S_j)$ is then used to design a VP-based self-attention mechanism which enhances the attention weight between S_i and S_j. Finally, the experiment shows that SVP-T has the best accuracy rank compared other SOTAs for MTSC on all UEA time series datasets.

2.2 Highlights of DARKER

A major challenge of the Transformer model is $O(N^2)$ time complexity, which means that when processing long sequences, the computational cost will increase significantly. Random feature attention (RFA) has been a popular approach proposed to improve the efficiency of Transformers [1, 5]. However, RFA methods fail when applied to time series because they rely on a single fixed projection, which does not consider the distribution of input data and leads to large approximation errors. Recently, Zuo et al. proposed *a*

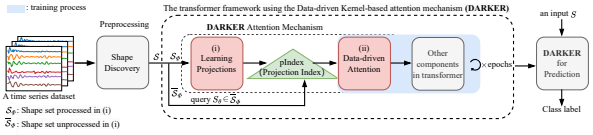


Figure 3: The overview of DARKER[10]

data-driven kernel attention mechanism named DARKER to solve the problem. As shown in Figure 3, DARKER consists of two stages: learning projection and data-driven attention.

First, in learning projection, DARKER employs machine learning models trained on time series data as projections Φ to approximate softmax, instead of directly applying a fixed projection as previous RFA methods. The learned projections are stored and indexed in a projection index named pIndex. Then, in the second stage, a Transformer model is trained based on a data-driven attention mechanism, which is defined as follows:

$$\begin{aligned} DARKERAttn(Q,K,V) &= \Phi(Q,K)V \\ &= \phi_1(S)(\phi_w(W_Q W_K)(\phi_2(S^\top)V)) \end{aligned} \tag{3}$$

where $\Phi = \langle \phi_1, \phi_w, \phi_2 \rangle$ is the projection indexed in pIndex, and S is the input shapes of a time series. For a time series instance, DARKER queries the projection in pIndex for the Q, K, and V matrices. As a result, DARKER not only reduces the time complexity from $O(N^2)$ to $O(N)$, but also achieves comparable accuracies [10].

3 Demonstration Outlines

In this section, we first give a description of our visualization tool named SDD[1] and then describe in detail how to use it in three scenarios. The software is implemented in Python, and its source code, along with four example datasets, is available at the link[2]. SDD consists of two tabs: ① time series and shapes visualization (shown in Figure 4), and ② attention visualization (shown in Figure 5).

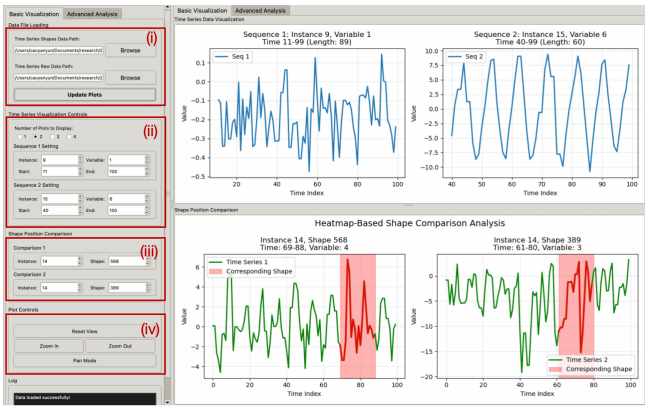


Figure 4: Time series and shapes visualization

Time series and shapes visualization. The first part of our demo is used to display the time series of one variable of a dataset and a shape with its VP information. The user interface is presented in Figure 4. We briefly introduce each of its parts as follows: (i) In the "Data File Loading" panel, the "Load Data Files" button is used to load a specific time series dataset and the shapes of this dataset obtained by shape discovery in Figure 1. The prompt box at the bottom of (iv) displays some statistics of the loaded dataset, *e.g.*, , the number of time series instances, and the length of time series. This allows users to have a quick view of the dataset. (ii) In the "Time Series Visualization Controls" panel, the user selects the number of time series for comparison and exploration. It allows visualizing up to four time series simultaneously. Then, the user specifies which variable ("Variable") of a particular time series instance ("Instance") to be viewed, and determines the time range ("Start", "End") of the time series. (iii) In "Shape Position Comparison", users first choose the shape ID ("Shape") and the corresponding time series instance ("Instance"). (iv) In the 'Plot Controls' panel, there are 4 buttons used to control plots, among which the 'Zoom In' button is used to zoom in on plots. Then, the entire time series is displayed, and the shape is highlighted in red, which is used for SVP-T and DARKER. The VP information corresponding to the shape is also shown.

Attention visualization The second tab shows the attention weights and attention values of input shapes after the Transformer is trained. The GUI is presented in Figure 5. (v) The "Heatmap Controls" canvas shows the "Attention Weight Visualization" on the right. The "Load Heatmap Data" button can be used to load any trained attention weight file, and by selecting "Instance Number" and "Shape Number Range", users can drill into the heatmap of the attention weights of shapes by selecting a range and visualize some selected shapes that occur in a specific instance. In addition, users can view the VP information of the two shapes at the the 'Interactive Comparison' message box by simply clicking on the highlighted point in the heatmap of attention weights, and view the two shapes in the panel of "Shape Comparison Analysis" in the Figure 4. These show some details of the shapes in the Transformer model. (vi) In the "Attention Controls" panel, users can load the trained attention value file through "Load Attention Data". In addition, the user can select the instance number they want to view through the "Instance Number" button, and can select the number of shape attention values (sorted in descending order) by clicking the "Number of Shapes" button. Finally, the shape ID file can be exported by clicking the "Export Top Shapes Indices" button. Thus, we can use the top shapes and pass them to train SVP-T and get a new model. By skipping some shapes that have low attention, we efficiently train a Transformer and still obtain good accuracies. Next, we present the demonstration scenarios to highlight what ideas participants will experience.

Scenario 1: Visualization time series of BasicMotions. A participant of the demonstration would like to view the time series of different variables from the famous dataset BasicMotions. This dataset records 3D accelerations and 3D gyroscopes when four students wear smart watches and perform four activities (walking, resting, running and badminton). He clicks on "Load Data File" to load the time series and shapes of dataset BasicMotions. Next, he selects the time series to be compared, such as comparing different students in the same activity, or comparing different activities of

[1]An online video of this demonstration software can be found at https://youtu.be/c05TOFqU_xQ.
[2]https://github.com/CaoYanyun1225/VISA.

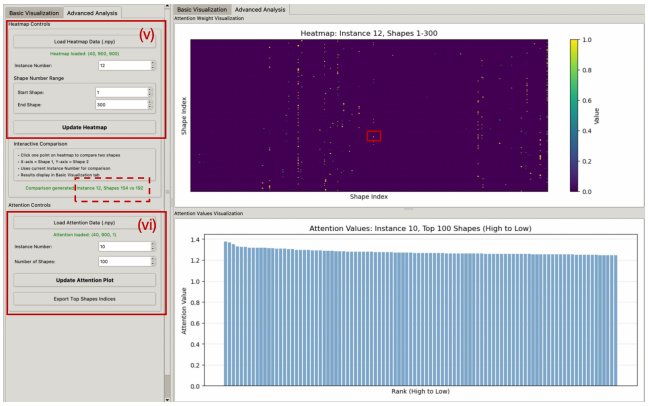


Figure 5: Attention visualization

the same student. He can load the complete time series he wants to view into the "Time Series Data Visualization" section. In addition, he can freely select the plots at different start and end times he wants to view through the two option bars "Start" and "End".

Scenario 2: Visualization of shapes and other input to Transformer training. Next, a participant would like to view the shapes obtained by shape discovery, their corresponding VP information and overlapping. He can click the "Update" button to load the shape he wants to view into the "Shape Position Comparison" and to observe the VP information of the shape above the plots. For example, he wants to observe two different shapes within the first time series instance, specifically the 237th shape and the 43th shape. He can set the "Instance" at the bottom of Figure 4 to 1, and set the "Shape" to 237 and 43 respectively. Subsequently, these two shapes will be displayed on the right side (as shown in the lower right part of Figure 1). At the same time, the VP information corresponding to the shape will be displayed above the plotted shapes. With these, the user understands clearly what are the input of the models.

Scenario 3: Exploration of the relations between different shapes by attention visualization. In a scenario where a user wants to find the relationships between different shapes, they can do so by observing the attention weight matrix of the trained Transformer. He can select the instance to be observed through the "Instance Number" button after loading the data. Then, by viewing the attention weight heatmap in the RHS of Figure 5, it can be observed in the heatmap that the point between the 237th shape and the 43rd shape is relatively bright (shown in the red circle of Figure 5), indicating a great importance between these two shapes. Meanwhile, after clicking this point, the VP information of the two shapes is shown in the red dashed box. Then, we utilize their VP information to track them back to the original time series instance in the first tab of our demo. As shown in Figure 4, we observe that the two shapes are from different variables, but have an overlapping in time. This indicates the effectiveness of VP-based self-attention mechanism in SVP-T. The above-mentioned investigation of the internals of a shape-based time-series Transformer is enabled by this demo. One case study of this scenario is provided in SVP-T [11].

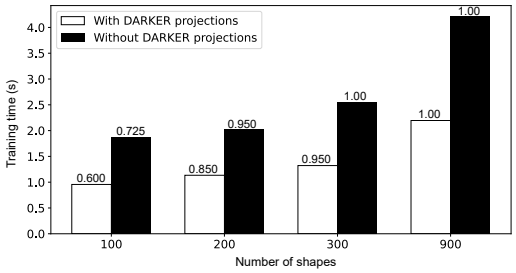


Figure 6: Using learned projections in DARKER or not

4 Highlights of Experiments

In this section, we present an experiment to show whether selecting the top-k shapes through our demo, instead of using all shapes, can improve efficiency while still maintaining accuracy. We apply the data-driven attention mechanism in DARKER [10] and use top-k shapes exported from Scenario 3 for training SVP-T, where the top-k shapes are simply sorted by the attention values in descending order. We choose the dataset BasicMotions for consistency of this demo. The experiment is conducted on a NVIDIA V100S GPU. The details of datasets and parameters can be found in the link[3].

The experimental results are shown in Figure 6. We vary the number of top-k shapes along the x-axis and report both training time (y-axis) and accuracy (indicated by the number above each bar). The results demonstrate that the training times using DARKER learned projections are about twice as fast as those without using them. We also found that a greater number of shapes generally leads to higher accuracy but results in lower efficiency. This is because more shapes increase the features learned by SVP-T, but they also extend the input sequences, thereby increasing the training time. Specifically, the results show that using top 300 shapes achieves an accuracy of 0.95, which is only 0.05 less than the accuracy achieved using all 900 shapes. However, using fewer shapes reduces training time by 2.5 times. Users can use the saved time to explore more combinations of hyperparameters. This demonstrates that by selecting the top-k shapes through our demo, we can achieve high efficiency while maintaining similar levels of accuracy. Finally, our results suggest a research opportunity: future time series Transformer models could achieve high accuracy with less training time by selecting high-quality shapes as input [4].

Acknowledgments

Thanks for the code review from Silver, Ho-Fai, Liu. This work was supported by the Hong Kong Research Grant Council (HKRGC), RIF R2002-20F.

References

[1] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2021. Rethinking attention with performers. In *ICLR*.

[2] Chang George Dong, Zhengyang David Li, Liangwei Nathan Zheng, Weitong Chen, and Wei Emma Zhang. 2024. Boosting Certificate Robustness for Time Series Classification with Efficient Self-Ensemble. In *CIKM*. 477–486.

[3] https://github.com/rdzuo/darker.

[3] Guozhong Li, Byron Choi, Sourav S. Bhowmick, Grace Lai-Hung Wong, Kwok-Pan Chun, and Shiwen Li. 2020. Visualet: Visualizing Shapelets for Time Series Classification. In *CIKM*. 3429–3432.
[4] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace LH Wong. 2021. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *AAAI*. 8375–8383.
[5] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2021. Random Feature Attention. In *ICLR*.
[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
[7] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A Transformer-Based Framework for Multivariate Time Series Representation Learning. In *ACM SIGKDD*. 2114–2124.
[8] Junru Zhang, Lang Feng, Yang He, Yuhan Wu, and Yabo Dong. 2023. Temporal Convolutional Explorer Helps Understand 1D-CNN's Learning Behavior in Time Series Classification from Frequency Domain. In *CIKM*. 3351–3360.
[9] Weiqi Zhang, Jianfeng Zhang, Jia Li, and Fugee Tsung. 2023. A Co-training Approach for Noisy Time Series Learning. In *CIKM*. 3308–3318.
[10] Rundong Zuo, Guozhong Li, Rui Cao, Byron Choi, Jianliang Xu, and Sourav S Bhowmick. 2024. DARKER: Efficient Transformer with Data-Driven Attention Mechanism for Time Series. *PVLDB* (2024), 3229–3242.
[11] Rundong Zuo, Guozhong Li, Byron Choi, Sourav S Bhowmick, Daphne Ngai-Yin Mah, and Grace Lai-Hung Wong. 2023. SVP-T: A Shape-Level Variable-Position Transformer for Multivariate Time Series Classification. In *AAAI*. 11497–11505.