南京郵電大學

Nanjing University of Posts and Telecommunications

ShapeFormer

汇报人：黄科涛

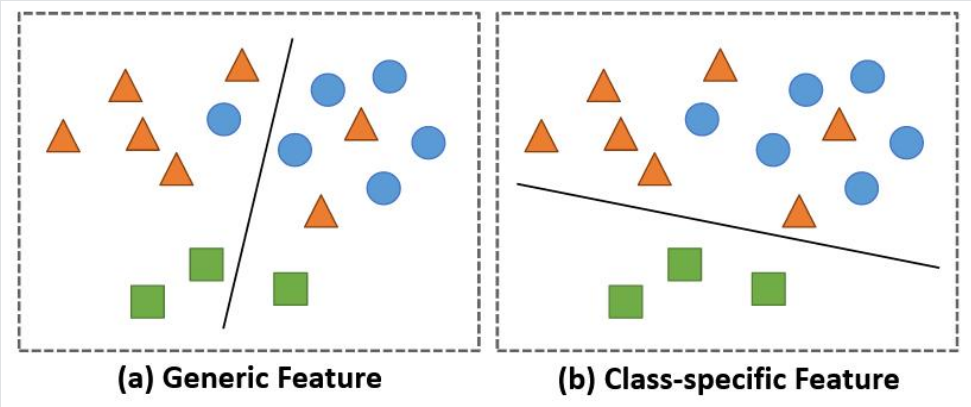
汇报时间：5月8日

论文背景

利用 Transformer 进行多变量时间序列分类(MTSC)已经取得了最先进的性能。然而，现有方法主要关注通用特征，却忽略了对于学习每个类别代表性特征至关重要的类别特定特征。

引发问题

这导致在数据集不平衡或整体模式相似但在类别特定细节上存在差异的数据集上表现不佳。



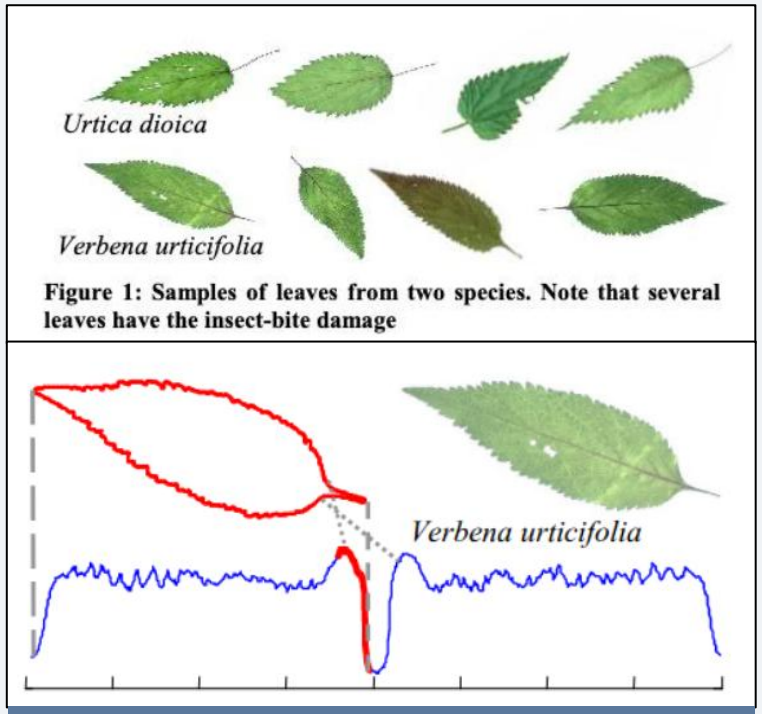
使用（a）通用特征的分离超平面具有更高的总体准确率，而使用（b）类别特定特征的分离超平面在分类单个类别时表现更好。

提出思路

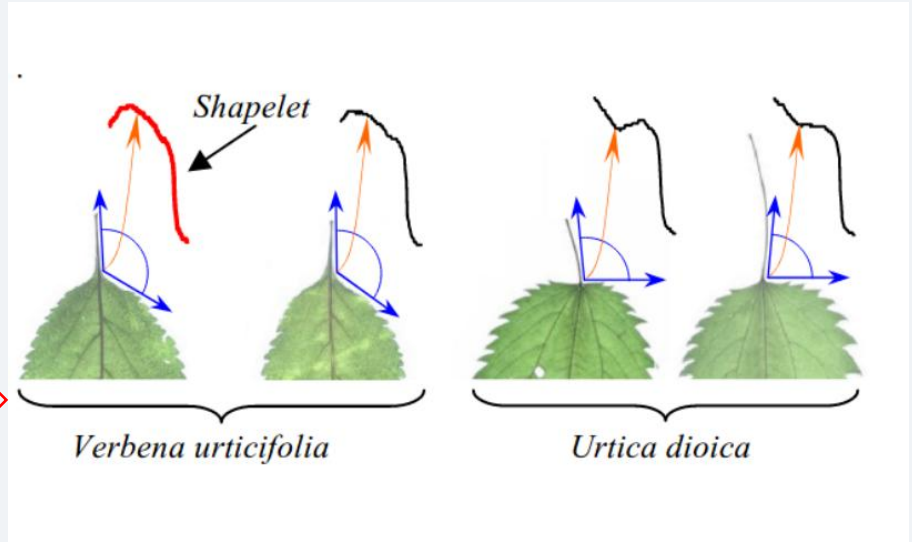
除通用特征外，从训练集中提取少量高质量的shapelets来捕捉类别特定特征

Shapelet

Lexiang Ye 等于2009年发表的论文《Time Series Shapelets: A New Primitive for Data Mining》引入了一种新的时间序列特征提取方法Shapelets



文章以荨麻草和马鞭草的分类为例。马鞭草和荨麻草非常的相似，并且还会因为虫子啃咬导致数据产生噪音。



它们的重要差别是叶柄与叶片之间的角度，一个是直角，一个是钝角。因此，如果使用序列中的小片断（子序列）作为序列的表征，就很容易将二者区分开来。

Shapelets提取

暴力法

- 利用穷举方法找到所有可选子序列作为备选项
- 找到其中信息增益最大（或若干）的片断

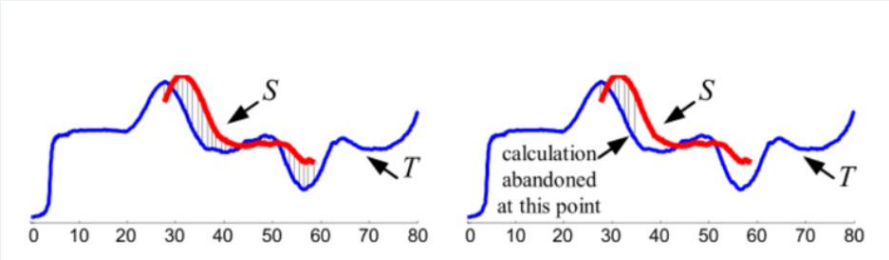
Table 2: Brute force algorithm for finding shapelet

```
FindingShapeletBF (dataset D, MAXLEN, MINLEN)
1   candidates ← GenerateCandidates(D, MAXLEN, MINLEN)
2   bsf_gain ← 0
3   For each S in candidates
4       gain ← CheckCandidate(D, S)
5       If gain > bsf_gain
6           bsf_gain ← gain
7           bsf_shapelet ← S
8       EndIf
9   EndFor
10  Return bsf_shapelet
```

时间复杂度高！$O(N^2L^3)$

改进1

子序列早弃



为减少计算量，在计算过程中发现距离比已知最大的距离还大时，则不再继续计算。

改进2

熵剪枝

- 在计算过程中，实时估计当前信息增益的上界
- 若潜在的最大IG上界(基于当前部分数据)仍无法超过目前最大值，则立即终止对该 S 的进一步计算

熵剪枝

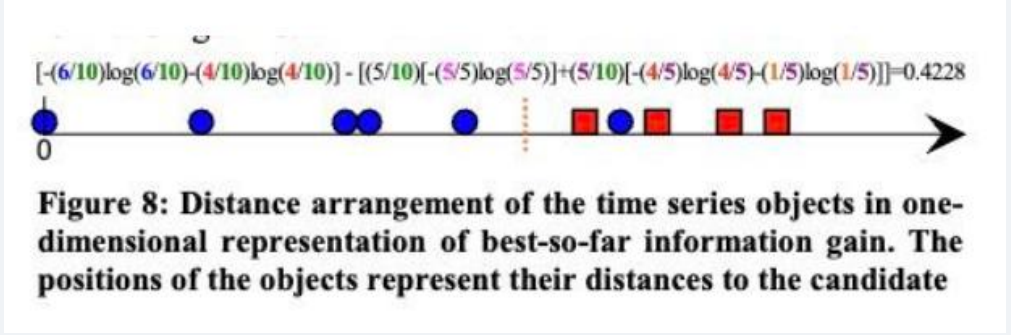


Figure 8: Distance arrangement of the time series objects in one-dimensional representation of best-so-far information gain. The positions of the objects represent their distances to the candidate

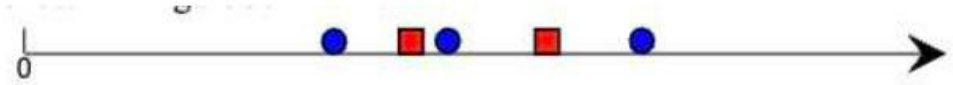


Figure 9: The arrangement of first five distances from the time series objects to the candidate

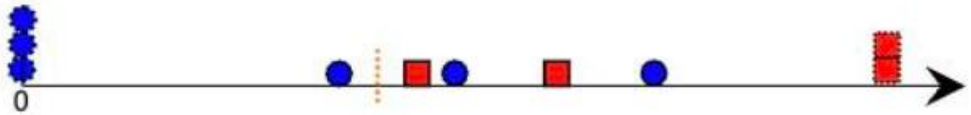


Figure 10: One optimistic prediction of distance distribution based on distances that have already been calculated in Figure 9. The dashed objects are in the optimistically assumed placements

信息增益更小

假设截止目前最好的gain如图，每个蓝色和红色都是一个要分类的样本。

根据新的子序列进行分类时，进行到一半时

假设后续分类情况是最优的，即后续的5个序列分类结果，蓝/红的全在左/右边

ShapeFormer采用的Shapelets提取方法

Amini 等于2022年发表的论文《Learning Perceptual Position-Aware Shapelets for Time Series Classification》引入了新的离线Shapelets发现方法（OSD），ShapeFormer借鉴了其主要思想

阶段1：Shapelet提取

在这一阶段，算法遍历训练集中的每条时间序列，从中提取潜在的shapelets并存储在集合 C 中。

1. 初始化空集合 C 用于存储候选的shapelets。
2. 对于训练集中每条时间序列 X：
   - 对每个变量 v（总共 V 个）：
     - 初始化集合 P，并将时间序列的第一个和最后一个索引加入 P 作为初始PIPs。
     - 通过循环（从1到 $k-2$），选择新的PIP点，使得每次新选择的点能够最大化与当前PIPs的**重构距离**。
     - 将新点加入 P 并按索引排序。
     - 验证新生成的候选PIPs：
       - 确保新点位置不会超出边界，保持在合理范围。
       - 通过索引 idx_s 和 idx_e 来确定当前shapelet的开始和结束位置。
       - 将新的shapelet候选 C 连同其开始索引、结束索引、变量索引加入 C。

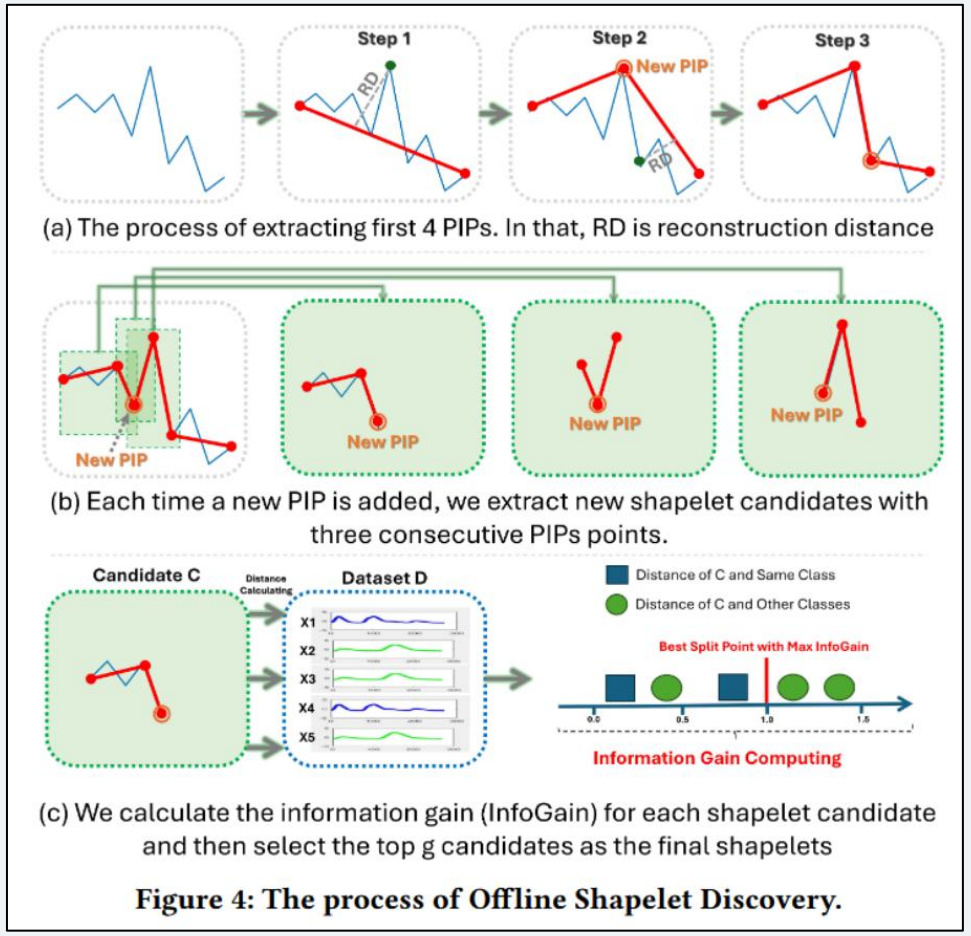


Figure 4: The process of Offline Shapelet Discovery.

ShapeFormer采用的Shapelets提取方法

南京郵電大學 Nanjing University of Posts and Telecommunications

阶段2：Shapelet选择

在这一阶段，算法从候选shapelets中选择具有最高信息增益的shapelets，以提高区分不同类别的能力。

1. 对于每个候选shapelet S_i：
   - 初始化距离集合 D 用于存储 S_i 与其他序列的PSD距离。
   - 对训练集中的每条时间序列 X：
     - 计算 X 与 S_i 的PSD距离 d（使用等式1）。
     - 将距离 d 加入 D。
   - 基于距离集合 D 计算shapelet的最优信息增益。
2. 初始化集合 S 用于存储最终选择的shapelets。
3. 对于每个类别 Y_i：
   - 从 C 中选择信息增益排名前 g 的shapelets，加入到集合 S。
4. 返回最终选择的shapelet集合 S。

原论文中，作者提到PIP点的数量为0.2*T，其中T为时间序列的长度。在实验中，平均为每个数据集提取的的候选shapelets个数仅为5500，若采用传统方法，这个数字将达到4500万个。

ShapeFormer结构

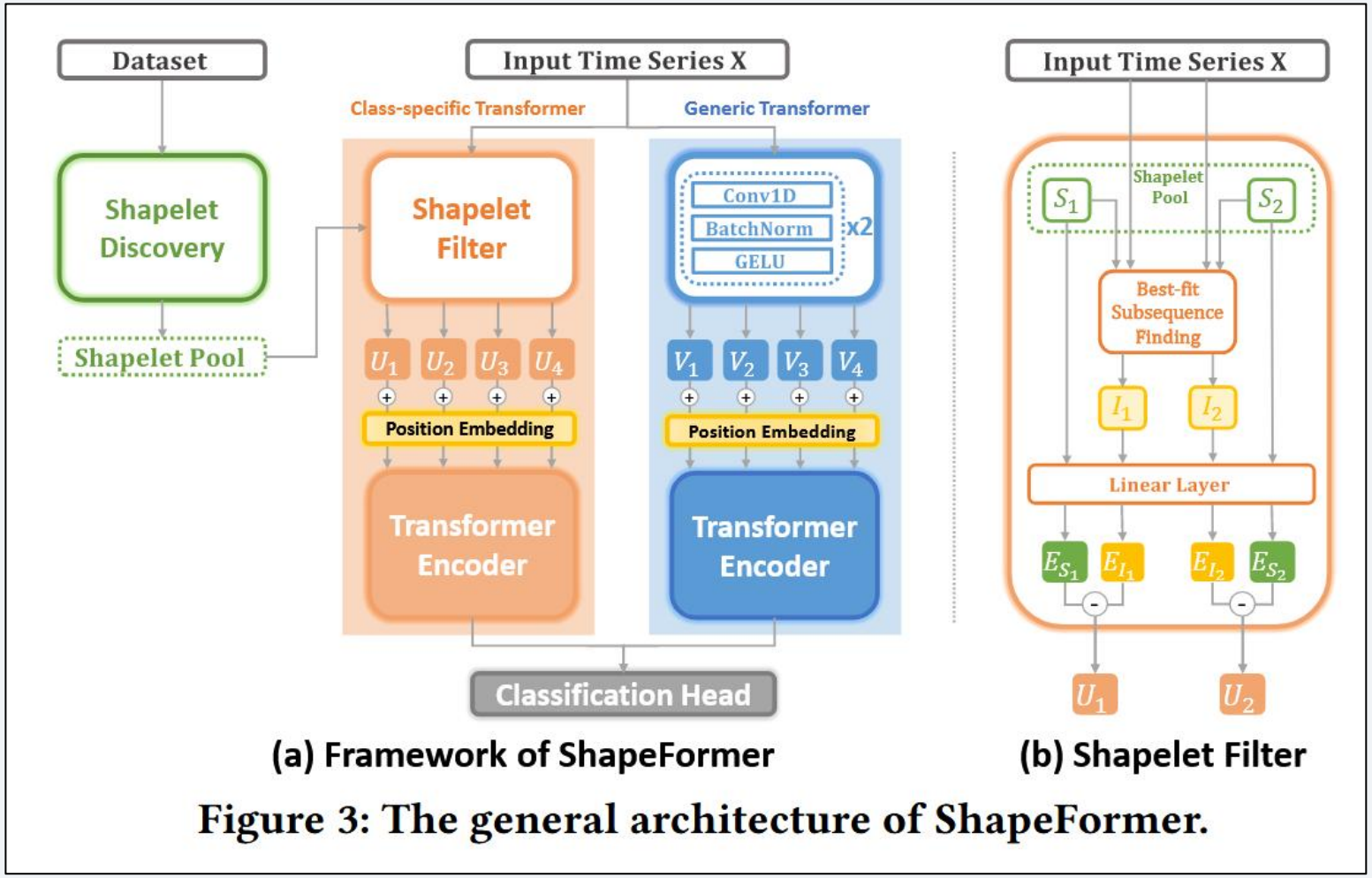
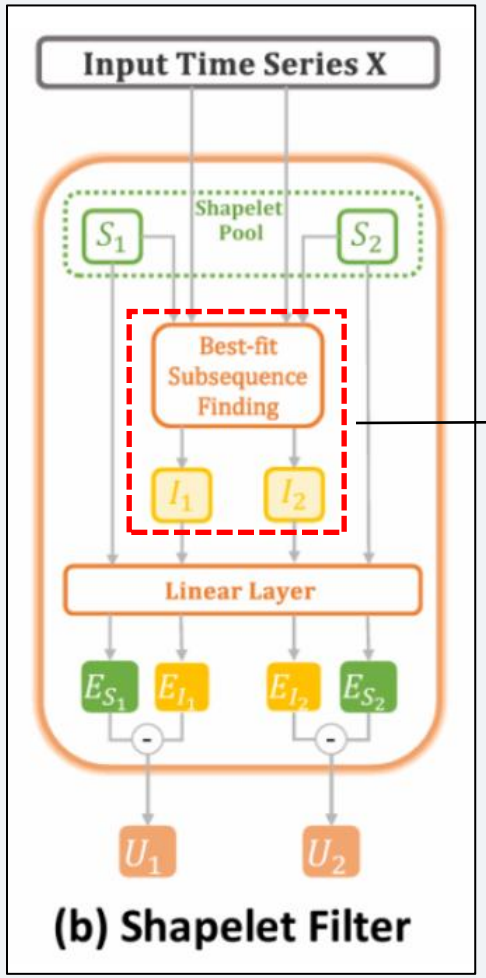
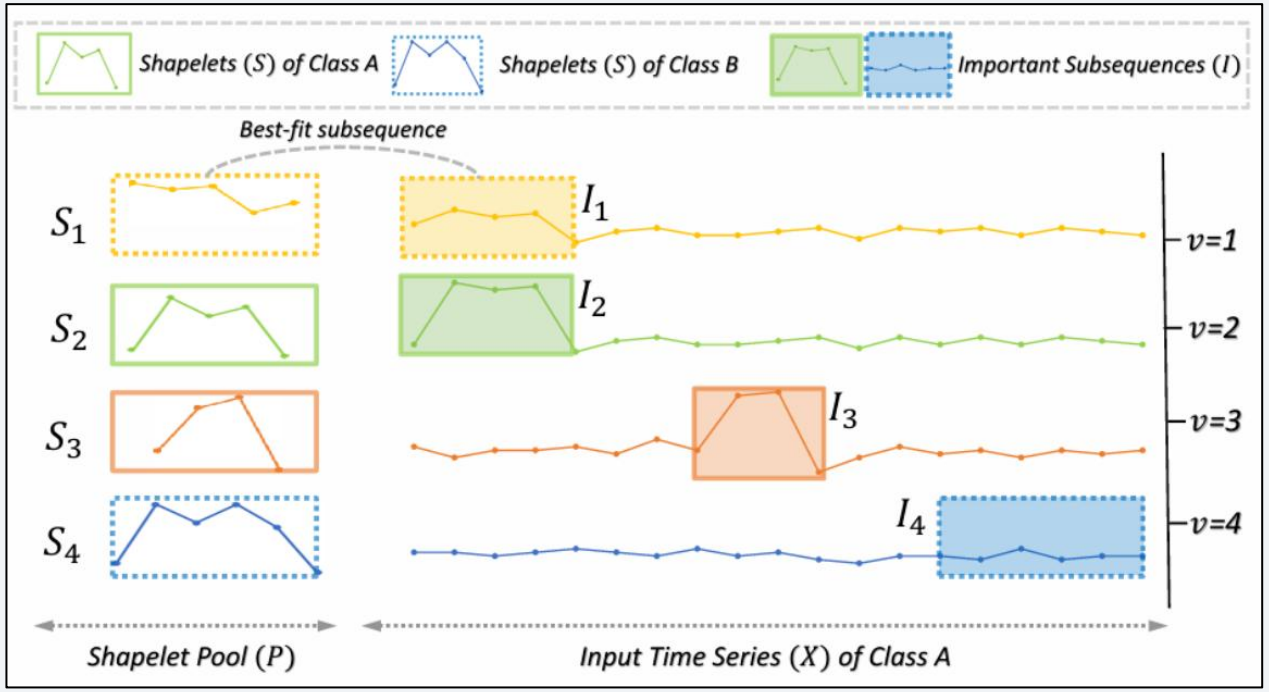


Figure 3: The general architecture of ShapeFormer.

Shapelet Filter



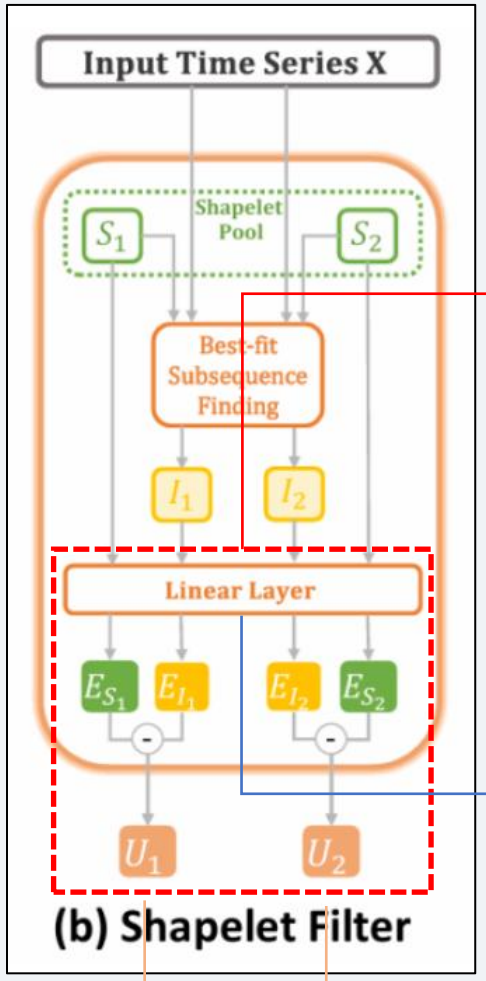


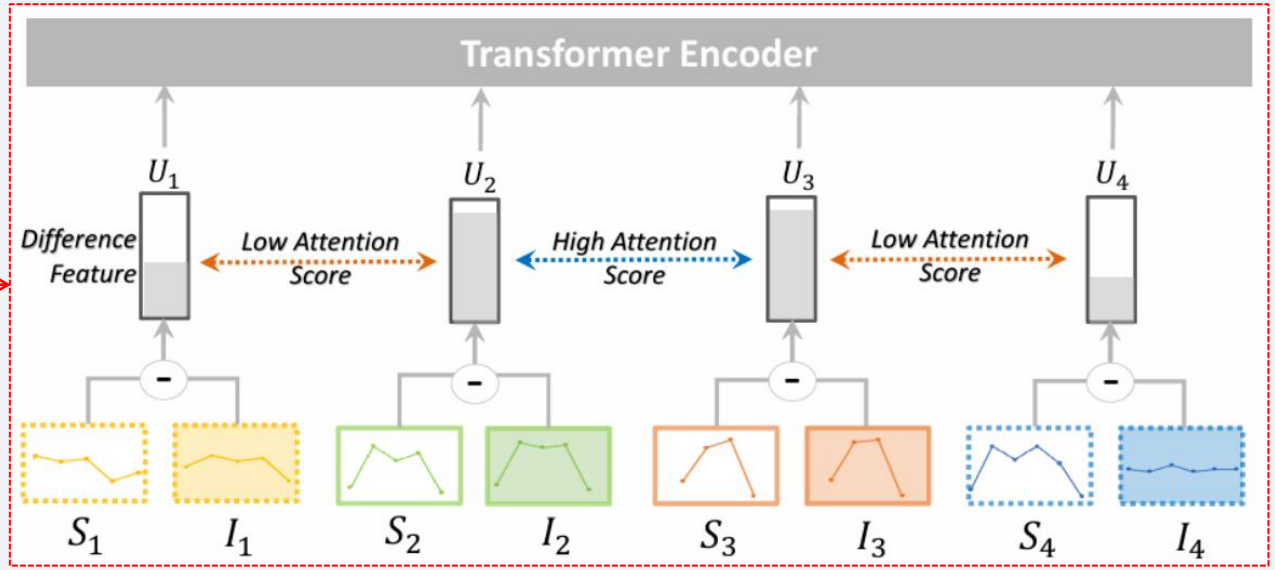
为减少计算时间并有效利用 shapelet 的位置信息，作者提出将最佳匹配子序列的搜索限制在 shapelet 实际位置左右两侧的超参数窗口大小w 内的邻近区域

$$\text{index} = \text{argmin}_{j=0}^{T-l+1} CID(X[j:j+l], S_i)\ ,$$
$$I_i = X[\text{index} : \text{index} + l]\ .$$

$$\text{CID}(X,Y) = \text{ED}(X,Y) \times \text{CF}(X,Y)$$
$$\text{CF}(X,Y) = \frac{\max(\text{CE}(X), \text{CE}(Y))}{\min(\text{CE}(X), \text{CE}(Y))}$$
$$\text{CE}(X) = \sqrt{\sum_{t=2}^{T}(x_t - x_{t-1})^2}$$

Shapelet Filter



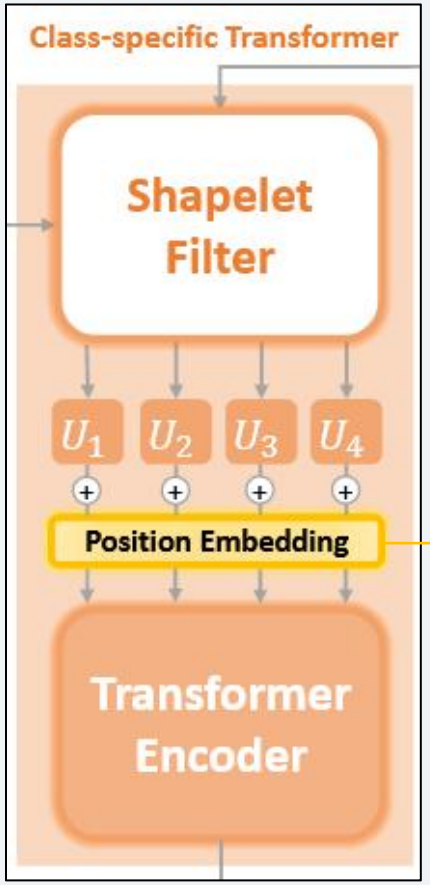


线性投影器 P 是一个可学习的矩阵，用于将Shapelet和输入时间序列的最佳匹配子序列映射到一个高维嵌入空间，**目的是：**

- 提取高阶差异特征：通过线性变换，将原始子序列的局部模式转换为更具判别性的特征
- 实现可学习性：Shapelet本身作为参数参与投影计算，使得模型能通过反向传播优化Shapelet的形状和投影矩阵，而不仅仅是静态匹配。
- 适配Transformer架构，提供高维输入

差异特征计算：$U_i = \mathcal{P}_I(I_i) - \mathcal{P}_S(S_i)$

Class-Specific Transformer的其他结构



独热向量（One-Hot）

每个位置索引（如 p_{start}）转换为独热编码向量 $\text{one-hot}(p) \in \mathbb{R}^L$，其中 L 是时间序列的最大长度（或变量总数）。

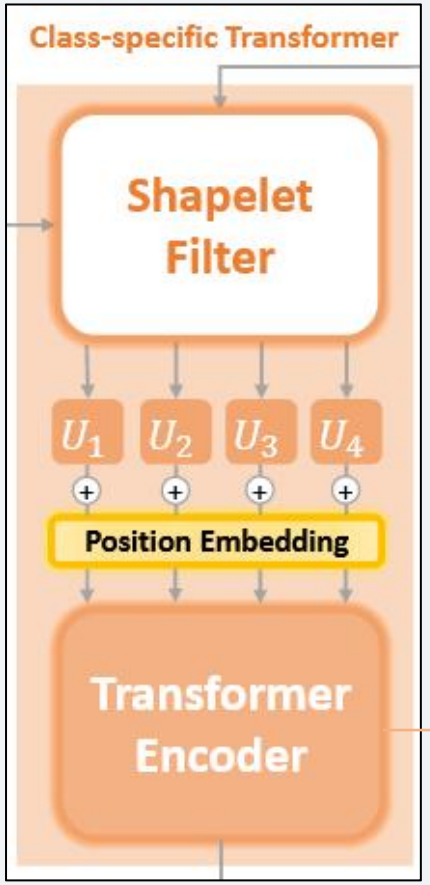
- 例如，若时间序列长度 $L = 100$，$p_{\text{start}} = 10$，则 $\text{one-hot}(p_{\text{start}})$ 是第10维为1、其余为0的100维向量。

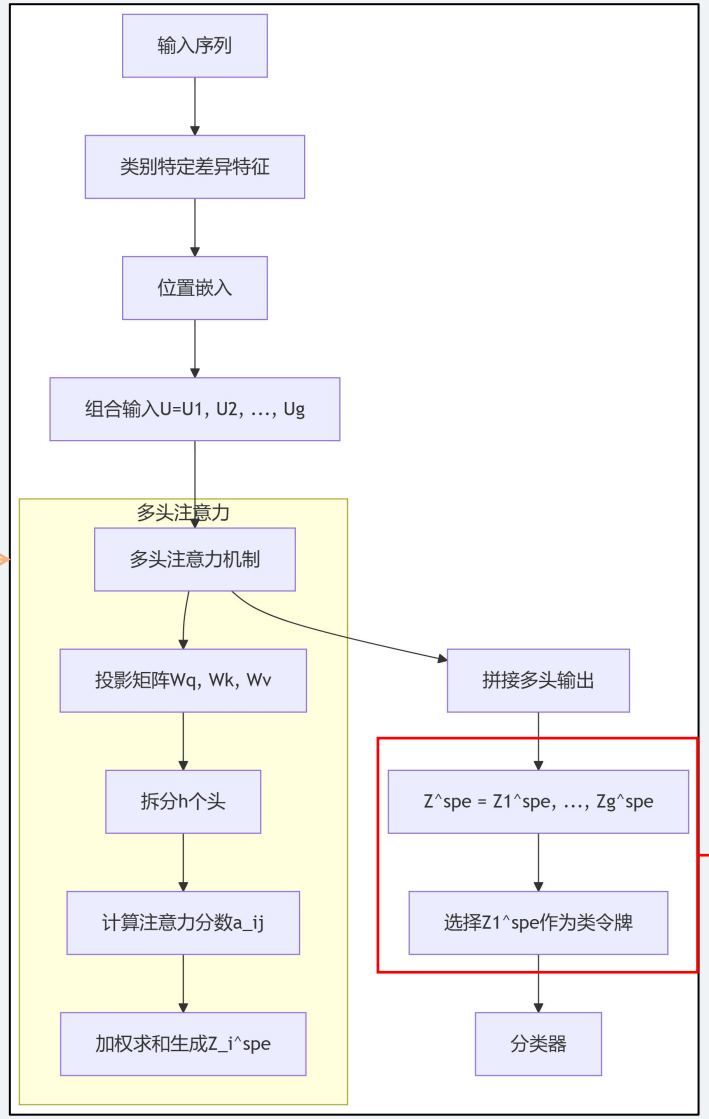
$$\text{PE}(p) = \text{Linear}(\text{one-hot}(p)),$$
$$U_i = U_i + \text{PE}(p_s^i) + \text{PE}(p_e^i) + \text{PE}(v^i).$$

通过位置编码，让模型感知Shapelet的起始位置、结束位置和所属变量（维度），从而区分不同位置的Shapelet对分类的贡献差异。

- 起始索引：Shapelet在原始时间序列中的开始时间戳
- 结束索引：Shapelet的结束时间戳
- 变量索引：Shapelet所属的多元时间序列维度（如第几个导联）

Class-Specific Transformer的其他结构





原有方法的缺陷

现有的基于 transformer 的方法将平均池化应用于 Z_{spe}，以获得用于分类的最终标记。然而，特定于类别的 transformer 模块利用了不同的功能来捕捉每个 shapelet 的独特特征。应用平均池化可能会减少这些属性，从而可能限制性能。

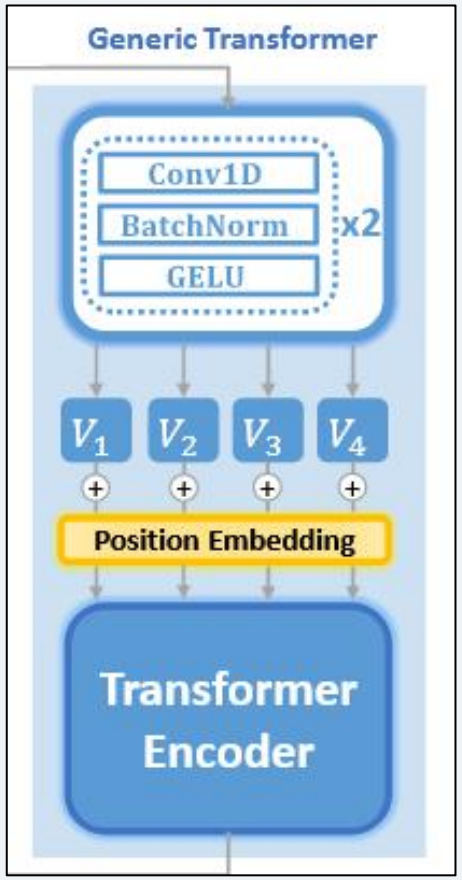
改进

仅使用最高信息增益 shapelet 的第一个差分特征作为最终分类的类令牌

改进的原因

在平均所有标记时，会丢失有关Ui不同功能的信息 。此外，使用该方法的类令牌具有最高的信息增益，具有有效分类时间序列的最关键特征。

Generic Transformer



$$\text{ConvBlock}(X) = \text{GELU}(\text{BatchNorm}(\text{Conv1D}(X))),$$
$$V = \text{ConvBlock}(\text{ConvBlock}(X)).$$

由于CNN输出的特征 V 缺乏位置信息，
引入可学习的位置嵌入

$$Z^{\text{gen}} = \text{MHA}(V + P),$$
$$Z_*^{\text{gen}} = \text{AvgPooling}(Z^{\text{gen}}).$$

Experimental Setting

数据集

采用UEA数据集，这是一个由 30 个不同的 MTSC 数据集组成的著名数据集。它涵盖多个领域，包括人类活动识别、运动分类、心电图分类、脑电图/脑磁图分类、音频频谱分类等

评估

使用分类准确性来评估模型性能
使用Friedman和Wilcoxon符号秩检验的p值来评估表现差异的统计学意义

实现细节

- RAdam 优化器
- 初始学习率：0.01
- 权重衰减：5e-4
- epoch：200
- 多头数量：16
- batch_size：16
- Momentum：0.9

比较的方法

基于距离的方法：EDI、DTWD

基于模式的算法：WEASEL+MUSE

基于特征的算法：MiniRocket

一种集成方法：LCEM

三种深度学习模型：MLSTM-FCN 、Tapnet、Shapenet

基于注意力的模型：WHEN

三种基于transformer的模型：TST 、ConvTran、SVP-T

Experimental Result

	EDI	DTWD	WEASEL +MUSE	MiniRocket	LCEM	MLSTM -FCNs	Tapnet	Shapenet	WHEN	TST	ConvTran	SVPT	Our
ArticularyWordRecognition	0.970	0.987	0.990	0.992	**0.993**	0.973	0.987	0.987	**0.993**	0.983	0.983	**0.993**	**0.993**
AtrialFibrillation	0.267	0.220	0.333	0.133	0.467	0.267	0.333	0.400	0.467	0.200	0.400	0.400	**0.660**
BasicMotions	0.676	0.975	**1.000**	**1.000**	**1.000**	0.950	**1.000**	**1.000**	**1.000**	0.975	**1.000**	**1.000**	**1.000**
CharacterTrajectories	0.964	0.989	0.990	0.993	0.979	0.985	**0.997**	0.980	0.996	0.000	0.992	0.990	0.996
Cricket	0.944	**1.000**	**1.000**	0.986	0.986	0.917	0.958	0.986	**1.000**	0.958	**1.000**	**1.000**	**1.000**
DuckDuckGeese	0.275	0.600	0.575	0.650	0.375	0.675	0.575	**0.725**	0.700	0.480	0.620	0.700	**0.725**
ERing	0.133	0.929	0.133	**0.981**	0.200	0.133	0.133	0.133	0.959	0.933	0.963	0.937	0.966
EigenWorms	0.549	0.618	0.890	**0.962**	0.527	0.504	0.489	0.878	0.893	N/A	0.593	0.925	0.925
Epilepsy	0.666	0.964	**1.000**	**1.000**	0.986	0.761	0.971	0.987	0.993	0.920	0.986	0.986	0.993
EthanolConcentration	0.293	0.323	0.430	**0.468**	0.372	0.373	0.323	0.312	0.422	0.337	0.361	0.331	0.378
FaceDetection	0.519	0.529	0.545	0.620	0.614	0.545	0.556	0.602	0.658	**0.681**	0.672	0.512	0.658
FingerMovements	0.550	0.530	0.490	0.550	0.590	0.580	0.530	0.580	0.660	**0.776**	0.560	0.600	0.700
HandMovementDirection	0.278	0.231	0.365	0.392	**0.649**	0.365	0.378	0.338	0.554	0.608	0.405	0.392	0.486
Handwriting	0.200	0.286	**0.605**	0.507	0.287	0.286	0.357	0.452	0.561	0.305	0.375	0.433	0.507
Heartbeat	0.619	0.717	0.727	0.771	0.761	0.663	0.751	0.756	0.780	0.712	0.785	0.790	**0.800**
InsectWingbeat	0.128	N/A	N/A	0.595	0.228	0.167	0.208	0.250	0.657	0.684	**0.713**	0.184	0.314
JapaneseVowels	0.924	0.949	0.973	0.989	0.978	0.976	0.965	0.984	0.995	0.994	0.989	0.978	**0.997**
LSST	0.456	0.551	0.590	0.643	0.652	0.373	0.568	0.590	0.663	0.381	0.616	0.666	**0.700**
Libras	0.833	0.870	0.878	0.922	0.772	0.856	0.850	0.856	0.933	0.844	0.928	0.883	**0.961**
MotorImagery	0.510	0.500	0.500	0.550	0.600	0.510	0.590	0.610	0.630	N/A	0.560	0.650	**0.670**
NATOPS	0.850	0.883	0.870	0.928	0.916	0.889	0.939	0.883	0.978	0.900	0.944	0.906	**0.989**
PEMS-SF	**0.973**	0.711	N/A	0.522	0.942	0.699	0.751	0.751	0.925	0.919	0.828	0.867	0.925
PenDigits	0.705	0.977	0.948	N/A	0.977	0.978	0.980	0.977	0.987	0.974	0.987	0.983	**0.990**
PhonemeSpectra	0.104	0.151	0.190	0.292	0.288	0.110	0.175	0.298	0.293	0.088	**0.306**	0.176	0.293
RacketSports	0.868	0.803	0.934	0.868	**0.941**	0.803	0.868	0.882	0.934	0.829	0.862	0.842	0.895
SelfRegulationSCP1	0.771	0.775	0.710	**0.925**	0.839	0.874	0.652	0.782	0.908	**0.925**	0.918	0.884	0.911
SelfRegulationSCP2	0.483	0.539	0.460	0.522	0.550	0.472	0.550	0.578	0.589	0.589	0.583	0.600	**0.633**
SpokenArabicDigits	0.967	0.963	0.982	0.620	0.973	0.990	0.983	0.975	**0.997**	0.993	N/A	0.986	**0.997**
StandWalkJump	0.200	0.200	0.333	0.333	0.400	0.067	0.400	0.533	0.533	0.267	0.333	0.467	**0.600**
UWaveGestureLibrary	0.881	0.903	0.916	0.938	0.897	0.891	0.894	0.906	0.919	0.903	0.891	**0.941**	0.922
Average rank	11.200	9.783	7.933	5.900	6.600	9.833	8.233	6.850	3.117	8.117	5.650	5.283	**2.500**
Number of top-1	1	1	4	6	4	0	2	2	4	3	4	5	**15**
Wins	29	29	24	21	25	30	28	27	16	25	24	24	-
Draws	0	1	2	2	2	0	1	2	9	0	2	5	-
Loses	1	0	4	7	3	0	1	1	5	5	4	1	-
P-value	0.000	0.000	0.001	0.014	0.003	0.000	0.000	0.002	0.475	0.005	0.024	0.000	-

Ablation Study and Model Design

Effectiveness of Using Shapelets

作者比较了使用随机子序列、常见子序列和shapelet 时的性能。结果表明，在所有五个数据集中，shapelet 的准确性都优于其他两种方法。

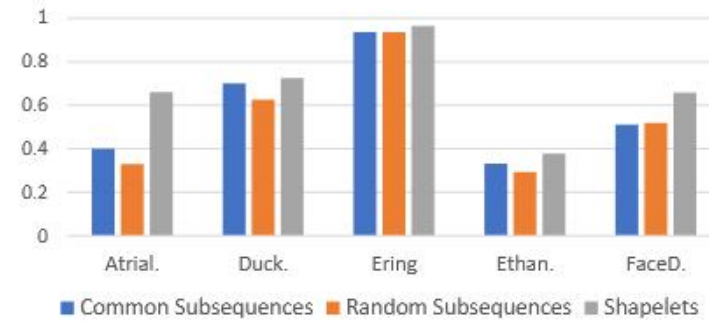


Figure 6: Accuracies of using shapelets and two other types of subsequences.

Component Evaluation

评估 ShapeForm 中两个关键模块的影响

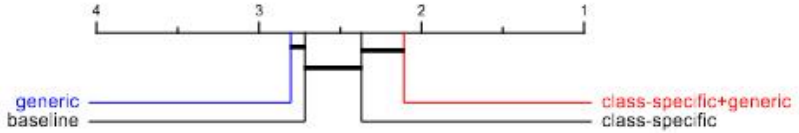


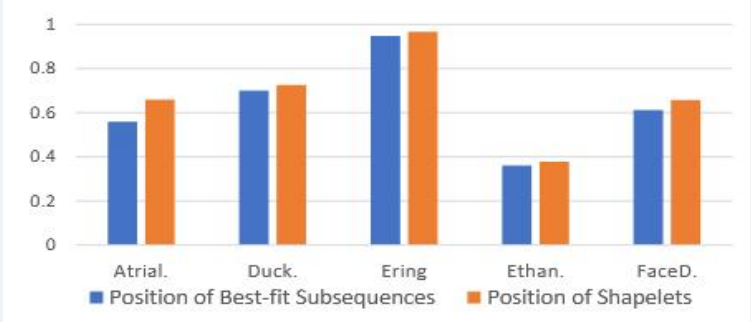
Figure 7: Average ranks for 3 variations of ShapeFormer and the baseline (SVP-T [50] - the current SOTA transformer-based method).

Ablation Study and Model Design

Choosing between the Position of Shapelets and Best-fit Subsequences

$$U_i = U_i + \mathrm{PE}(p_s^i) + \mathrm{PE}(p_e^i) + \mathrm{PE}(v^i)$$

在位置嵌入时，是嵌入shapelet的位置还是最佳匹配子序列的位置?



Comparison with Various Methods for Calculating Difference Features

$$U_i = \mathcal{P}_I(I_i) - \mathcal{P}_S(S_i)\,,$$

差异特征，还是曼哈顿距离or欧几里得距离?

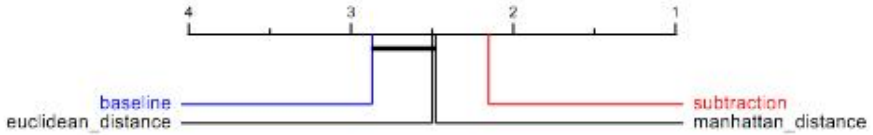


Figure 9: Average accuracy ranks of various calculation methods for difference features.

Different Class Token Designs

平均池化与信息增益最大shapelet的第一个差异特征之间的比较

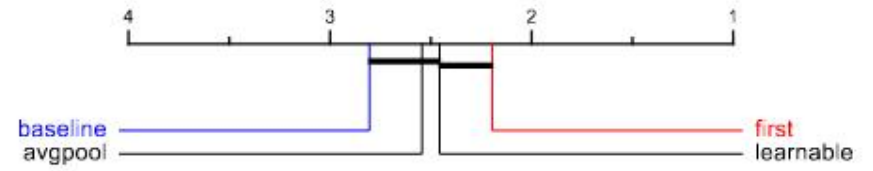


Figure 10: Average accuracy ranks of different class token designs.

Hyperparameter Sensitivity

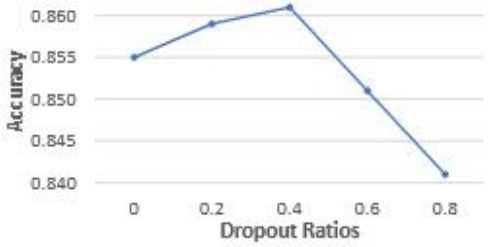
- Number of PIPs：从 0.05T 增加到 0.2T，模型精度也会提高。之后，即使 npips 进一步增加，准确性也保持稳定
- The Scale Factors d_{spe} and d_{gen}：在图 11a 中，比较了特定于类的嵌入大小和通用嵌入大小的不同比例因子对模型分类准确性的影响。更大的类特定嵌入大小取得了更好的性能
- Dropout Ratios：分析了 ShapeFormer 的不同dropout的影响，为0.4时可产生最好的性能

Table 2: The average accuracy for various numbers of PIPs.

Npips ($\times T$)	0.05	0.1	0.15	**0.2**	0.25	0.3	0.4	0.5
Accuracy	0.832	0.848	0.856	**0.864**	0.864	0.864	0.864	0.864

d_{gen} \ d_{spe}	32	64	128	256
32	0.845	0.858	**0.864**	0.86
64	0.838	0.855	0.861	0.852
128	0.834	0.842	0.859	0.858
256	0.829	0.836	0.844	0.855

(a) Different scale factors d_{global} and d_{local}



(b) Different dropout ratios

Figure 11: Effectiveness of (a) class-specific and generic scale factors and (b) different dropout ratios.

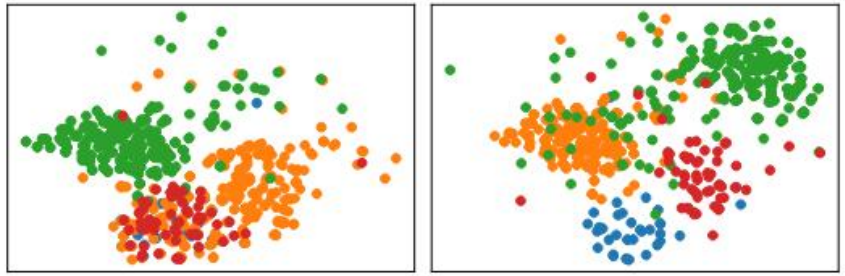
Other Experiment

就 InsectWingbeat 数据集而言，作者观察到，与 32 （0.314） （作者选择的参数） 的 d_{gen} 相比，将 d_{gen} （通用特征的嵌入大小） 设置为 256 会导致性能明显提高 （0.704）。但是，这种改进是以其他数据集的性能降低（从 0.864 降低到 0.831）为代价的。

Table 3: Accuracy for InsectWingbeat dataset and the first 10 UEA datasets with various d_{gen} factors.

d_{gen}	32 (our choice)	64	128	256
InsectWingbeat	0.314	0.500	0.634	**0.704**
First 10 UEA datasets	**0.864**	0.852	0.844	0.831

在 不平衡数据集LSST 上进行了实验。该数据集包括 16 个类，作者随机选择了 4 个类，分别用蓝色、橙色、绿色和红色表示，分别有 35、270、382 和 63 个实例。图 12a 显示，通用变压器优先考虑多数类（绿色和橙色），但忽略了少数类（蓝色和红色）。然而，在图 12b 中，特定于类的转换器和通用转换器的组合有效地区分了所有四个类。



(a) Generic Transformer (b) Class-specific and Generic Transformer

Figure 12: The t-SNE visualisation in 4 classes of LSST dataset using (a) our generic transformer and (b) both class-specific and generic transformers. Each point indicates an instance and the colors of the points signify the true labels.

南京郵電大學

Nanjing University of Posts and Telecommunications

谢 谢！

汇报人：黄科涛