TOWARDS DIVERSE PERSPECTIVE LEARNING WITH SWITCH OVER MULTIPLE TEMPORAL POOLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pooling is a widely used method for classification problems. In the time series classification (TSC) domain, pooling considering temporal information has been proposed. However, we found that each temporal pooling has a distinct and fixed perspective, which causes data dependency. In this paper, we propose a novel pooling architecture for diverse perspective learning: switch over multiple pooling (SoM-TP). SoM-TP dynamically selects the optimal pooling method suitable for each data. The massive case study using layer-wise relevance propagation (LRP) reveals the distinct perspective each pooling has and ultimately demonstrates the need for diverse perspective learning. The ablation study on SoM-TP shows how diverse perspective learning is achieved and performed. Furthermore, the pooling classification experiment also supports the need for diverse perspective learning by showing that more suitable pooling exists depending on the data. Extensive experiments are done with the UCR/UEA repository.

1 INTRODUCTION

Time series classification (TSC) has been one of the most valuable subjects in data mining (Längkvist et al., 2014). Also, the revolutionary success of deep neural networks (DNN) has led to their application on TSC (Wang et al., 2017; Le Guennec et al., 2016; Zhao et al., 2017; Tanisaro & Heidemann, 2016; Serrà et al., 2018). Especially the convolutional neural networks (CNN) based model architecture, fully convolutional networks (FCNs) (Long et al., 2015) and residual networks (ResNet) (He et al., 2016), achieved the current state-of-the-art (SOTA) in an end-to-end manner (Ismail Fawaz et al., 2019; LeCun et al., 2015).

In CNN, pooling is the key component with two primary purposes: 1) Decreasing the number of parameters for less computational cost and preventing overfitting, and 2) Position invariance learning. To this end, pooling combines the high-dimensional values of feature outputs into low-dimensional (Gholamalinezhad & Khosravi, 2020). Thus, global pooling is widely used as the most suitable pooling method for the above purposes.

However, a simple structure of global pooling has a drawback in the time series domain by losing the temporal information of hidden vectors. To solve this limitation, temporal poolings have been proposed to conserve temporal position information from convolutional hidden vectors(Lee et al., 2021). Global temporal pooling (GTP) outputs only one position-invariant feature, which is a simplified version of common global pooling. However, time-segmented pooling methods, static-temporal-pooling (STP) and dynamic-temporal-pooling (DTP) use the given number $\mathbf{n} \in \mathbb{Z}^+$ to segment the time-axis by temporal order. STP divides equally, while DTP does dynamically with different segmentation lengths via soft dynamic time warping (soft-DTW) (Lee et al., 2021). Due to the segmentation, STP and DTP represent multiple local features, whereas GTP represents only one. With multiple local features, the temporal order information in each segment is conserved by each pooled vector.

How to aggregate convolution features in pooling is a significant matter. Each temporal pooling has a distinct mechanism for aggregation, and we term the different mechanisms of temporal pooling as a 'perspective'. Depending on the use of segmentation in pooling, the perspective is divided into 'global' and 'local', and according to the segmentation method, the local is divided into 'rigid' and 'dynamic'. However, each temporal pooling only deals with a single perspective on hidden features as defined. And this fixed-perspective learning has the following constraints. A global view cannot

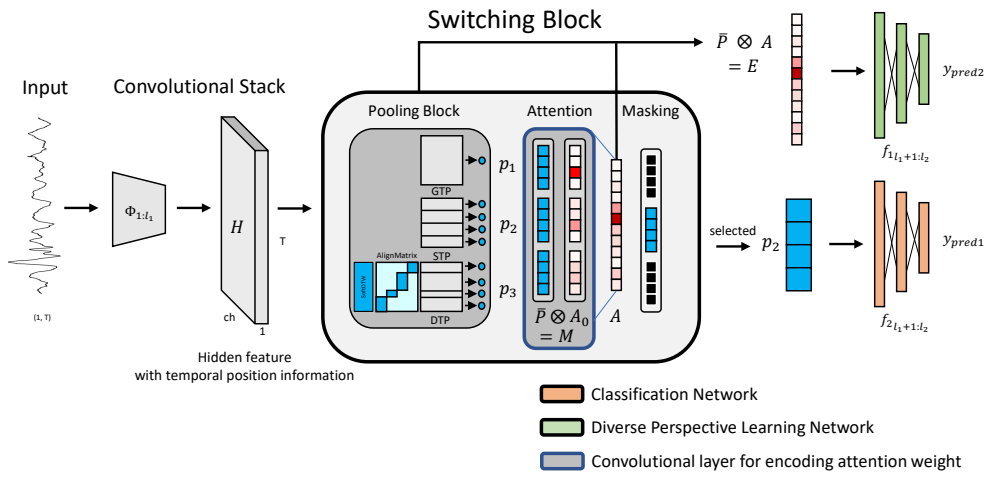


Figure 1: **SoM-TP architecture.** After a convolutional stack, each temporal pooling compresses the hidden features that conserve temporal position information. The pooling block consists of three types of temporal pooling: GTP, STP, and DTP, then it outputs pooled vectors (blue square, $\bar{\mathbf{P}} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$) in different perspectives. The attention weight, which is a learnable parameter, exists for selecting proper pooling by each data. The attention weight ($\mathbf{A_0}$) is multiplied by all pooled vectors ($\bar{\mathbf{P}}$) to reflect all the perspectives of the pooling block, resulting in a weighted pooled vector ($\mathbf{M}$). The final attention score ($\mathbf{A}$) is the encoded weight vector formed after a weight pooled vector ($\mathbf{M}$) passes through the convolutional layer (blue box). Then, this final attention score ($\mathbf{A}$) selects the best poolings in the block by the score value. Next, the parameters are updated with the following procedure; 1) a diverse perspective learning network (green) uses ensembled input ($\mathbf{E}$) which is the multiplication of the final attention score ($\mathbf{A}$) and all the pooled vectors ($\bar{\mathbf{P}}$); 2) The main network with (orange) gets only selected pooled vector (in here, $\mathbf{p}_2$) from one pooling method; 3) Each network predicts y_{pred1} and y_{pred2} respectively, and y_{pred2} works as a regularizer; 4) With these two outputs, SoM-TP is optimized with various perspectives.

capture multiple classification points, while a local view cannot concentrate on one dominant classification point. Therefore, the performance loss is inevitable if the task requires diverse viewpoints to simultaneously capture a dominant feature and hidden local features. Motivated by the limitations, the novel approach of pooling to fully utilize diverse perspectives is suggested.

In this paper, we propose "Switch over Multiple Temporal Pooling (SoM-TP)", a pooling architecture with diverse perspective learning. Diverse perspective learning is the opposite concept of fixed-perspective learning, which can overcome the limitation of existing temporal poolings. SoM-TP can reflect various views by dynamically selecting proper pooling based on attention. The challenge here is to train attention weight to dynamically select the optimal pooling for each specific data sample. For this, we propose three methods: 1) convolutional encoded attention weight; 2) diverse perspective learning network which is a sub-network to deal with various poolings; 3) perspective loss which is the combination of KL divergence and cross-entropy loss function.

The attention score selects proper pooling by highlighting important pooled vector's index. To get an attention score, the convolutional layer encodes the vectors of multiplying pooling features and attention weight to compress the information from all pooling features. Also, the sub-network and perspective loss is for effective optimization to SoM-TP to learn diverse perspectives. Next, by minimizing perspective loss from the KL divergence between each output from the main network and a diverse perspective learning network, all learnable weights in the SoM-TP learns diverse perspectives. By making the two output distributions similar, the model learns various views, which is considered as using the concept of parameter tying. This process makes a model robust by learning any pooled observations from different views.

The research scope is limited to temporal pooling methods with CNN. A model architecture with FCN and ResNet, customized with three primary temporal pooling schemes, is specifically used.

2 BACKGROUND

2.1 DEFINE THE PERSPECTIVE OF TEMPORAL POOLINGS

Convolutional Neural Network in Time Series Classification CNN outperformed conventional methods such as nearest neighbor classifiers (Yuan et al., 2019a) or COTE (Bagnall et al., 2015; Lines et al., 2016) by capturing local patterns in TSC problems (Ismail Fawaz et al., 2019; Wang et al., 2017).

As for TSC, the CNN model can be generally formulated as follows: a time series data $\mathbf{T} = \{(\mathbf{X}^1, y^1), ..., (\mathbf{X}^t, y^t)\}$, where $\mathbf{X} \in \mathbb{R}^{\mathbf{d}\times\mathbf{t}}$ of length $\mathbf{t}$ with $\mathbf{d}$ variables and $y \in \{1, ..., C\}$ from C classes. Then, convolution stack Φ of out channel dimension $\mathbf{k}$ encodes features as hidden representations with temporal position information $\mathbf{H} = \{h_0, ..., h_t\} \in \mathbb{R}^{\mathbf{k}\times\mathbf{t}}$ (Lee et al., 2021).

$$\mathbf{H} = \Phi(\mathbf{T}) \tag{1}$$

Global Temporal Pooling. GTP is a global pooling by using $\mathbf{H}$ as input. GTP pools just one representation $\mathbf{p_1} = [p_1] \in \mathbb{R}^{\mathbf{k}\times\mathbf{1}}$ in whole time range with ignoring temporal information. GTP entirely aggregates the hidden convolution output features by the time axis $\mathbf{t}$. As a result, GTP has a global view since the whole time axis information is represented as one pooled vector.

$$\mathbf{p}_1 = \phi_1(\mathbf{H}) \tag{2}$$

However, GTP has the crucial limitation that it cannot contain sequential information which can be an important feature. To utilize temporal position from $\mathbf{H}$, other temporal pooling methods have been proposed (Figure 1 with a pooling block) (Lee et al., 2021).

Both STP and DTP are multiple local pooling methods with time-axis segmentation. The main difference between the two methods is whether to consider a temporal relationship which means the relationship between each time steps in a time series data. Considering that time series data generally consist of a combination of specific local patterns, the reflection of temporal relationship can be effective in terms of loss of meaningful sequential information.

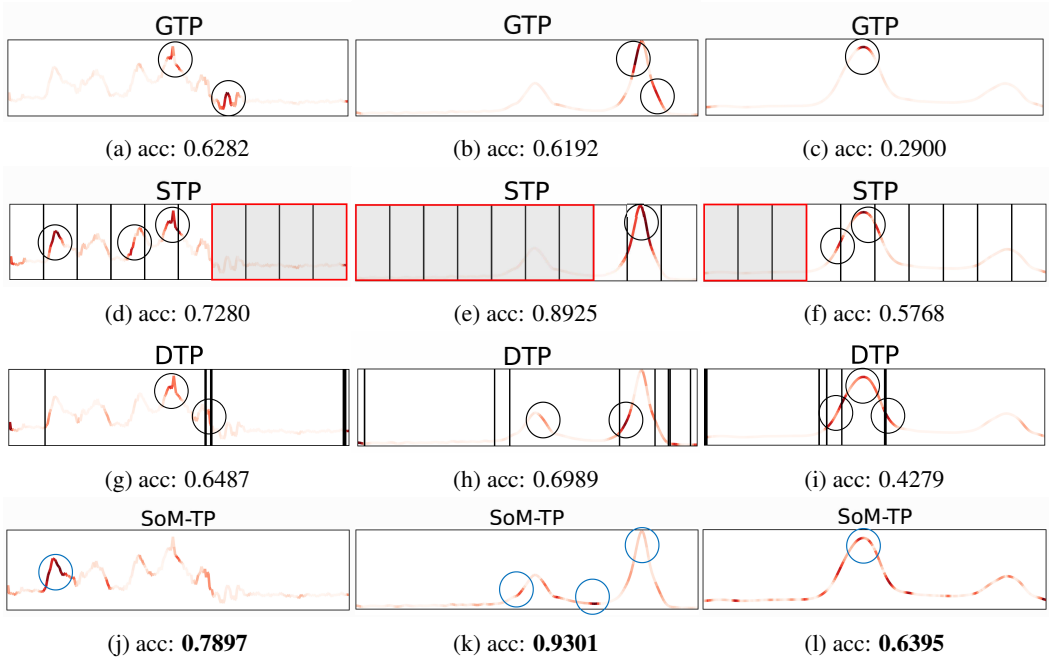
Static Temporal Pooling. STP divides the time axis equally into $\mathbf{n}$ segments with a length $\ell = \frac{t}{n}$. In other words, STP ϕ_2 pools each of the segments $\mathbf{h}_\ell$, where $\bar{\mathbf{H}} = \{\mathbf{h}_{\mathbf{0}:\ell}, \mathbf{h}_{\ell:\mathbf{2}\ell}, ..., \mathbf{h}_{(\mathbf{n}-\mathbf{1})\ell:\mathbf{n}\ell}\}$ (Lee et al., 2021). Note that $\mathbf{h}_\ell$ keeps local temporal information within each time segment, but there is no consideration of temporal relationship in the segmentation process. The pooling representations increase as $\mathbf{p_2} = [p_1, ..., p_n] \in \mathbb{R}^{\mathbf{k}\times\mathbf{n}}$. In terms of the pooling perspective, STP has a rigid local view since the whole time axis is represented by $\mathbf{n}$ equally segmented pooled vectors.

$$\mathbf{p}_2 = \phi_2(\bar{\mathbf{H}}) \tag{3}$$

Dynamic Temporal Pooling. DTP is the layer that is optimized by soft-DTW to segment the time axis while considering temporal relationship (Algorithm 1 in Appendix B). The DTW distance is calculated by point-to-point matching with temporal consistency,

$$\begin{aligned} \mathbf{DTW}\gamma(X, Y) &= \mathbf{min}\gamma\{\langle A, \Delta(X, Y)\rangle, \forall A \in \mathcal{A}\}, \\ \mathbf{min}\gamma\{a_1, ..., a_n\} &= \begin{cases} \text{min} i \le n a_i, & \gamma = 0 \\ -\gamma \text{log} \sum_{i=1}^{n} e^{-a_i/\gamma}, & \gamma > 0, \end{cases} \end{aligned} \tag{4}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are time series with lengths $\mathbf{t_1}$ and $\mathbf{t_2}$, and the cost matrix $\Delta(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{\mathbf{t_1}\times\mathbf{t_2}}$ represents the distance between $\mathbf{X}_{\mathbf{t_1}}$ and $\mathbf{Y}_{\mathbf{t_2}}$. $\mathbf{DTW}$ is defined as the minimum inner product of the cost matrix with any binary alignment matrix $\mathcal{A} \in \{0, 1\}^{\mathbf{t_1}\times\mathbf{t_2}}$ (Lee et al., 2021; Cuturi & Blondel, 2017). With the soft-DTW algorithm, a similar time sequence is grouped with different lengths. In this way, $\mathbf{H}$ is segmented in diverse time length $\bar{\ell} = [\ell_1, \ell_2, ..., \ell_n]$, where $\mathbf{t} = \sum \bar{\ell}$. Finally, DTP ϕ_3 effectively optimizes all the parameters, and the optimal pooled vectors $\mathbf{p}_3 = [p^{\ell_1}, ..., p^{\ell_n}]$ is pooled from each of $\mathbf{h}_{\bar{\ell}}$, where $p^{(\bar{\ell})} = \phi_3(\mathbf{h}_{\bar{\ell}})$. Note that DTP has a dynamic local view with $\mathbf{n}$ segmentation with different lengths, which is formed by considering the temporal relationship between time steps.



(a) acc: 0.6282 (b) acc: 0.6192 (c) acc: 0.2900

(d) acc: 0.7280 (e) acc: 0.8925 (f) acc: 0.5768

(g) acc: 0.6487 (h) acc: 0.6989 (i) acc: 0.4279

(j) acc: **0.7897** (k) acc: **0.9301** (l) acc: **0.6395**

Figure 2: **Input attribution comparison between fixed and diverse perspective learning.** The figure contains three different datasets in the UCR repository; CricketZ, Fungi, and WordSynonyms by each column. The performance of independent data with each temporal pooling varies and is highly dependent on each perspective. First, GTP focuses on dominant and extreme points as seen in all datasets (a), (b), and (c). Second, STP has n segments that can catch multiple classification points as (d). However, the compulsory segmentation to equal lengths has a high tendency of including unnecessary segmentation (gray with red box) as in all examples of STP (d), (e), and (f). Third, DTP has dynamic segmentation and forms more efficient segments. However, this segmentation sometimes downgrades the performance because it divides important classification points as (g) and (i). Both examples show that DTP divides the classification point SoM-TP highlights. SoM-TP overcomes these limitations of each temporal pooling by global and local mixed perspectives. As shown in (k) and (l), SoM-TP focuses on the humping point in a green circle as other poolings do (global view), but also focuses on other lower points (local view) and leverages the performance by diverse perspectives. Due to the flexibility of SoM-TP, it can also focus on the global important point as (j). Note that the black circle is where pooling is highly focused and the green circle is SoM-TP is highly focused on by new perspective.

2.2 Constraints of Temporal Pooling Methods

Each temporal pooling has different perspectives based on different mechanisms. And only one viewpoint exists, we define it as fixed-perspective learning. Through qualitative analysis with layer-wise relevance propagation (LRP) and loss landscape, we examine how the mechanisms of each pooling operate in the model to capture time series patterns.

LRP is a gradient-based method of computing input attribution for input features to explain network predictions (Binder et al., 2016b;a; Bach et al., 2015). The input attribution from LRP shows how each temporal poolings focus on different points: LRP z^+ rule for $\Phi_{1:l_1}$ and ϵ rule for $f_{l_1+1:l_2}$.

GTP perspective. GTP focuses on dominant features with a global view. However, GTP cannot capture multiple classification points dispersed on a time axis. Thus, the global view has constraints on complex time series data.

STP Perspective. STP captures multiple local features. However, STP segments important consecutive patterns by compulsory segmentation of equal length. Also, STP pools the unimportant segmentations. This inefficiency causes representation power to be distributed loosely, leading to a performance loss.

DTP Perspective. DTP has the most complexity by optimizing segmentation length dynamically. This flexibility enables the model to fully utilize segmentation power. However, since DTP is based on distance similarity, segmentation occurs at the change point. For this reason, DTP is hard to capture the consecutive pattern divided at the change point, such as an inflection point.

3 SoM-TP: Towards Diverse Perspective Learning

Diverse perspective learning is a new concept, fully utilizing all the views from each temporal pooling. Note that one fixed view, which is fixed-perspective learning done by individual temporal pooling, causes the model to have data dependency and performance degradation. However, SoM-TP can overcome these limitations with diverse perspective learning.

We first introduce how SoM-TP obtains valuable information from multiple poolings. Next, we analyze the performance of SoM-TP in two directions: 1) What are the mechanisms that make the model learn diverse perspectives in SoM-TP? and 2) Is SoM-TP more robust than other pooling methods?

3.1 Switch over Multiple Pooling

SoM-TP is a framework to induce diverse perspective learning to achieve robustness through the pooling method. To leverage existing temporal poolings, SoM-TP needs three components; attention score, ensemble network, and perspective loss. Through these components, SoM-TP selects proper pooling by specific time series samples in one dataset.

Learnable attention weight The critical component of SoM-TP is learnable attention weight vectors for scoring the importance of each temporal pooling, which allows a single model to learn diverse perspectives. The purpose of the attention score is to properly select a specific pooling for data samples in each mini-batch unit. For data-specific learning, learnable attention weights and a convolutional layer are used to calculate the attention score.

The attention score is derived from pooled vectors and one convolutional layer. The concatenated pooled vector $\bar{\mathbf{P}} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3] \in \mathbb{R}^{k \times 3 \cdot \mathbf{n}}$, where $\mathbf{p} \in \mathbb{R}^{k \times \mathbf{n}}$ from each temporal pooling, and attention weight vector $\mathbf{A}_0 \in \mathbb{R}^{1 \times 3 \cdot \mathbf{n}}$ are multiplied as the input of convolutional layer. For the attention weight vector, $\mathbf{A}_0$ is the parameter to weight pooled vectors to highlight important pooling. The multiplied input $\mathbf{M} \in \mathbb{R}^{k \times 3 \cdot \mathbf{n}}$ are encoded to attention score $\mathbf{A} \in \mathbb{R}^{1 \times 3 \cdot \mathbf{n}}$ through the convolutional layer. The final attention sore is normalized with softmax, $y_j = \frac{exp(x_j)}{\sum_i exp(x_i)}$, where $x \in \mathbf{A}$, to compare the contribution of each feature from temporal poolings. For selecting proper pooling based on attention, there are two options: 1) One pooling output $\mathbf{p} \in \bar{\mathbf{P}}$ with a maximum y_j is selected, or 2) average $y_n = \frac{\sum_n y_j}{n}$ is proceeded in each pooling output, in the unit of length $\mathbf{n}$ which is pooled number, then the pooling output with the highest average y_n is selected.

Diverse perspective learning network and KL Divergence. The SoM-TP framework has two fully-connected networks; the main decision network which is termed a 'classification network' and a sub-network which is termed a 'diverse perspective learning network (DPLN)'. To optimize attention weight and the whole model to learn diverse perspectives, a new loss function is introduced. We define "perspective loss" which is the addition of DPLN loss and KL-divergence. While the classification network only focuses on minimizing the whole model of cross-entropy loss with selected pooling for each mini-batch, DPLN minimizes the classification loss with all pooled vectors. In detail, the ensemble output $\mathbf{E} = \mathbf{A} \times \bar{\mathbf{P}} \in \mathbb{R}^{k \times 3 \cdot \mathbf{n}}$, from concatenated pooling vectors $\bar{\mathbf{P}}$ multiplied by attention $\mathbf{A}$, is given to DPLN. $\mathcal{L}_{DPLN}$ is included in $\mathcal{L}_{perspective}$ to optimize attention $\mathbf{A}$ with considering the classification result of DPLN.

The Kullback-Leibler divergence (KL-divergence) is specially used as a regularizer in the perspective loss. Through the optimization process, KL-divergence pulls y_{pred1} to y_{pred2} to reflect all perspectives, which prevents a single perspective pooling to be dominant.

The perspective loss driven by DPLN is defined as follows,

$$
\begin{aligned}
&KL(y_{pred1}, y_{pred2}) = y_{pred2} \cdot log\frac{y_{pred2}}{y_{pred1}}, \\
&\mathcal{L}_{DPLN}(\{\mathcal{W}_0\}, \{\mathbf{W}^{(p)}\}) = -\frac{1}{t}\sum n = 1^t logP(y = y^t|\mathbf{X}^t), \\
&\mathcal{L}_{perspective} = KL(y_{pred1}, y_{pred2}) + \mathcal{L}_{DPLN},
\end{aligned}
\tag{5}
$$

where input time series $\{(\mathbf{X}_1, y_1), ..., (\mathbf{X_t}, y_t)\}$, Φ with learnable parameter $\mathcal{W}_0$ of CNN, $y_{pred1} \in \mathbb{R}^{1\times c}$ from the classification network $\mathbf{W}^{(c)}$, and $y_{pred2} \in \mathbb{R}^{1\times c}$ from the DPLN $\mathbf{W}^{(p)}$. We define pooling weight matrix $\mathbf{W}^{(p)} = [\mathbf{w}_1^{(p_1)}, ..., \mathbf{w}_{3\mathbf{n}}^{(p_3)}] \in \mathbb{R}^{K\times 3\cdot\mathbf{n}}$, where $\mathbf{w}^{(p)} \in \mathbb{R}^k$ that weighted importance of each latent dimension for pooling $\mathbf{p}$ by fully-connected layer, whereas $\mathbf{W}^{(c)} = [\mathbf{w}_1^{(c)}, ..., \mathbf{w}_{\mathbf{n}}^{(c)}] \in \mathbb{R}^{k\times\mathbf{n}}$ is the class weight matrix (Lee et al., 2021).

Therefore, the final loss function of the SoM-TP framework is defined as follows,

$$
\begin{aligned}
&\mathcal{L}_{classification}(\{\mathcal{W}_0\}, \{\mathbf{W}^{(c)}\}) = -\frac{1}{t}\sum_{n=1}^{t} logP(y = y^t|\mathbf{X}^t), \\
&\mathcal{L}_{cost}(\{\mathcal{W}_0\}, \{\mathbf{W}\}) = \mathcal{L}_{classification} + \lambda \cdot \mathcal{L}_{perspective},
\end{aligned}
\tag{6}
$$

where $\mathbf{W} \in \{\mathbf{W}^{(c)}, \mathbf{W}^{(p)}, \mathbf{A}\}$ of learnable parameters. With classification accuracy as a priority, the loss of the classification network $\mathcal{L}_{classification}$ from y_{pred2} is calculated through cross-entropy, and $\mathcal{L}_{perspective}$ is added as λ decay. Therefore, CNN with learnable parameter $\mathcal{W}_0$ adopts two fully-connected layers relatively while minimizing the similarity between each network output, as the concept of parameter tying.

Furthermore, SoM-TP proceeds with additional optimization for $\mathbf{A}$. The dot product similarity is considered to regulate $\mathbf{A}$. $\mathcal{L}_{attn}$ is learned in the direction of lowering the dot product similarity between the outputs of the classification network and the DPLN.

$$
\begin{aligned}
&\mathcal{L}_{attn} = y_{pred1} \cdot y_{pred2}, \\
&\mathbf{A} \leftarrow \mathbf{A} - \eta \cdot \partial\mathcal{L}_{attn}/\partial\mathbf{A},
\end{aligned}
\tag{7}
$$

$\mathcal{L}_{attn}$ plays a role similar to KL-divergence in perspective loss but directly affects only attention weight.

Shortly, SoM-TP learns various perspectives through attention, DPLN, and perspective loss. Different pooled features affect convolutional and fully-connected networks to learn weight in distinct ways with individual pooling. This is the most significant distinction between SoM-TP and fixed-perspective learning.

3.2 Ablation study on SoM-TP

The novel point of SoM-TP is to make diversification of pooling selection that fits specific data even in one dataset. Therefore, the ablation study is done to show how diverse perspective learning occurs.

3.2.1 Experimental Setting

For extensive evaluation, 112 univariate and 21 multivariate time series datasets from the UCR/UEA repositories are used (Bagnall et al., 2018; Dau et al., 2019); collected from a wide range of domains and publicly available.

CNN classifiers with pooling layers are built on a common ground of model architecture. FCN and ResNet are specifically designed for TSC (Wang et al., 2017), and three different temporal poolings are customized with settings: normalization with BatchNorm (Ioffe & Szegedy, 2015), activation

Table 1: Detailed experimental settings.

Network	#Conv.	#Pooled Prototypes		#FC.	Optim.	lr	batch	window	epoch
		GTP	STP&DTP						
FCN ResNet	3 9	1	$\mathbf{n}$	3	Adam	$1e-4$	8	1	300

Table 2: Model capacity and performances. The best performance of SoM-TP is **bolded** and best performance of other temporal poolings are underlined.

CNN	# Params. of Conv.	POOL (type)		# Params. after Conv.	UCR (uni-variate)		UEA (multi-variate)	
					acc	f1macro	acc	f1macro
FCN	363,520	GTP	MAX	$(\mathbf{1}\times 256\times 512)+527,874$	0.7077	0.6663	0.6516	0.6214
			AVG		0.7227	0.6902	0.6642	0.6388
		STP	MAX	$(\mathbf{n}\times 256\times 512)+527,874$	0.7400	0.7069	0.6810	0.6507
			AVG		0.7302	0.6967	0.6816	0.6515
		DTP	MAX-euc	$(\mathbf{n}\times 256)+\{(\mathbf{n}\times 256\times 512)+527,874\}$	0.7480	0.7210	0.6687	0.6521
			AVG-euc		0.7137	0.6815	0.6449	0.6157
			MAX-cos		0.7318	0.7051	0.6648	0.6380
			AVG-cos		0.7106	0.6774	0.6372	0.6088
		SoM-TP	MAX	$(\mathbf{n}\times 256)+\{\{(\mathbf{n}\times 256\times 512)+527,874\}$	**0.7503**	**0.7212**	**0.6969**	**0.6648**
			AVG	$+\{(\mathbf{3n}\times 1)+(256\times 1)\}\}$	**0.7485**	**0.7219**	**0.6909**	**0.6753**
ResNet	1,103,744	GTP	MAX	$(\mathbf{1}\times 256\times 512)+527,874$	0.7162	0.6914	0.6612	0.6193
			AVG		0.7544	0.7244	0.6419	0.6149
		STP	MAX	$(\mathbf{n}\times 256\times 512)+527,874$	0.7464	0.7162	0.6480	0.6114
			AVG		0.7583	0.7323	0.6612	0.6321
		DTP	MAX-euc	$(\mathbf{n}\times 256)+\{(\mathbf{n}\times 256\times 512)+527,874\}$	0.7512	0.7223	0.6420	0.6071
			AVG-euc		0.7258	0.6967	0.6475	0.6227
			MAX-cos		0.7425	0.7152	0.6572	0.6350
			AVG-cos		0.7256	0.6963	0.6299	0.6044
		SoM-TP	MAX	$(\mathbf{n}\times 256)+\{\{(\mathbf{n}\times 256\times 512)+527,874\}$	**0.7690**	**0.7398**	**0.6766**	**0.6542**
			AVG	$+\{(\mathbf{3n}\times 1)+(256\times 1)\}\}$	0.7518	0.7219	**0.6669**	**0.6493**

function with ReLU, optimizer with Adam (Kingma & Ba, 2014). The validation dataset is made from 20% of the training set for a more accurate evaluation. For an imbalanced class, the weighted loss is used to train. The prototype number $\mathbf{n}$ is distributed greedily while accounting for each dataset's unique class number. A more detailed experimental setting is in Appendix A.1.

3.2.2 Learning process analysis

How specific pooling is chosen by each data? SoM-TP identifies data for each mini-batch unit and dynamically selects appropriate pooling. In Figure 3, the learning process is introduced: How the attention finds optimal pooling. SoM-TP reflects every data without converging to one dominant pooling. Dynamic selection can be done during the inference process without the use of a sub-network DPLN. SoM-TP extended ablation study on the individual datasets is in Appendix A.3.

The difference in perspective when compared to individual poolings. A qualitative analysis using LRP is performed to examine how the perspectives of SoM-TP are different from those of other temporal pooling methods. In Figure 2, SoM-TP overcomes the limitation of independent pooling and learns better by taking mixed views of global and local. In detail, SoM-TP outperforms by focusing on hidden important features that others are not concentrating on.

What is the role of DPLN? DPLN is an ensemble network in that the network uses the weighted pooling outputs. SoM-TP employs DPLN's result for the perspective loss. In other words, SoM-TP uses its ensemble result for optimization. Furthermore, the λ ablation study on perspective loss is in Appendix A.2.

3.2.3 Performance analysis

SoM-TP shows robust performance for the various domain TSC datasets. To investigate the performance of SoM-TP in detail, we analyzed both ways; quantitative results and robustness.

We calculated the average performance of the entire dataset (Table 2). We considered accuracy and f1 macro score to deal with the imbalanced class in datasets. SoM-TP outperforms the existing temporal pooling methods both in univariate and multivariate time series datasets. Furthermore, Figure 4 histograms show the superior performance of SoM-TP than DTP which is the SOTA of temporal pooling. SoM-TP has long tail than DTP, which means that highly beats the DTP in accuracy. The most highest beaten in one dataset achieved a 5% increase. We take additional

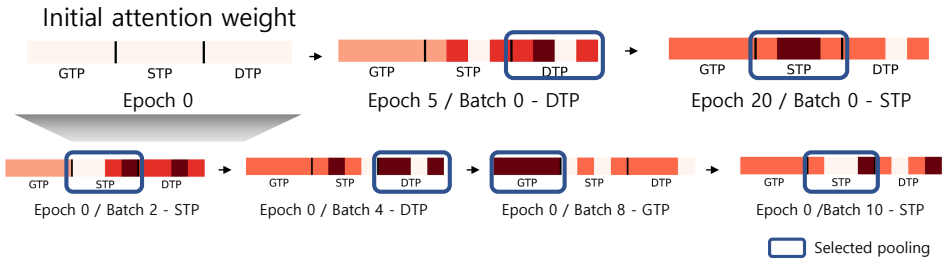
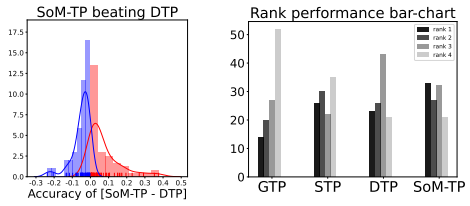
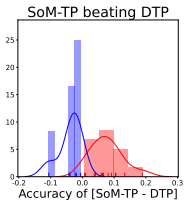


Figure 3: **SoM-TP learning process.** Based on each data sample, attention score provides guidance for selecting pooling with an appropriate view. This heatmap represents the attention score $\mathbf{A}$. The initial attention score before learning is initialized as zero. During the learning process, the attention score is updated with every epoch For example, in epoch 5 of the first row, the attention picks DTP as the best pooling with batch-0. In the second row, the attention picks different pooling by different data samples. Therefore, after several epochs with learning, attention is optimized to select STP as the best pooling on the same batch-0.



(a) ResNet-MAX performance results on UCR.



(b) ResNet-MAX performance results on UEA.

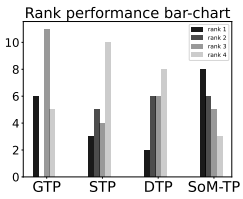


Figure 4: **Detailed performance analysis between temporal poolings and SoM-TP on ResNet.** We choose histograms and bar charts to show the superiority of SoM-TP in a different way. The performance in the graph is based on accuracy. For the histogram, the x-axis means the value obtained by subtracting DTP from SoM-TP and the y-axis is the number of datasets. Here, we can see that there are both cases in which each pooling performs better than the other. However, there are more cases in that SoM-TP outperforms with a big gap in both UCR and UEA repositories. Also, as shown in the bar charts with performance ranks between four temporal poolings, SoM-TP has the most robust performance than any other pooling with rank 1 highest and the following ranks becoming lower, clearly shown in UEA. However, for DTP, which is the SOTA of temporal pooling, it is not clear that overall performance is better than STP. Through these results, we can check the robustness and performance of SoM-TP compared to other temporal pooling.

evaluation metrics as a histogram under area. As shown in (a), (b)-histogram, the area of SoM-TP is bigger than DTP's as much as $+1.1083$ and $+0.1763$ respectively. Through (a), (b)-bar charts, we also check that SoM-TP has the most rank 1 and least rank 4.

However, there are a few cases where fixed-perspective learning is dominant due to its data nature. The high degree of freedom of the SoM-TP mechanism can cause performance degradation for data with simple classification points. This problem appears as an outlier, especially in the feature extraction with FCN.

3.3 Pooling Classification

SoM-TP selects the most appropriate pooling method based on specific data. However, if there is no relationship between data and pooling, selecting suitable pooling according to each characteristic of the data samples cannot be generalized. Without this proposition, SoM-TP is not different from the random selection of poolings. Through an empirical study of pooling classification, we prove the existence of a relationship between distinct data and best-fitted poolings.

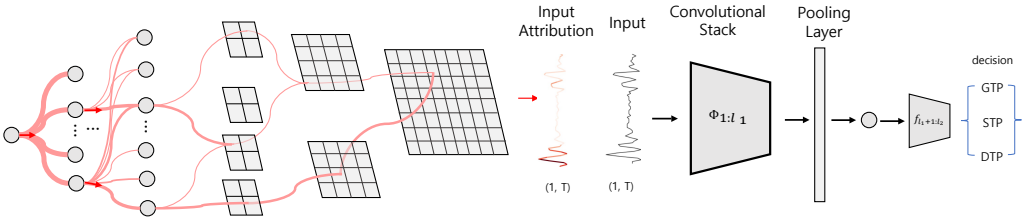


Figure 5: **Pooling classification model architecture.** The input attribution of best temporal pooling by fixed-perspective learning is first calculated with LRP; z^+ rule for $\Phi_{1:l_1}$ and ϵ rule for $f_{l_1+1:l_2}$. Then multivariate input with raw time series and input attribution goes through the model to classify the best pooling methods for each dataset. In this network, the simple global pooling layer is used.

The challenge here is to prove not only the relationship between data and pooling but also its perspective. Therefore, we use relevance score as a multivariate input along with raw time series. The relevance score, which is input attribution, is the LRP result from fixed-perspective learning. By adding LRP values as input, the model can recognize important features that contribute to classification, which is captured by distinct perspective of each polings. This setting is also consistent with the fact that TSC is mainly done by capturing a particular pattern in time series.

Experimental settings. Given a training set of N samples of time series and an LRP value, $\mathbf{T} = \{(\mathbf{X}_1, \mathbf{L}_1, y), ..., (\mathbf{X}_N, \mathbf{L}_N, y)\}$, from class c of three temporal pooling $\{$GTP: 0, STP: 1, DTP: 2$\}$, we aim to learn the CNN classifier. Note that we set the best pooling method as a target class.

Performance analysis. Considering the imbalance of classification target labels, we use F1-score as an evaluation metric. When only time series input is used as univariate classification, the performance is 74% of the f1-score. However, with a relevance score, the performance increases to 78%. This result shows that the data categorization is possible with the relationship proved; each time series has appropriate pooling with distinct views.

4 Conclusion

This paper proposes SoM-TP, switch over multiple pooling that learns diverse perspectives to fully utilize temporal pooling. The pooling perspective that we defined is from the distinct mechanism of either segmentation on temporal order, or dynamic segmentation length by considering temporal relationships. SoM-TP is not one pooling layer, but an architecture, which has a learning framework with perspective loss. With the attention weight vector in SoM-TP, pooling is selected dynamically based on the each data. The extensive experiment with the UCR/UEA repository validated the performance of our new method, and with LRP, we saw SoM-TP view every data sample with a new perspective that is not seen in existing temporal pooling methods.

REFERENCES

Amaia Abanda, Usue Mori, and Jose A Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412, 2019.

Robert J Alcock, Yannis Manolopoulos, et al. Time-series similarity queries employing a feature-based approach. In *7th Hellenic conference on informatics*, pp. 27–29, 1999.

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.

Anthony Bagnall, Jason Lines, Jon Hills, and Aaron Bostrom. Time-series classification with cote: The collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535, 2015. doi: 10.1109/TKDE.2015.2416723.

Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31(3):606–660, 2017.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Gustavo EAPA Batista, Eamonn J Keogh, Oben Moses Tataw, and Vinicius de Souza. Cid: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, 28(3):634–669, 2014.

Mustafa Gokce Baydogan, George Runger, and Eugene Tuv. A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2796–2802, 2013.

Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for deep neural network architectures. In Kuinam J. Kim and Nikolai Joukov (eds.), *Information Science and Applications (ICISA) 2016*, pp. 913–922, Singapore, 2016a. Springer Singapore. ISBN 978-981-10-0557-2.

Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In Alessandro E.P. Villa, Paolo Masulli, and Antonio Javier Pons Rivero (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2016*, pp. 63–71, Cham, 2016b. Springer International Publishing. ISBN 978-3-319-44781-0.

Ka-Hou Chan, Giovanni Pau, and Sio-Kei Im. Chebyshev pooling: An alternative layer for the pooling of cnns-based classifier. In *2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET)*, pp. 106–110. IEEE, 2021.

Sohee Cho, Wonjoon Chang, Ginkyeng Lee, and Jaesik Choi. Interpreting internal activation patterns in deep temporal neural networks by finding prototypes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 158–166, 2021.

Vincent Christlein, Lukas Spranger, Mathias Seuret, Anguelos Nicolaou, Pavel Král, and Andreas Maier. Deep generalized max pooling. In *2019 International conference on document analysis and recognition (ICDAR)*, pp. 1090–1096. IEEE, 2019.

Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

Min Dong, Yongfa Li, Xue Tang, Jingyun Xu, Sheng Bi, and Yi Cai. Variable convolution and pooling convolutional neural network for text sentiment classification. *IEEE Access*, 8:16174–16186, 2020.

Ziheng Duan, Haoyan Xu, Yueyang Wang, Yida Huang, Anni Ren, Zhongbin Xu, Yizhou Sun, and Wei Wang. Multivariate time-series classification with hierarchical variational graph pooling. *Neural Networks*, 154:481–490, 2022.

Ben D Fulcher and Nick S Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014.

Ziteng Gao, Limin Wang, and Gangshan Wu. Lip: Local importance-based pooling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3355–3364, 2019.

Alan H Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. Explaining deep classification of time-series data with learned prototypes. In *CEUR workshop proceedings*, volume 2429, pp. 15. NIH Public Access, 2019.

Pierre Geurts. Pattern extraction for time series classification. In Luc De Raedt and Arno Siebes (eds.), *Principles of Data Mining and Knowledge Discovery*, pp. 115–127, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44794-8.

Alireza Ghods and Diane J Cook. Pip: Pictorial interpretable prototype learning for time series classification. *IEEE Computational Intelligence Magazine*, 17(1):34–45, 2022.

Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review. *arXiv preprint arXiv:2009.07485*, 2020.

Rafael Giusti and Gustavo EAPA Batista. An empirical comparison of dissimilarity measures for time series classification. In *2013 Brazilian Conference on Intelligent Systems*, pp. 82–88. IEEE, 2013.

Fang He, Tao-yang Fu, and Wang-Chien Lee. Rel-cnn: Learning relationship features in time series for classification. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Mai Ibrahim, Ayman Shaawat, and Marwan Torki. Covariance pooling layer for text classification. *Procedia Computer Science*, 189:61–66, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL `http://arxiv.org/abs/1502.03167`.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern recognition*, 44(9):2231–2240, 2011.

Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

Kathan Kashiparekh, Jyoti Narwariya, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. Convtimenet: A pre-trained deep convolutional neural network for time series classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Dongha Lee, Seonghyeon Lee, and Hwanjo Yu. Learnable dynamic temporal pooling for time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8288–8296, 2021.

Jason Lines and Anthony Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2015.

Jason Lines, Luke M Davis, Jon Hills, and Anthony Bagnall. A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 289–297, 2012.

Jason Lines, Sarah Taylor, and Anthony Bagnall. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1041–1046, 2016. doi: 10.1109/ICDM.2016.0133.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Qianli Ma, Wanqing Zhuang, Sen Li, Desen Huang, and Garrison Cottrell. Adversarial dynamic shapelet networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5069–5076, 2020.

Shangbo Mao, Deepu Rajan, and Liang Tien Chia. Deep residual pooling network for texture recognition. *Pattern Recognition*, 112:107817, 2021.

Pierre-François Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):306–318, 2008.

Nikolaos Passalis and Anastasios Tefas. Learning bag-of-features pooling for deep convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5755–5763, 2017.

Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. Explainable artificial intelligence (xai) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950*, 2021.

Patrick Schäfer. The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530, 2015.

Pavel Senin and Sergey Malinchik. Sax-vsm: Interpretable time series classification using sax and vector space model. In *2013 IEEE 13th international conference on data mining*, pp. 1175–1180. IEEE, 2013.

Joan Serrà, Santiago Pascual, and Alexandros Karatzoglou. Towards a universal neural network encoder for time series. In *CCIA*, pp. 120–129, 2018.

Alexandra Stefan, Vassilis Athitsos, and Gautam Das. The move-split-merge metric for time series. *IEEE transactions on Knowledge and Data Engineering*, 25(6):1425–1438, 2012.

Manli Sun, Zhanjie Song, Xiaoheng Jiang, Jing Pan, and Yanwei Pang. Learning pooling for convolutional neural network. *Neurocomputing*, 224:96–104, 2017.

Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey I Webb. Multirocket: Multiple pooling operators and transformations for fast and effective time series classification. *arXiv preprint arXiv:2102.00457*, 2021.

Pattreeya Tanisaro and Gunther Heidemann. Time series classification using time warping invariant echo state networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 831–836. IEEE, 2016.

Peng Wang, Yuanzhouhan Cao, Chunhua Shen, Lingqiao Liu, and Heng Tao Shen. Temporal pyramid pooling-based convolutional neural network for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2613–2622, 2016.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

Travis Williams and Robert Li. Wavelet pooling for convolutional neural networks. In *International Conference on Learning Representations*, 2018.

Jidong Yuan, Ahlame Douzal-Chouakria, Saeed Varasteh Yazdi, and Zhihai Wang. A large margin time series nearest neighbour classification under locally weighted time warps. *Knowledge and Information Systems*, 59(1):117–135, Apr 2019a. ISSN 0219-3116. doi: 10.1007/s10115-018-1184-z. URL https://doi.org/10.1007/s10115-018-1184-z.

Jidong Yuan, Qianhong Lin, Wei Zhang, and Zhihai Wang. Locally slope-based dynamic time warping for time series classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1713–1722, 2019b.

Shuangfei Zhai, Hui Wu, Abhishek Kumar, Yu Cheng, Yongxi Lu, Zhongfei Zhang, and Rogerio Feris. S3pool: Pooling with stochastic spatial sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4970–4978, 2017.

Boxue Zhang, Qi Zhao, Wenquan Feng, and Shuchang Lyu. Alphamex: A smarter global pooling method for convolutional neural networks. *Neurocomputing*, 321:36–48, 2018.

Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6845–6852, 2020.

Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.

Jiaping Zhao and Laurent Itti. shapedtw: Shape dynamic time warping. *Pattern Recognition*, 74: 171–184, 2018.

Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. *Frontiers of Computer Science*, 10(1):96–112, 2016.

A EXTENDED ANALYSIS ON SOM-TP

A.1 DETAILED EXPERIMENTAL SETTING

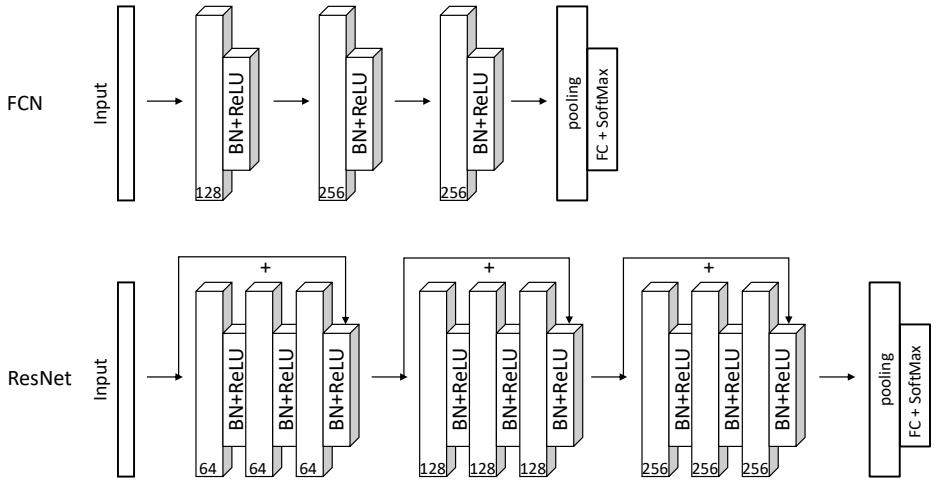
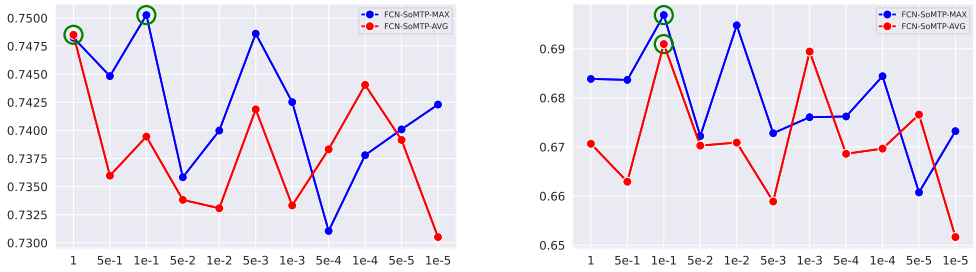


Figure 6: **Convolutional stack used in the experiment.** We use FCN and ResNet which are specially designed for TSC. In here, we can see the embedding dimensions of each convolutional layer before the pooling and FC layers for the classification decision.

A.2 PERSPECTIVE LOSS: λ ABLATION STUDY



(a) FCN SoM-TP λ study in UCR

(b) FCN SoM-TP λ study in UEA

Figure 7: **The λ ablation study for SoM-TP.** λ is the decay value of the perspective loss, which is one of the e most important hyper-parameter in SoM-TP. Therefore, in the range of [1, 1e-5] with 11 intervals, the ablation study is shown in the figure. The color of the line represents each pooling type between MAX and AVG. And the optimal λ with the highest performance is circled with green. We can see that scale of λ affects SoM-TP performance directly.

A.3 EXTENDED SOM-TP ABLATION STUDY WITH AN INDIVIDUAL DATASET

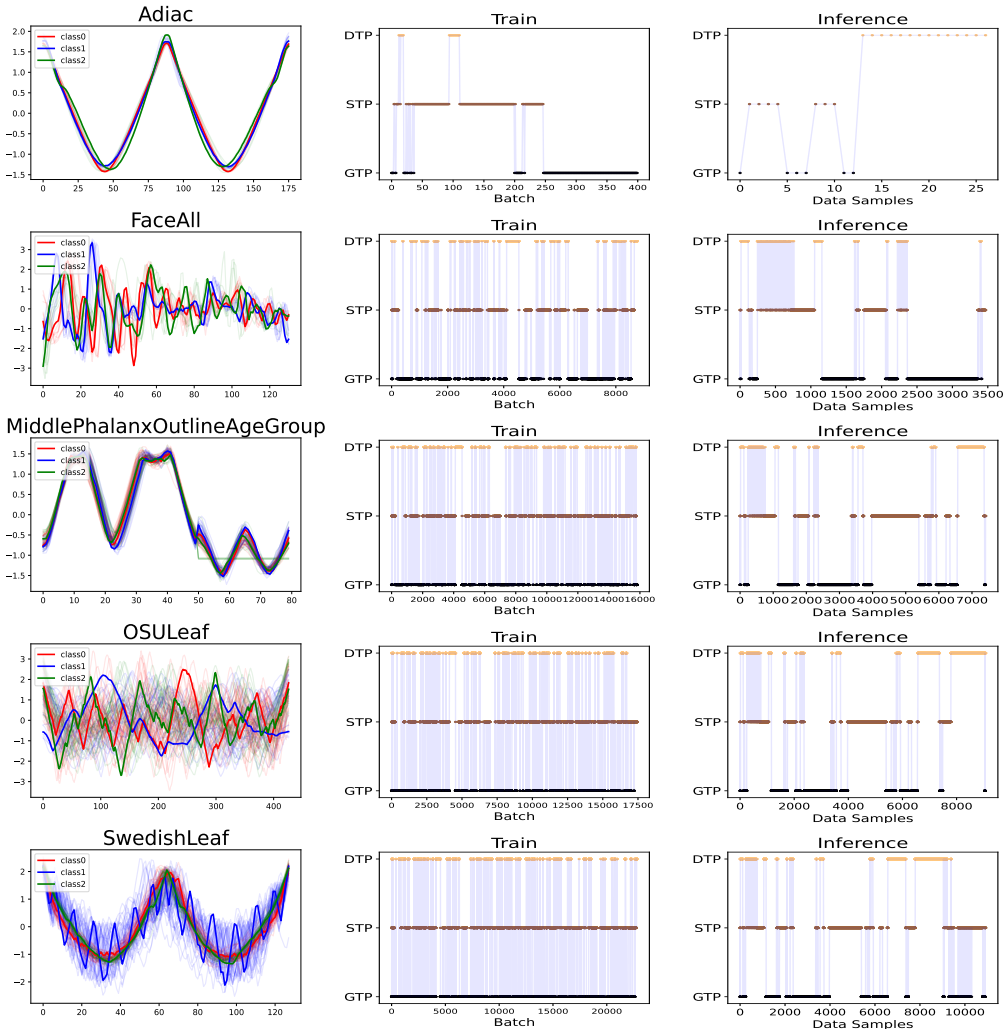


Figure 8: **SoM-TP on different time lengths and data sizes.** With different datasets which contain different data sizes and time lengths, the SoM-TP works robustly and dynamically selects appropriate pooling both at training and inference procedure. With the first column, the x-axis, we can see the different time lengths of each dataset. Also, with the same 20 epoch and 8 batch size, the batch number is different, which means the difference in data size of each individual dataset. However, SoM-TP learns diverse perspectives from the training procedure and peaks different but appropriate pooling by specific data samples.

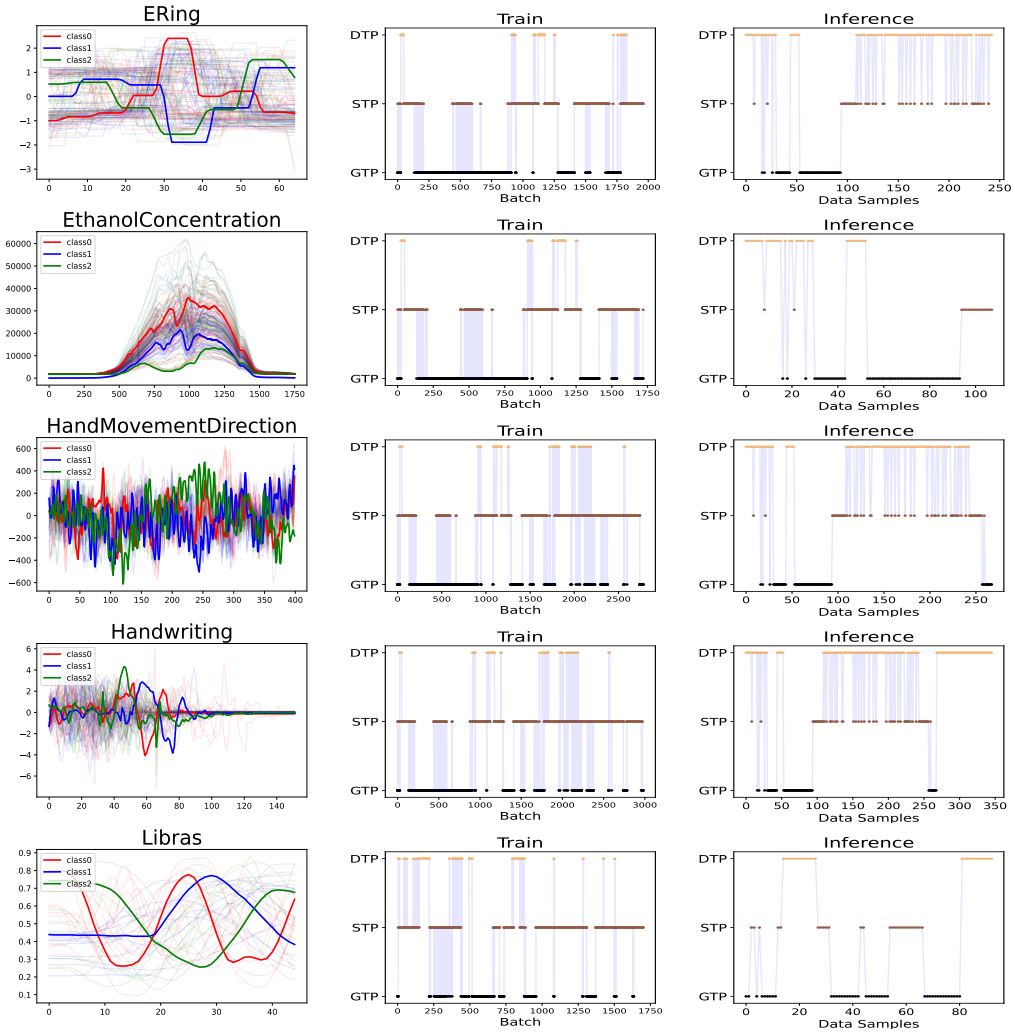


Figure 9: **SoM-TP on different time lengths, data sizes, and dimensions.** With different datasets which contain different dimensions in the UEA repository, the SoM-TP works robustly and dynamically select appropriate pooling both at training and inference procedure. In here, the multivariate datasets with different dimensions and data sizes are shown and the detailed selection of pooling by each data with SoM-TP robustly performs with proper selection.

B DTP ALGORITHM

Algorithm 1 DTP layer optimization

Function `DTP` ($\mathbf{P}$, $\mathbf{H}$) **:**

$\delta(p_l, h_t) = 1 - \frac{p_l \cdot h_t}{||p_l||_2 ||h_t||_2}$

Function `forward` ($\mathbf{P}$, $\mathbf{H}$) **:**

▷ fill the alignment cost matrix $R \in \mathbb{R}^{L \times T}$

$R_{0,0} = 0, R_{:,0} = R_{0,:} = \infty$

for $l = 1, ..., L$ **do**

  for $t = 1, ..., T$ **do**

    $R_{l,t} = \delta(p_l, h_t) + min_\gamma\{R_{l-1,t-1}, R_{l,t-1}\}$

return $DTW_\gamma(\mathbf{P}, \mathbf{H}) = R_{L,T}$

Function `backward` ($\mathbf{P}$, $\mathbf{H}$) **:**

▷ fill the soft alignment matrix $E \in \mathbb{R}^{L \times T}$

$E_{l,t} = \partial R_{L,T} / \partial R_{l,t}$

$E{:}, T+1 = E_{L+1,:} = 0$

$R_{:,T+1} = R_{L+1,:} = -\infty$

for $l = L, ..., 1$ **do**

  for $t = T, ..., 1$ **do**

    $a = exp\frac{1}{\gamma}(R_{l,t+1} - R_{l,t} - \delta(p_l, h_{t+1}))$

    $b = exp\frac{1}{\gamma}(R_{l+1,t+1} - R_{l,t} - \delta(p_{l+1}, h_{t+1}))$

    $E_{l,t} = a \cdot E_{l,t+1} + b \cdot E_{l+1,t+1}$

return $\nabla_P DTW_\gamma(\mathbf{P}, \mathbf{H}) = (\frac{\partial \Delta(\mathbf{P}, \mathbf{H})}{\partial \mathbf{P}})^T E$

Function `optimization` $((\mathbf{X}, y), \mathbf{P}, \Phi)$ **:**

▷$\mathcal{W}$: network Φ parameter

▷$w^{(c)} \in \mathbb{R}^K$ of class weight vector

▷$\mathbf{W}^c = [w_1^{(c)}, ..., w_L^{(c)}] \in \mathbb{R}^{k \times L}$ of class weight matrix

▷$P(y = c|\mathbf{X}) = \frac{exp(\sum_{l=1}^{L} \bar{h_l} \cdot w_l^{(c)})}{\sum_{c'=1}^{L} exp(\sum_{l=1}^{L} \bar{h_l} \cdot w_l^{c'})}$ of posterior

$\mathcal{L}_{proto}(\mathbf{P}) = \frac{1}{N} \sum_N DTW_\gamma(P, \Phi(\mathbf{X}^n; \mathcal{W}))$

$\mathcal{L}_{class}(\mathcal{W}, \{\mathbf{W}^{(c)}\}) = -\frac{1}{N} \sum_{n=1}^{N} logP(y = y^n|\mathbf{X}^n)$

procedure OPTIMIZE($\mathbf{P}, \mathbf{W}, \mathcal{W}$)

  $\mathbf{P} \leftarrow \mathbf{P} - \eta \cdot \partial \mathcal{L}_{proto} / \partial \mathbf{P}$

  $\mathcal{W} \leftarrow \mathcal{W} - \eta \cdot \mathcal{L}_{class} / \partial \mathcal{W}$

  $\mathbf{W}^{(c)} \leftarrow \mathbf{W}^{(c)} - \eta \cdot \partial \mathcal{L}_{class} / \partial \mathbf{W}^{(c)}$

C EXTENDED RELATED WORK

Conventional approach for TSC. In early TSC, there were various approaches to classifying time series data based on the similarity of distance or feature pattern (Geurts, 2001; Alcock et al., 1999; Fulcher & Jones, 2014; Abanda et al., 2019; Bagnall et al., 2017; Giusti & Batista, 2013). The distance-based metric developed variously, and especially DTW, which is also used in DTP, is applied dynamically to TSC (Batista et al., 2014; Lines & Bagnall, 2015; Cuturi & Blondel, 2017; Jeong et al., 2011; Marteau, 2008; Stefan et al., 2012; Yuan et al., 2019b; Zhao & Itti, 2018). There are also various algorithms that serve as a noble foundation (Bagnall et al., 2015; Lines et al., 2016; Baydogan et al., 2013; Lines et al., 2012; Ma et al., 2020; Schäfer, 2015; Yuan et al., 2019a). However, these baselines are defeated by DNN architecture (Lee et al., 2021; Ismail Fawaz et al., 2019).

Deep Neural Network for TSC. Especially, CNN has diverse architecture for TSC, which tried to consider time series characteristics (Cui et al., 2016; Karim et al., 2017; Ismail Fawaz et al., 2020; Zheng et al., 2016; Kashiparekh et al., 2019; He et al., 2022; Duan et al., 2022). Also, there were novel trials to interpret time series data and its classification (Senin & Malinchik, 2013), specifically, prototype and attention-based approaches were usually driven (Zhang et al., 2020; Gee et al., 2019; Ghods & Cook, 2022; Cho et al., 2021). However, for TSC with CNN, there were few studies that used LRP for explaining and interpreting the model and its time series input attribution (Rojat et al., 2021).

Pooling for TSC. In other fields, especially computer vision and natural language processing, there were many approaches to novel pooling methods (Mao et al., 2021; Sun et al., 2017; Ibrahim et al., 2021; Christlein et al., 2019; Dong et al., 2020; Zhang et al., 2018; Williams & Li, 2018; Chan et al., 2021; Zhai et al., 2017; Passalis & Tefas, 2017; Gao et al., 2019). There are, however, few approaches for the time series domain (Dempster et al., 2020; Tan et al., 2021; Wang et al., 2016).