Towards Diverse Perspective Learning with Selection over Multiple Temporal Poolings

Jihyeon Seong*1, Jungmin Kim*1, Jaesik Choi1,2

1 Korea Advanced Institute of Science and Technology (KAIST), South Korea
2 INEEJI, South Korea
{jihyeon.seong, aldirl7, jaesik.choi}@kaist.ac.kr

Abstract

In Time Series Classification (TSC), temporal pooling methods that consider sequential information have been proposed. However, we found that each temporal pooling has a distinct mechanism, and can perform better or worse depending on time series data. We term this fixed pooling mechanism a single perspective of temporal poolings. In this paper, we propose a novel temporal pooling method with diverse perspective learning: Selection over Multiple Temporal Poolings (SoM-TP). SoM-TP dynamically selects the optimal temporal pooling among multiple methods for each data by attention. The dynamic pooling selection is motivated by the ensemble concept of Multiple Choice Learning (MCL), which selects the best among multiple outputs. The pooling selection by SoM-TP’s attention enables a non-iterative pooling within a single classifier. Additionally, we define a perspective loss and Diverse Perspective Learning Network (DPLN). The loss works as a regularizer to reflect all the pooling perspectives from DPLN. Our perspective analysis using Layer-wise Relevance Propagation (LRP) reveals the limitation of a single perspective and ultimately demonstrates diverse perspective learning of SoM-TP. We also show that SoM-TP outperforms CNN models based on other temporal poolings and state-of-the-art models in TSC with extensive UCR/UEA repositories.

Introduction

Time Series Classification (TSC) is one of the most valuable tasks in data mining, and Convolutional Neural Network (CNN) with global pooling shows revolutionary success on TSC (Längkvist, Karlsson, and Loutfi 2014; Ismail Fawaz et al. 2019). However, global pooling in TSC poses a significant challenge, as it disregards the fundamental characteristic of time series data, which is the temporal information, by compressing it into a single scalar value (Lecun et al. 1998; Yu et al. 2014). To tackle this issue, temporal pooling methods were introduced, which preserve the temporal nature of the time series at the pooling level (Lee, Lee, and Yu 2021).

Temporal pooling involves employing operations such as ‘maximum’ (MAX) and ‘average’ (AVG), categorized by segmentation types: ‘no segment,’ ‘uniform,’ and ‘dynamic.’ These segmentation types correspond respectively to Global-Temporal-Pooling (GTP), Static-Temporal-Pooling (STP), and Dynamic-Temporal-Pooling (DTP) (Lee, Lee, and Yu 2021). We refer to each distinct pooling mechanism as a perspective based on the segmentation types. However, we discovered that the most effective temporal pooling varies depending on the characteristics of the time series data, and there is no universally dominant pooling method for all datasets (Esling and Agon 2012). This underlines the necessity for a learnable pooling approach adaptable to each data sample’s characteristics.

In this paper, we propose Selection over Multiple Temporal Poolings (SoM-TP). SoM-TP is a learnable ensemble pooling method that dynamically selects heterogeneous temporal poolings through an attention mechanism ( Vaswani et al. 2017). Aligned with our observation that a more suitable pooling exists for each data sample, a simple ensemble weakens the specified representation power (Lee et al. 2017). Therefore, SoM-TP applies advanced ensemble learning, motivated by Multiple Choice Learning (MCL), that selects the best among the multiple pooling outputs (Guzmán-rivera, Batra, and Kohli 2012).

MCL is a selection ensemble that generates $M$ predictions from multiple instances, computes the oracle loss for the most accurate prediction, and optimizes only the best classifier. Capitalizing on the advantage of deep networks having access to intermediate features, SoM-TP ensembles diverse pooling features in a single classifier. To achieve non-iterative optimization, SoM-TP dynamically selects the most suitable pooling method for each data sample through attention, which is optimized by Diverse Perspective Learning Network (DPLN) and perspective loss. DPLN is a sub-network that utilizes all pooling outputs, and the perspective loss reflects DPLN’s result to make a regularization effect. Finally, the CNN model based on SoM-TP forms fine representations through diverse pooling selection, allowing it to capture both the ‘global’ and ‘local’ features of the dataset.

Recognizing the crucial role of pooling in selecting the most representative values from encoded features in CNNs, we have chosen CNNs as the suitable model for our study. We apply our new selection ensemble pooling to Fully Convolutional Networks (FCNs) and Residual Networks (ResNet), which show competitive performances in TSC as a CNN-based model (Wang, Yan, and Oates 2017; Ismail Fawaz et al. 2019). SoM-TP outperforms the exist-
ing temporal pooling methods and state-of-the-art models of TSC both in univariate and multivariate time series datasets from massive UCR/UEA repositories. We also provide a detailed analysis of the diverse perspective learning result by Layer-wise Relevance Propagation (LRP) (Bach et al. 2015) and the dynamic selection process of SoM-TP. To the best of our knowledge, this is the first novel approach to pooling-level ensemble study in TSC.

Therefore, our contributions are as follows:

• We investigate data dependency arising from distinct perspectives of existing temporal poolings.
• We propose SoM-TP, a new temporal pooling method that fully utilizes the diverse temporal pooling mechanisms through an MCL-inspired selection ensemble.
• We employ an attention mechanism to enable a non-iterative ensemble in a single classifier.
• We define DPLN and perspective loss as a regularizer to promote diverse pooling selection.

Background

Different Perspectives between Temporal Poolings

Convolutional Neural Network in TSC  In TSC, CNN outperforms conventional methods, such as nearest-neighbor classifiers (Yuan et al. 2019) or COTE (Bagnall et al. 2015; Lines, Taylor, and Bagnall 2016), by capturing local patterns of time series (Ismail Fawaz et al. 2019).

The TSC problem is generally formulated as follows: a time series data $T = \{(X_1, y_1), \ldots, (X_t, y_t)\}$, where $X \in \mathbb{R}^{d \times t}$ of length $t$ with $d$ variables and $y \in \{1, \ldots, C\}$ from $C$ classes. Then the convolution stack $\Phi$ of out channel dimension $k$ encodes features as hidden representations with temporal position information $H = \{h_0, \ldots, h_t\} \in \mathbb{R}^{k \times t}$ (Lee, Lee, and Yu 2021; Wang, Yan, and Oates 2017).

$$H = \Phi(T)$$  \hspace{1cm} (1)

After convolutional layers, global pooling plays a key role with two primary purposes: 1) reducing the number of parameters for computational efficiency and preventing overfitting, and 2) learning position invariance. For this purpose, pooling combines the high-dimensional feature outputs into low-dimensional representations (Ghobaldinezhad and Khosravi 2020). However, global pooling presents an issue of losing temporal information, which has led to the development of temporal pooling methods (Lee, Lee, and Yu 2021). We investigate different mechanisms of temporal poolings, which we refer to as perspective.

Global Temporal Pooling  GTP pools only one representation $p_g = [p_1] \in \mathbb{R}^{k \times 1}$ in the entire time range. GTP ignores temporal information by aggregating the $H$ to $p_g = h$: the global view.

$$p_g = pool_g(H)$$  \hspace{1cm} (2)

GTP effectively captures globally dominant features, such as trends or the highest peak, but has difficulty capturing multiple points dispersed on a time axis. To solve this constraint, temporal poolings based on sequential segmentation have been proposed: STP and DTP (Lee, Lee, and Yu 2021). Both have multiple local segments with the given number $n \in \mathbb{Z}^+$: the local view.

Static Temporal Pooling  STP divides the time axis equally into $n$ segments with a length $\ell = \frac{t}{n}$, where $H = \{h_{0, \ell}, h_{\ell, 2\ell}, \ldots, h_{(n-1)\ell, n\ell}\}$ and $p_s = [p_1, \ldots, p_n] \in \mathbb{R}^{k \times n}$. Note that $h_{\ell}$ retains temporal information, but there is no consideration of the temporal relationship between time series in the segmentation process: the uniform local view.

$$p_s = pool_s(H)$$  \hspace{1cm} (3)

STP functions well on a recursive pattern, such as a stationary process. However, forced uniform segmentation can divide important consecutive patterns or create unimportant segmentations. This inefficiency causes representation power to be distributed to non-informative regions.

Dynamic Temporal Pooling  DTP is a learnable pooling layer optimized by soft-DTW (Cuturi and Blondel 2017) for dynamic segmentation considering the temporal relationship. By using the soft-DTW layer, $H$ is segmented in diverse time lengths $\ell = [\ell_1, \ell_2, \ldots, \ell_n]$, where $t = \sum \ell$. Finally, the optimal pooled vectors $p_d = [p_{\ell_1}, \ldots, p_{\ell_n}] \in \mathbb{R}^{k \times n}$ are extracted from each segment of $H\ell$, where $H\ell = \{h_{\ell_1}, h_{\ell_2}, \ldots, h_{\ell_n}\}$: the dynamic local view.

$$p_d = pool_d(H\ell)$$  \hspace{1cm} (4)

DTP has the highest complexity in finding different optimal segmentation lengths, enabling the pooling to fully represent segmentation power. However, since DTP is based on temporally aligned similarity of hidden features with a constraint that a single time point should not be aligned with multiple consecutive segments, the segmentation can easily divide informative change points that need to be preserved in time series patterns (Appendix. DTP Algorithm).

Limitation of Single Perspective  Traditional temporal pooling methods only focus on a single perspective when dealing with hidden features $H$. A global perspective cannot effectively capture multiple classification points, while a local perspective struggles to emphasize a dominant classification point. Consequently, datasets that require the simultaneous capture of dominant and hidden local features from diverse viewpoints inevitably exhibit lower performance when using a single perspective. Motivated by these limitations, we propose a novel pooling approach that fully leverages diverse perspectives.
Multiple Choice Learning for Deep Temporal Pooling

The traditional ML-based ensemble method focuses on aggregating multiple outputs. However, the aggregation of the simple ensemble makes outputs smoother, due to the generalization effect (Lee et al. 2017). To solve this limitation, MCL has been proposed as an advanced ensemble method that selects the best among multiple outputs using oracle loss (Guzmán-rivera, Batra, and Kohli 2012). More formally, MCL generates $M$ solutions $Y_i = (\hat{y}^1_i, ..., \hat{y}^M_i)$, and learns a mapping $g : \mathcal{X} \rightarrow \mathcal{Y}^M$ that minimizes oracle loss $\min_m \ell(y_i, \hat{y}^m_i)$. The ensemble mapping function $g$ consists of multiple predictors, $g(x) = \{f_1(x), f_2(x), ..., f_M(x)\}$ (Lee et al. 2016).

The effects of diverse solution sets in MCL can be summarized as addressing situations of ‘Ambiguous evidence’ and ‘Bias towards the mode’. ‘Ambiguous evidence’ refers to situations with insufficient information to make a definitive prediction. In such cases, presenting a small set of reasonable possibilities can alleviate the over-confidence problem of deep learning (Nguyen, Yosinski, and Clune 2015), rather than striving for a single accurate answer. The other situation is ‘Bias towards the mode’, indicating the model’s tendency to learn a mode-seeking behavior to reduce the expected loss across the entire dataset. When only a single prediction exists, the model eventually learns to minimize the average error. In contrast, MCL generates multiple predictions, allowing some classifiers to cover the low-density regions of the solution space without sacrificing performance on the high-density regions (Lee et al. 2016).

MCL faces computational challenges in deep networks due to the iterative optimization process of the oracle loss, with a complexity of $O(N^2)$. Although sMCL has partially alleviated this issue through stochastic gradient descent, the method still requires identifying the best output among multiple possibilities (Lee et al. 2016). CMCL, which is another approach to address MCL’s over-confidence problem, cannot be applied at the feature level due to optimization at the output level (Lee et al. 2017). In summary, the integration of MCL into the pooling level is not feasible due to the structural constraints imposed by the oracle loss design. To overcome this challenge, we establish a model structure to incorporate the concept of MCL into a pooling-level ensemble.

Selection over Multiple Temporal Pooling

SoM-TP Architecture and Selection Ensemble

Diverse Perspective Learning (DPL) is achieved by dynamically selecting heterogeneous multiple temporal poolings in a single classifier. The overview architecture, as illustrated in Figure 2, consists of four parts: 1) a common feature extractor with CNN $\Phi$; 2) a pooling block with multiple temporal pooling layers; 3) an attention block with an attention weight vector $A_0$, and a convolutional layer $\phi_0$, as well as 4) Fully Connected layers (FC); a classification network (CLS) $f_{CLS}$ and a $DPLN$ $f_{DPLN}$. Through these modules, SoM-TP can cover the high probability prediction space, which is aligned with MCL where multiple classifiers are trained to distinguish specific distributions (Lee et al. 2016).
DPL Attention  SoM-TP ensembles multiple temporal pooling within a single classifier. The advantage of using a single classifier lies in the absence of the need to compare prediction outputs, making it non-iterative and computationally efficient, in contrast to MCL. Through the attention mechanism (Vaswani et al. 2017), we achieve a ‘comparison-free’ ensemble by dynamically selecting the most suitable temporal pooling for each batch.

DPL attention is an extended attention mechanism that simultaneously considers two factors: the overall dataset and each data sample. In Figure 2, the attention weight vector $A_0$ is learned in the direction of adding weight to specific pooling, which has a minimum loss for every batch. Consequently, $A_0$ has a weight that reflects the entire dataset.

Next, a convolutional layer $\phi_0$ is used to reflect a more specific data level. As shown in Algorithm 1, the weighted pooling vector $M$ which is the element-wise multiplication between $A_0$ and $P$ is given as an input to $\phi_0$. By encoding $M$ through $\phi_0$, $A$ can reflect more of each batch characteristic, not just following the dominant pooling in terms of the entire dataset. As a result, this optimized pooling selection by attention can solve the ‘Bias towards the mode’ problem by assigning the most suitable pooling for each data batch, which is aligned with learning the multiple experts in MCL.

Diverse Perspective Learning Network and Perspective Loss

To optimize DPL attention, we introduce DPLN and perspective loss. DPLN is a sub-network that utilizes the weighted aggregation ensemble $E$. The main role of DPLN is regularization through perspective loss. In contrast, the CLS network, which predicts the main decision, does not directly utilize attention $A$. Instead, it employs a chosen pooling feature output (denoted as $p_0$ in Figure 2), determined by the index with the biggest score in $A$.

Perspective Loss  Perspective loss serves as a cost function to maximize the utilization of the sub-network, DPLN, through network tying between the two FC networks. Ultimately, it aims to prevent the CLS network from converging to one dominant pooling and to maintain the benefits of ensemble learning continuously.

To achieve its purpose, perspective loss is designed as the sum of DPLN cross-entropy loss and the Kullback-Leibler (KL) divergence between $y_{CLS}$ and $y_{DPL}$. Specifically, KL divergence works similarly to CMCL’s KL term, which regulates one model to be over-confident through a uniform distribution, while the KL term of perspective loss regulates based on DPLN (Lee et al. 2017).

$$KL(y_{CLS}; y_{DPL}) = y_{DPL} \cdot log \frac{y_{DPL}}{y_{CLS}},$$
$$L_{DPLN}(\{W_\Phi\}, \{W^{(dpln)}\}) = - \frac{1}{t} \sum_{n=1}^{t} \log P(y = y_n | X_n),$$
$$L_{\text{perspective}} = KL(y_{CLS}; y_{DPL}) + L_{DPLN},$$

(5)

Algorithm 1: SoM-TP selecting algorithm

Function Attention Block($H$):

- select proper temporal pooling by attention $A \in R^{1 \times 3n}$
- attention weight $A_0 \in R^{1 \times 3n}$ is initialized as zero
- GTP, STP, DTP: $pool_g, pool_s, pool_d$
- convolutional encoding layer: $\phi_0$

Function Pooling Block($H$):

- convolutional hidden feature:
  $H \in R^{k \times t}$
- static segmented hidden feature:
  $\bar{H} = \{h_{0,\ell}, h_{t,\ell}, \ldots, h_{(n-1),\ell}\}$, $\ell = \frac{t}{n}$
- dynamic segmented hidden feature:
  $\bar{H}_d = \{h_{0,\ell}, h_{t,\ell}, \ldots, h_{\ell_n}\}$, $\ell = \frac{t}{n}$, where $t = \sum \ell$
- pooling outputs:
  $p_g, p_s, p_d = pool_g(H), pool_s(\bar{H}), \text{pool}_d(\bar{H}_d)$
  return $p_g, p_s, p_d$

$\bar{P} = [p_g, p_s, p_d]$
$M = A_0 \odot \bar{P}$
$A = \phi_0(M)$, where $x \in A$

$idx = \{\text{argmax}(y), \text{where } y_i = \frac{\exp(x_i)}{\sum_{j} \exp(x_j)}\}$
$\text{argmax}(y)$, where $y_j = \frac{\exp(x_j)}{\sum_{j} \exp(x_j)}$

return $p = \bar{P}(idx), E = A \odot \bar{P}$

where input time series $\{(X_1, y_1), \ldots, (X_t, y_t)\}$, $\Phi$ with learnable parameter $W_\Phi$ of CNN, $y_{CLS} \in R^{1 \times c}$ from the ‘CLS network’ $W^{(cls)}$, and $y_{DPL} \in R^{1 \times c}$ from the DPLN $W^{(dpln)}$. Then, we set first $f_{DPLN}$ weight matrix $W_0^{(dpln)} = [w_1^{(p_g)}, \ldots, w_n^{(p_g)}], \ldots, w_n^{(p_d)} \in R^{k \times 3n}$, where $w^{(p)} \in R^k$ is weight matrix of each latent dimension $k$ of pooling $p_i$, whereas $w_0 \in R^{1 \times c}$ is the first $f_{CLS}$ weight matrix (Lee, Lee, and Yu 2021). Note that the results of GTP are repeated $n$ times to give the same proportion for each pooling by attention weight.

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

$$L_{CLS}(\{W_\Phi\}, \{W^{(cls)}\}) = - \frac{1}{t} \sum_{n=1}^{t} \log P(y = y_n | X_n),$$
$$L_{cost}(\{W_\Phi\}, \{W\}) = L_{CLS} + \lambda \cdot L_{\text{perspective}},$$

(6)

where $\{W^{(cls)}, W^{(dpln)}\}, A_0, \phi_0 \in W$ are learnable parameters. Prioritizing classification accuracy, the loss $L_{CLS}$ is computed, and $L_{\text{perspective}}$ is added with $\lambda$ decay.

As a result, SoM-TP can address the ‘Ambiguous evidence’ problem through DPLN and perspective loss. In a pooling ensemble, the ‘Ambiguous evidence’ can be conceived as a scenario where a single pooling is not dominant. Even though SoM-TP selects only one pooling, $y_{DPL}$ in the perspective loss enables the model to consider the importance of other poolings.
Table 1: SoM-TP Comparison with Single Perspective Temporal Poolings. The table presents the effectiveness of the selection ensemble of SoM-TP compared to traditional temporal poolings. The best performances where SoM-TP outperforms others are bolded and the best performances of other temporal poolings are underlined.

<table>
<thead>
<tr>
<th>CNN</th>
<th>POOL (type)</th>
<th>UCR (uni-variate)</th>
<th>UEA (multi-variate)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAX</td>
<td>ACC</td>
<td>F1 macro</td>
</tr>
<tr>
<td>FCN</td>
<td>GTP</td>
<td>0.6992</td>
<td>0.6666</td>
</tr>
<tr>
<td></td>
<td>STP</td>
<td>0.7462</td>
<td>0.7133</td>
</tr>
<tr>
<td></td>
<td>DTP</td>
<td>0.7406</td>
<td>0.7123</td>
</tr>
<tr>
<td></td>
<td>euc</td>
<td>0.7335</td>
<td>0.7062</td>
</tr>
<tr>
<td></td>
<td>cos</td>
<td>0.7889</td>
<td>0.6795</td>
</tr>
<tr>
<td>ResNet</td>
<td>SoM-TP</td>
<td>0.7556</td>
<td>0.7241</td>
</tr>
<tr>
<td></td>
<td>STP</td>
<td>0.7420</td>
<td>0.7169</td>
</tr>
<tr>
<td></td>
<td>DTP</td>
<td>0.7456</td>
<td>0.7197</td>
</tr>
<tr>
<td></td>
<td>euc</td>
<td>0.7452</td>
<td>0.7198</td>
</tr>
<tr>
<td></td>
<td>cos</td>
<td>0.7773</td>
<td>0.7489</td>
</tr>
</tbody>
</table>

*This table is for pooling type MAX. Please refer to Table 7 in Appendix for pooling type AVG.

Optimization

For attention weight $A_0$, SoM-TP proceeds with additional optimization: a dot product similarity to regulate $A_0$. The similarity term is defined as,

$$L_{attn} = -y_{CLS} \cdot y_{DPL},$$  \hspace{1cm} (7)

Due to the KL-divergence cost function in perspective loss, the CLS network and DPLN can be overly similar during the optimization process. As the additional regulation for output over-similarity, $L_{attn}$ plays the opposite regulation to the perspective loss. Note that the dot-product similarity considers both the magnitude and direction of two output vectors.

Finally, the overall optimization process is as follows,

$$A_0 \leftarrow A_0 - \eta \cdot \partial L_{attn}/\partial A_0,$$
$$W_\Phi \leftarrow W_\Phi - \eta \cdot \partial L_{cost}/\partial W_\Phi,$$
$$W \leftarrow W - \eta \cdot \partial L_{cost}/\partial W.$$  \hspace{1cm} (8)

As a result, even with a non-iterative optimization process, SoM-TP learns various perspectives through DPL attention, DPLN, and perspective loss. Consequently, $\Phi$ reflects $f_{CLS}$ and $f_{DPLN}$ relatively while minimizing the similarity between each network output.

Experiments

Experimental Settings

For the extensive evaluation, 112 univariate and 22 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. To ensure the validity of our experiments, we exclude a few datasets from the UCR/UEA repositories due to the irregular data lengths. While zero padding could resolve this, it might cause bias in some time series models.

All temporal pooling methods have the same CNN architecture. FCN and ResNet are specifically designed as a feature extractor (Wang, Yan, and Oates 2017), and temporal poolings are constructed with the same settings: normalization with BatchNorm (Ioffe and Szegedy 2015), activation function with ReLU, and optimizer with Adam (Kingma and Ba 2015). The validation set is made from $20\%$ of the training set for a more accurate evaluation. In the case of imbalanced classes, a weighted loss is employed. The prototype number $n$ is searched in a greedy way, taking into consideration the unique class count of each dataset. Specifically, we observe that selecting 4-10 segments based on the class count in each dataset enhances performance. Consequently, we use an equal number of segments in each dataset for segment-based poolings (Appendix, Table 5).

Baselines

We conduct two experiments to evaluate the performance of SoM-TP. First, we compare it with traditional temporal poolings, GTP, STP, and DTP, to demonstrate the effectiveness of selection-ensemble in temporal poolings (Lee, Lee, and Yu 2021). Second, we compare SoM-TP with other state-of-the-art models that utilize advanced methods, including scale-invariant methods (ROCKET (Dempster, Petitjean, and Webb 2020), InceptionTime (Ismail Fawaz et al. 2020), OS-CNN (Tang et al. 2021), and DSN (Xiao et al. 2022)), sequential models (MLSTM-FCN (Karim et al. 2019), and TCN (Bai, Kolter, and Koltun 2018)), and Transformer-based models (Vanilla-Transformer (Vaswani et al. 2017), TST (Zerveas et al. 2021), and ConvTran (Foumani et al. 2024)). Models leveraging temporal information use attention or RNNs to emphasize long-term dependencies. On the other hand, scale-invariant learning models employ a CNN-based architecture with various kernel sizes to find the optimum through global average pooling.

Experimental Evaluation

Performance Analysis  As shown in Table 1, SoM-TP shows superior performance for overall TSC datasets when compared to conventional temporal poolings. We calculate the average performance of the entire repository. We consider not only accuracy but also the F1 macro score, ROC-AUC, and PR-AUC to consider the imbalanced class. Quantitatively, SoM-TP outperforms the existing temporal pooling methods both in univariate and multivariate time series datasets. Through these results, we can confirm that the dynamic selection ensemble of SoM-TP boosts the performance of the CNN model.
Figure 3: Dynamic Pooling Selection in SoM-TP. This figure represents the graph of dynamic selection in the FCN SoM-TP MAX on the UCR repository: ArrowHead, Chinatown, and ACSF1.

Table 2: SoM-TP Comparison with Advanced TSC Methods. This table compares the performance of SoM-TP with advanced TSC models that leverage temporal information and those that exploit scale-invariant properties, respectively. The best performances, where SoM-TP beat others, are bolded, and the best performances among other models are underlined.

Table 3: SoM-TP Module Ablation Study.

Additionally, Table 2 compares SoM-TP with other state-of-the-art TSC models from two different approaches. In Table 2-1, ResNet SoM-TP MAX significantly outperforms other sequential models in terms of comparing models leveraging temporal information. As SoM-TP clearly outperforms all other models in accuracy metric, we demonstrate the robustness of performance by providing the number of datasets where SoM-TP achieves higher accuracy. Considering the lowest average rank of SoM-TP, we can conclude that dynamic pooling selection leverages the model to keep important temporal information in a more optimal way than other methods in the massive UCR/UEA repository.

Next, Table 2-2 highlights SoM-TP’s comparable performance alongside scale-invariant methods, even with SoM-TP’s significant computational efficiency. Since SoM-TP and scale-invariant methods have different learning approaches, it is suitable to consider various metrics. Regarding the time complexity of models, scale-invariant methods consider various receptive fields of a CNN, which results in longer training times. In contrast, SoM-TP achieves comparable performance with only one-third of the time.

Finally, in Table 3, we present the results of an ablation study on the modules of SoM-TP discussed in Section 3. When each module, including $A_0$ and $\phi_0$ constituting DPL Attention, DPLN, and $L_{\text{attn}}$, is removed, it decreases SoM-TP’s performance. In terms of dataset robustness, we can observe through the rank results that all modules contribute to promoting SoM-TP’s Diverse Perspective Learning.

Perspective Analysis with LRP In Figure 3, we can observe that SoM-TP dynamically selects pooling during inference. ArrowHead, Chaintown, and ACSF1, in order, are datasets where GTP, STP, and DTP pooling selections are most frequently chosen. The DPL attention is trained to select the optimal pooling for each batch during the training process, and during inference, it continues to choose the most suitable pooling without DPLN (Appendix. Figure 7).
For qualitative analysis, we employ Layer-wise Relevance Propagation (LRP) to understand how different temporal pooling perspectives capture time series patterns. LRP attributes relevance to input features, signifying their contribution to the output. Note that the conservation rule maintains relevance sum in backward propagation, ensuring that the sum of attribution is 1. We use the LRP $z^+$ rule for the convolutional stack $\Phi$, and the $\epsilon$ rule for the FC layers.

In Figure 4, GTP focuses on globally crucial parts (a, e, i), while STP and DTP use local views within segmented time series for a more balanced representation (b, c, f, g, j, k). However, GTP’s limitation lies in concentrating only on specific parts, neglecting other local aspects (e, i). Conversely, STP and DTP risk diluting the primary representation by reflecting all local segments (b, j) or cutting significant segments due to forced segmentation (c, k). DTP often segments at change points, losing essential information (c, k).

SoM-TP addresses these issues by combining global and local views of each pooling method via diverse perspective learning. In Figure 4-(d), SoM-TP captures GTP’s points (d-1) and enhances multiple representations by effectively capturing local patterns (d-2, d-3). In (h), SoM-TP identifies the common important points (h-2, h-3) and complements GTP’s missed local points (h-1). Finally, in (l), SoM-TP captures GTP’s missed local points (l-1) and fully utilizes important time series (l-2) cut by STP and DTP (j-1, k-1, k-2).

**Complexity of SoM-TP**

We compare the complexity of independent temporal poolings: pooling and optimization complexity. We exclude the maximum or average operation, which is common for all pooling complexity.

As shown in Table 4, for the pooling complexity, GTP has $O(1)$ while STP and DTP have $O(L)$ from segmenting. SoM-TP has increased complexity as $O(L_P + L_{mul}) = O(L)$ for computation of the attention score, where $O(L_P)$ is for a sum of the three temporal poolings’ complexity, making it $O(L)$, and $O(L_{mul})$ for the complexity of multiplication between $\bar{P}$ and $A_0$, and between $\bar{P}$ and $A$. As for the optimization complexity, SoM-TP and other temporal poolings have all $O(N)$, while MCL has $O(N^2)$ to generate and compare multiple outputs. Therefore, compared with independent pooling, SoM-TP has little degradation of complexity, while optimization is effectively achieved even with an ensemble.

**Conclusion**

This paper proposes SoM-TP, a novel temporal pooling method employing a selection ensemble to address data dependency in temporal pooling by learning diverse perspectives. Utilizing a selection ensemble inspired by MCL, SoM-TP adapts to each data batch’s characteristics. Optimal pooling selection with DPL attention achieves a comparison-free ensemble. We define DPLN and perspective loss for effective ensemble optimization. In quantitative evaluation, SoM-TP surpasses other pooling methods and state-of-the-art TSC models in UCR/UEA experiments. In qualitative analysis, LRP results highlight SoM-TP’s ability to complement existing temporal pooling limitations. We re-examine the conventional role of temporal poolings, identify their limitations, and propose an efficient data-driven temporal pooling ensemble as a first attempt.
Acknowledgements

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation; No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics; No. 2021-0-02068, Artificial Intelligence Innovation Hub; No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).

References


