Duke University School of Medicine

Learning Representations for Time Series Clustering

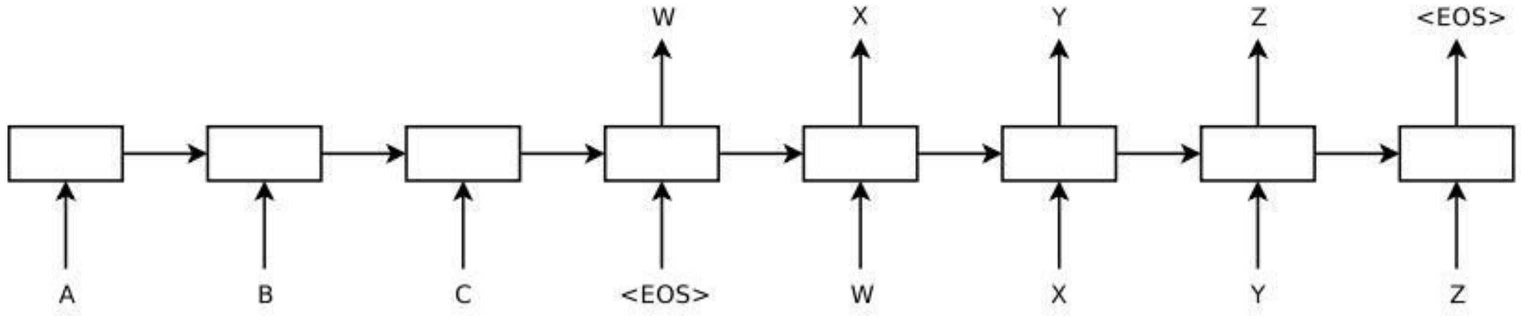
Qianli Ma, Jiawei Zheng, Sen Li, Garrison Cottrell

Present by

Yuankang Zhao

3.27.2023

Background



Seq2seq model

- Classical model in NLP
- The model can learn general representations from sequence data in an unsupervised manner by designing appropriate learning objectives
- Example: Reconstruction and context prediction

Background

Raw data method

- Raw-data-based methods mainly modify the distance function to adapt to the time series characteristics (e.g., scaling and distortion).
- Example: K-means based method
- Sensitive to outliers and noise, since all time points are taken into account
- Unstable result

Background

Feature-based methods

- Feature-based methods use clustering algorithms on the extracted feature representations of input time series, which mitigates the impact of noise or outliers while also reducing the dimensionality of the data
- Reduce the dimension first, then clustering
- Two approaches:
- (i) two-stage approaches that cluster after extracting features
- **(ii) jointly optimize the feature learning and clustering.**

Method

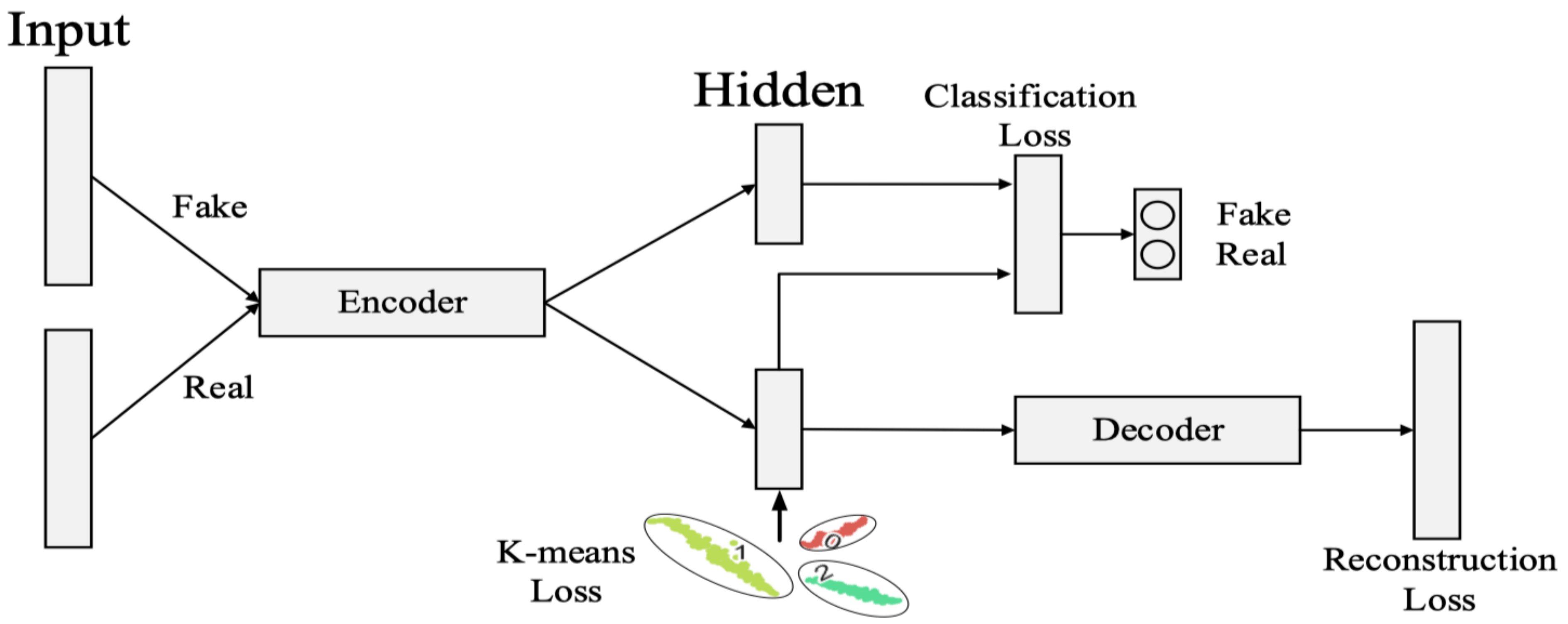


Figure 1: The general architecture of the Deep Temporal Clustering Representation (DTCR).

Method

Deep Temporal Representation Clustering- Reconstruction Loss

Given a set of n time series $\boldsymbol{D} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}\}$, each time series $\boldsymbol{x_i}$ contains T ordered real values denoted as $\boldsymbol{x_i} = (x_{i,1}, x_{i,2}, ...x_{i,T})$. Define non-linear mappings $f_{enc} : \boldsymbol{x_i} \rightarrow \boldsymbol{h_i}$ and $f_{dec} : \boldsymbol{h_i} \rightarrow \hat{\boldsymbol{x}}_{\boldsymbol{i}}$. f_{enc}, f_{dec}, denotes the encoding and decoding process, respectively. $\boldsymbol{h_i} \in \mathbb{R}^m$ is the m-dimensional latent representation of time series $\boldsymbol{x_i}$, defined by:

$$\boldsymbol{h_i} = f_{enc}(\boldsymbol{x_i}) \tag{1}$$

We aim to train a good f_{enc}, making the learned representations facilitate the clustering task. We instantiate the non-linear mapping as a bidirectional RNN. Furthermore, considering that time series are commonly multi-scale, the encoder RNN is instantiated by a multi-layer Dilated RNN The latent representation is obtained by concatenating the last hidden state output of each layer of the Dilated RNN. After decoding, we can obtain the output $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$, where $\hat{\boldsymbol{x}}_{\boldsymbol{i}} \in \mathbb{R}^T$ is given by:

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = f_{dec}(\boldsymbol{h_i}) \tag{2}$$

We use Mean Square Error (MSE) as the reconstruction loss, which is defined by:

$$\mathcal{L}_{reconstruction} = \frac{1}{n} \sum_{i=1}^{n} \| \boldsymbol{x_i} - \hat{\boldsymbol{x}}_{\boldsymbol{i}} \|_2^2 \tag{3}$$

Method

Deep Temporal Representation Clustering- K-means Loss

Reformulate the minimization of K-means to a trace maximization problem

$$\mathcal{L}_{K-means} = Tr(\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}) - Tr(\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}\boldsymbol{F}) \quad (4)$$

where Tr denotes the matrix trace. $\boldsymbol{F} \in \mathbb{R}^{N \times k}$ is the cluster indicator matrix. Considering $\boldsymbol{H}$ is given, the minimization of Eq. (4) can be further relaxed to a trace maximization problem by setting $\boldsymbol{F}$ to be an arbitrary orthogonal matrix:

$$\max_{\boldsymbol{F}} Tr(\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}\boldsymbol{F}),\ s.t.\ \boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{F} = \boldsymbol{I} \quad (5)$$

Together with reconstruction loss and classification loss

$$\min_{\boldsymbol{H},\boldsymbol{F}} J(\boldsymbol{H}) + \frac{\lambda}{2}[Tr(\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}) - Tr(\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}\boldsymbol{F})],\ s.t.\ \boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{F} = \boldsymbol{I} \quad (6)$$

Method

Deep Temporal Representation Clustering

Classification Loss

- Fake Sample strategy:

Given a time series $X_i \in \mathbb{R}^T$, they generate its fake version by randomly shuffling some time steps. The number of selected time steps is $[\alpha \times T]$, where $\alpha \in (0, 1]$ is a hyper-parameter

- Loss function:

$$\hat{\boldsymbol{y}}_{\boldsymbol{i}} = \boldsymbol{W}_{fc2}(\boldsymbol{W}_{fc1}\boldsymbol{h}_{\boldsymbol{i}}) \tag{7}$$

$$\mathcal{L}_{classification} = -\frac{1}{2N}\sum_{i=1}^{2N}\sum_{j=1}^{2} 1\{y_{i,j} = 1\} \log \frac{\exp \hat{y}_{i,j}}{\sum_{j=1}^{2} \exp(\hat{y}_{i,j})} \tag{8}$$

where $\boldsymbol{y}_{\boldsymbol{i}}$ is a 2-dim one-hot vector indicating real or fake, and $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the classification result. For simplicity, we ignore the bias term. $\boldsymbol{W}_{fc1} \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_{fc2} \in \mathbb{R}^{d \times 2}$ are parameters of the fully connected layers and d is set to 128.

Method

Deep Temporal Representation Clustering

Overall Loss and Training

Finally, the overall training loss $\mathcal{L}_{DTCR}$ of DTCR is defined by:

$$\mathcal{L}_{DTCR} = \mathcal{L}_{reconstruction} + \mathcal{L}_{classification} + \lambda \mathcal{L}_{K-means} \tag{9}$$

Algorithm 1 DTCR Training Method

Input: Data set: $\boldsymbol{D}$; Number of clusters: K; Alternate update: T; Maximum iterations: $MaxIter$
Output: Cluster result s

For each time series in $\boldsymbol{D}$, generate the corresponding fake samples.
for $\boldsymbol{iter} = 1$ to $MaxIter$ **do**
    Update latent representation $\{\boldsymbol{h}_i = f_{enc}(\boldsymbol{x_i})\}_{i=1}^{n}$ using SGD based on Eq. (9).
    if $\boldsymbol{iter} \ \% \ T \ = \ 0$ **then**
        Update F using the closed-form solution of Eq. (5).
    end if
end for
Apply K-means to the learned representation and get the cluster result s.

Experiment

Data source:

UCR Time Series Data

Table 1: Statistics of the benchmark time series datasets

No.	Dataset	#Train/Test	Length	#classes	No.	Dataset	#Train/Test	Length	#classes
1	Arrow	36/175	252	3	19	Mid.phal.outl.correct	291/600	81	2
2	Beef	30/30	471	5	20	Mid.phal.TW	154/399	81	6
3	BeetleFly	20/20	513	2	21	MoteStrain	20/1252	85	2
4	BirdChicken	20/20	513	2	22	OSULeaf	200/242	428	6
5	Car	60/60	578	4	23	Plane	105/105	145	7
6	ChlorineConcentration	467/3840	167	3	24	Prox.phal.outl.ageGroup	400/205	81	3
7	Coffee	28/28	287	2	25	Prox.phal.TW	205/400	81	6
8	DiatomsizeReduction	16/306	346	4	26	SonyAIBORobotSurface	20/601	71	2
9	Dist.phal.outl.agegroup	139/400	81	3	27	SonyAIBORobotSurfaceII	27/953	66	2
10	Dist.phal.outl.correct	276/600	81	2	28	SwedishLeaf	500/625	129	15
11	ECG200	100/100	97	2	29	Symbols	25/995	399	6
12	ECGFiveDays	23/861	137	2	30	ToeSegmentation1	40/228	278	2
13	GunPoint	50/150	151	2	31	ToeSegmentation2	36/130	344	2
14	Ham	109/105	432	2	32	TwoPatterns	1000/4000	129	4
15	Herring	64/64	513	2	33	TwoLeadECG	23/1139	83	2
16	Lighting2	60/61	638	2	34	Wafer	1000/6164	153	2
17	Meat	60/60	449	3	35	Wine	57/54	235	2
18	Mid.phal.outl.agegroup	154/400	81	3	36	WordsSynonyms	267/638	271	25

Experiment

Experiment Setting:

- Employ the bidirectional multi-layer Dilated RNN as the encoder
- Fixed the number of layers and the number of dilation per layer to 3 and 1, 4, and 16, respectively
- Decoder is a single-layer RNN with GRU unit
- Metric:

  Rand Index and Normalized Mutual Information

$$RI = \frac{TP + TN}{n(n-1)/2}$$

$$NMI = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} N_{ij} \log(\frac{N \cdot N_{ij}}{|G_i||A_j|})}{\sqrt{(\sum_{i=1}^{M} |G_i| \log \frac{|G_i|}{N})(\sum_{j=1}^{M} |A_j| \log \frac{A_j}{N})}}$$

Table 1: Rand Index (RI) comparisons on 36 time series datasets (the values in parentheses present standard deviations)

Dataset	K-means [37]	UDFS [6]	NDFS [7]	RUFS [8]	RSFS [9]	KSC [22]	KDBA [20]	k-shape [5]	u-shapelet [24]	DTC [25]	USSL [29]	DEC [26]	IDEC [27]	DTCR
Arrow	0.6905	0.7254	0.7381	**0.7476**	0.7108	0.7254	0.7222	0.7254	0.6460	0.6692	0.7159	0.5817	0.6210	0.6868(0.0026)
Beef	0.6713	0.6759	0.7034	0.7149	0.6975	0.7057	0.6713	0.5402	0.6966	0.6345	0.6966	0.5954	0.6276	**0.8046**(0.0018)
BeetleFly	0.4789	0.4949	0.5579	0.6053	0.6516	0.6053	0.6052	0.6053	0.7314	0.5211	0.8105	0.4947	0.6053	**0.9000**(0.0001)
BirdChicken	0.4947	0.4947	0.7316	0.5579	0.6632	0.7316	0.6053	0.6632	0.5579	0.4947	**0.8105**	0.4737	0.4789	**0.8105**(0.0033)
Car	0.6345	0.6757	0.6260	0.6667	0.6708	0.6898	0.6254	0.7028	0.6418	0.6695	0.7345	0.6859	0.6870	**0.7501**(0.0022)
chlorineConcentration	0.5241	0.5282	0.5225	0.5330	0.5316	0.5256	0.5300	0.4111	0.5318	0.5353	0.4997	0.5348	0.5350	**0.5357**(0.0011)
coffee	0.7460	0.8624	**1.0000**	0.5476	**1.0000**	**1.0000**	0.4851	**1.0000**	**1.0000**	0.4841	**1.0000**	0.4921	0.5767	0.9286(0.0016)
diatomsizeReduction	0.9583	0.9583	0.9583	0.9333	0.9137	**1.0000**	0.9583	**1.0000**	0.7083	0.8792	**1.0000**	0.9294	0.7347	0.9682(0.0032)
dist.phal.outl.agegroup	0.6171	0.6531	0.6239	0.6252	0.6539	0.6535	0.6750	0.6020	0.6273	0.7812	0.6650	0.7785	0.7786	**0.7825**(0.0008)
dist.phal.outl.correct	0.5252	0.5362	0.5362	0.5252	0.5327	0.5235	0.5203	0.5252	0.5098	0.5010	0.5962	0.5029	0.5330	**0.6075**(0.0024)
ECG200	0.6315	0.6533	0.6315	0.7018	0.6916	0.6315	0.6018	0.7018	0.5758	0.6018	**0.7285**	0.6422	0.6233	0.6648(0.0034)
ECGFiveDays	0.4783	0.5020	0.5573	0.5020	0.5953	0.5257	0.5573	0.5020	0.5968	0.5016	0.8340	0.5103	0.5114	**0.9638**(0.0032)
GunPoint	0.4971	0.5029	0.5102	0.6498	0.4994	0.4971	0.5420	0.6278	0.6278	0.5400	**0.7257**	0.4981	0.4974	0.6398(0.0011)
Ham	0.5025	0.5219	0.5362	0.5107	0.5127	0.5362	0.5141	0.5311	0.5362	0.5648	**0.6393**	0.5963	0.4956	0.5362(0.0035)
Herring	0.4965	0.5099	0.5164	0.5238	0.5151	0.4940	0.5164	0.4965	0.5417	0.5045	**0.6190**	0.5099	0.5099	0.5759(0.0017)
Lighting2	0.4966	0.5119	0.5373	0.5729	0.5269	0.6263	0.5119	0.6548	0.5192	0.5770	**0.6955**	0.5311	0.5519	0.5913(0.0016)
Meat	0.6595	0.6483	0.6635	0.6578	0.6657	0.6723	0.6816	0.6575	0.6742	0.3220	0.7740	0.6475	0.6220	**0.9763**(0.0016)
Mid.phal.outl.agegroup	0.5351	0.5269	0.5350	0.5315	0.5473	0.5364	0.5513	0.5105	0.5396	0.5757	0.5807	0.7059	0.6800	**0.7982**(0.0028)
Mid.phal.outl.correct	0.5000	0.5431	0.5047	0.5114	0.5149	0.5014	0.5563	0.5114	0.5218	0.5272	**0.6635**	0.5423	0.5423	0.5617(0.0006)
Mid.phal.TW	0.0983	0.1225	0.1919	0.7920	0.8062	0.8187	0.8046	0.6213	0.7920	0.7115	0.7920	0.8590	0.8626	**0.8638**(0.0007)
MoteStrain	0.4947	0.5579	0.6053	0.5579	0.6168	0.6632	0.4789	0.6053	0.4789	0.5062	**0.8105**	0.7435	0.7324	0.7686(0.0036)
OSULeaf	0.5615	0.5372	0.5622	0.5497	0.5665	0.5714	0.5541	0.5538	0.5525	0.7329	0.6551	0.7484	0.7607	**0.7739**(0.0014)
Plane	0.9081	0.8949	0.8954	0.9220	0.9314	0.9603	0.9225	0.9901	**1.0000**	0.9040	**1.0000**	0.9447	0.9447	0.9549(0.0037)
Prox.phal.outl.ageGroup	0.5288	0.4997	0.5463	0.5780	0.5384	0.5305	0.5192	0.5617	0.5206	0.7430	0.7939	0.4263	**0.8091**	**0.8091**(0.0038)
Prox.phal.TW	0.4789	0.4947	0.6053	0.5579	0.5211	0.6053	0.5211	0.5211	0.4789	0.8380	0.7282	0.8189	**0.9030**	0.9023(0.0023)
SonyAIBORobotSurface	0.7721	0.7695	0.7721	0.7787	0.7928	0.7726	0.7988	0.8084	0.7639	0.5563	0.8105	0.5732	0.6900	**0.8769**(0.0033)
SonyAIBORobotSurfaceII	0.8697	0.8745	0.8865	0.8756	0.8948	**0.9039**	0.8684	0.5617	0.8770	0.7012	0.8575	0.6514	0.6572	0.8354(0.0016)
SwedishLeaf	0.4987	0.4923	0.5500	0.5192	0.5038	0.4923	0.5500	0.5333	0.6154	0.8871	0.8547	0.8837	0.8893	**0.9223**(0.0021)
Symbols	0.8810	0.8548	0.8562	0.8525	0.9060	0.8982	**0.9774**	0.8373	0.9603	0.9053	0.9200	0.8841	0.8857	0.9168(0.0022)
ToeSegmentation1	0.4873	0.4921	0.5873	0.5429	0.4968	0.5000	0.6143	0.6143	0.5873	0.5077	**0.6718**	0.4984	0.5017	0.5659(0.0006)
ToeSegmentation2	0.5257	0.5257	0.5968	0.5968	0.5826	0.5257	0.5573	0.5257	0.5020	0.5348	0.6778	0.4991	0.4991	**0.8286**(0.0028)
TwoPatterns	0.8529	0.8259	0.8530	0.8385	**0.8588**	0.8585	0.8446	0.8046	0.7757	0.6251	0.8318	0.6293	0.6338	0.6984(0.0025)
TwoLeadECG	0.5476	0.5495	0.6328	0.8246	0.5635	0.5464	0.5476	0.8246	0.5404	0.5116	**0.8628**	0.5007	0.5016	0.7114(0.0014)
wafer	0.4925	0.4925	0.5263	0.5263	0.4925	0.4925	0.4925	0.4925	0.4925	0.5324	**0.8246**	0.5679	0.5597	0.7338(0.0006)
Wine	0.4984	0.4987	0.5123	0.5021	0.5033	0.5006	0.5064	0.5001	0.5033	0.4906	**0.8985**	0.4913	0.5157	0.6271(0.0039)
WordsSynonyms	0.8775	0.8697	0.8760	0.8861	0.8817	0.8727	0.8159	0.7844	0.8230	0.8855	0.8540	0.8893	0.8947	**0.8984**(0.0003)
AVG Rank	10.6667	9.6806	7.2222	7.3889	6.8750	7.1389	7.9167	8.2361	8.2500	8.8194	3.5000	8.6528	7.5833	**3.0694**
AVG RI	0.5975	0.6077	0.6402	0.6478	0.6542	0.6582	0.6335	0.6419	0.6402	0.6238	0.7676	0.6351	0.6515	**0.7714**
Best	0	0	1	1	1	3	2	0	1	0	12	0	1	**17**
p-value	2.089E-6	4.8823E-6	3.4131E-5	5.7729E-5	4.1222E-5	1.3545E-4	1.2565E-5	1.4814E-4	3.4141E-5	3.0287E-7	9.7386E-1	8.7697E-07	3.2916E-7	-

Experiment

Table 2: Normalized Mutual Information (NMI) comparisons on StarLightCurves

Dataset	YADING	DEC	IDEC	DTC	DTCR
StarLightCurves	0.6000	0.6058	0.6056	0.6072	**0.6731**

Ablation study

Table 3: Rand Index (RI) ablation study results of DTCR

No.	Dataset	w/o K-means	w/o classification	DTCR	No.	Dataset	w/o K-means	w/o classification	DTCR
1	Arrow	0.5980	0.5698	**0.6868**	19	Mid.phal.outl.correct	0.5137	0.5033	**0.5617**
2	Beef	0.7352	0.6497	**0.8046**	20	Mid.phal.TW	0.8625	0.8620	**0.8638**
3	BeetleFly	0.6305	0.6053	**0.9000**	21	MoteStrain	0.7121	0.7239	**0.7686**
4	BirdChicken	0.5600	0.4821	**0.8105**	22	OSULeaf	0.7416	0.7314	**0.7739**
5	Car	0.6610	0.6688	**0.7501**	23	Plane	0.9530	0.9409	**0.9549**
6	chlorineConcentration	0.5341	0.5004	**0.5357**	24	Prox.phal.outl.ageGroup	0.8004	0.7922	**0.8091**
7	coffee	0.6672	0.5434	**0.9286**	25	Prox.phal.TW	0.8549	0.8359	**0.9023**
8	diatomsizeReduction	0.8892	0.7851	**0.9682**	26	SonyAIBORobotSurface	0.7561	0.7702	**0.8769**
9	dist.phal.outl.agegroup	0.7775	0.7780	**0.7825**	27	SonyAIBORobotSurfaceII	0.7069	0.6332	**0.8354**
10	dist.phal.outl.correct	0.5056	0.5051	**0.6075**	28	SwedishLeaf	0.9107	0.9047	**0.9223**
11	ECG200	0.6064	0.5412	**0.6648**	29	Symbols	0.8989	0.9043	**0.9168**
12	ECGFiveDays	0.6970	0.5623	**0.9638**	30	ToeSegmentation1	0.5598	0.4993	**0.5659**
13	GunPoint	0.5589	0.4969	**0.6398**	31	ToeSegmentation2	0.6878	0.6012	**0.8286**
14	Ham	0.5330	0.5040	**0.5362**	32	TwoPatterns	0.6537	0.6650	**0.6984**
15	Herring	0.5173	0.4967	**0.5759**	33	TwoLeadECG	0.5316	0.5262	**0.7114**
16	Lighting2	0.5626	0.5554	**0.5913**	34	wafer	0.5900	0.5322	**0.7338**
17	Meat	0.8245	0.7181	**0.9763**	35	Wine	0.5642	0.5159	**0.6271**
18	Mid.phal.outl.agegroup	0.7981	0.7923	**0.7982**	36	WordsSynonyms	0.8920	0.8891	**0.8984**

Visualization study-Contribution of Each Loss

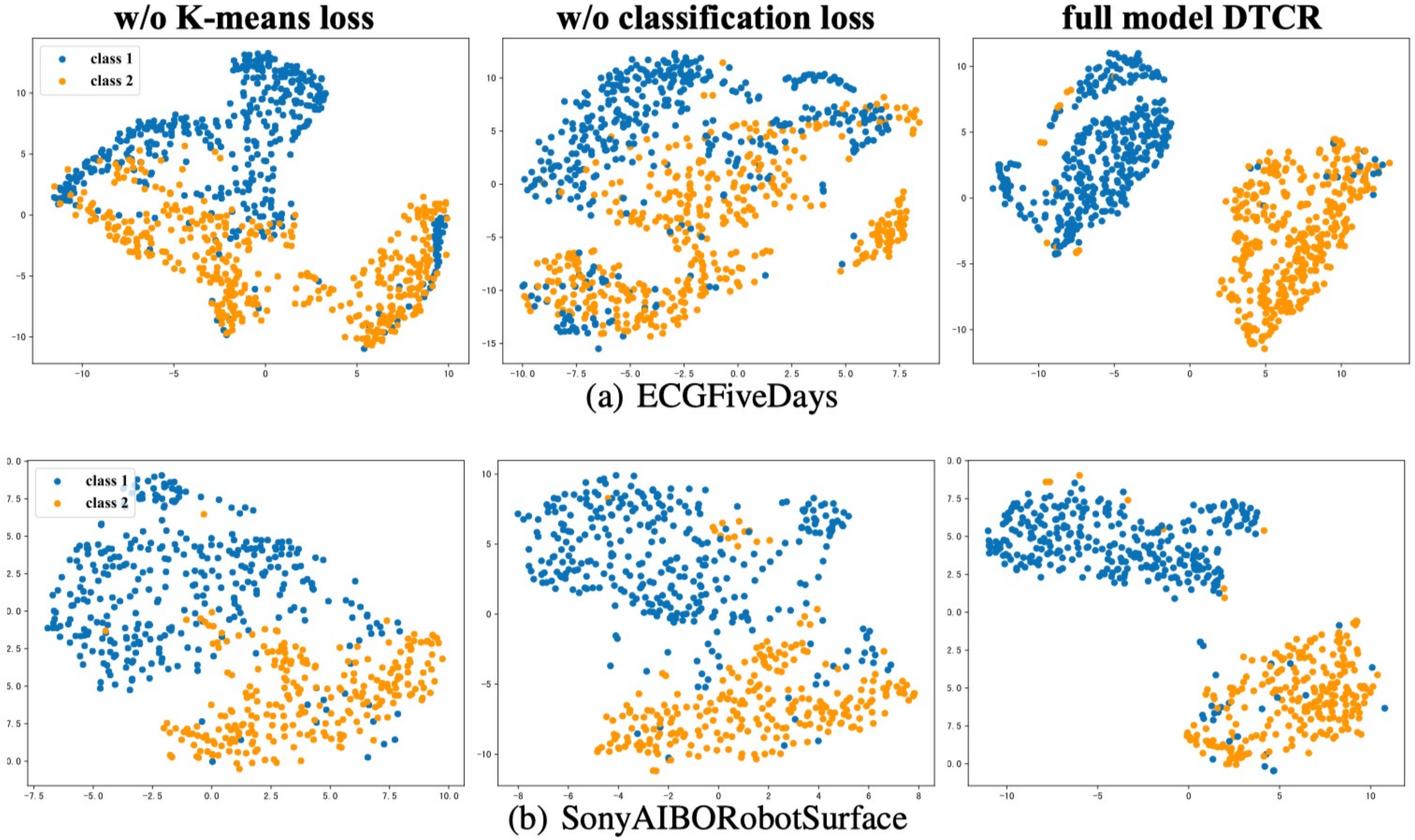


Figure 2: The visualizations with t-SNE on the datasets (a) *ECGFiveDays* and (b) *SonyAIBORobot-Surface*. The colors of the points indicate the actual labels.

Visualization study- The Process of Learning Representations

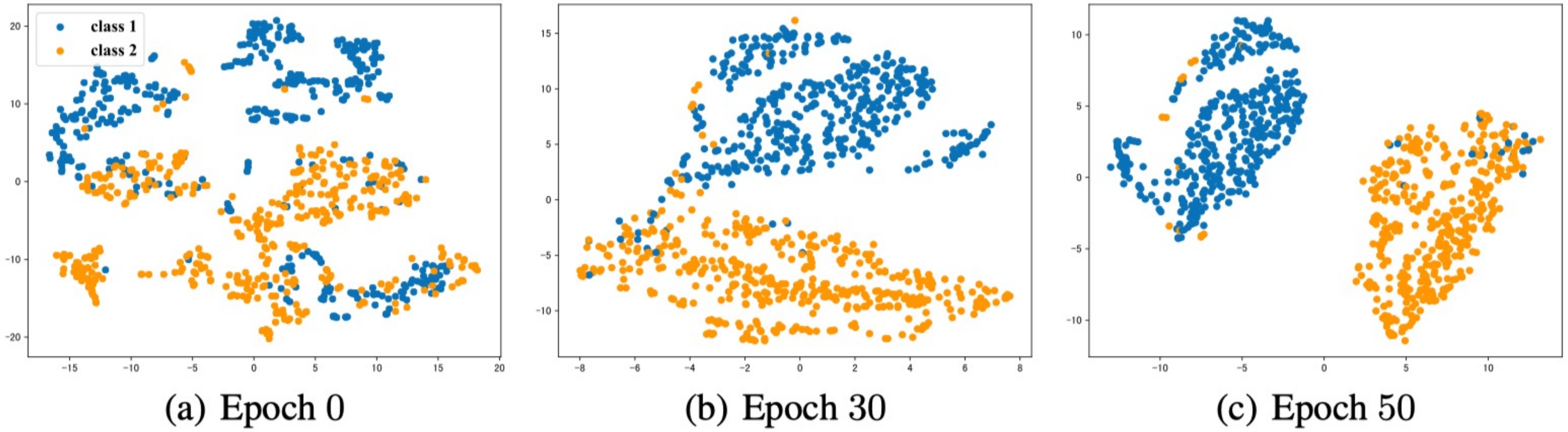


Figure 3: The learned representations on data set *ECGFiveDays* during the training process. From left to the right, the subfigure is obtained at Epoch 0, 30 and 50, respectively.

Visualization study- Robustness Analysis

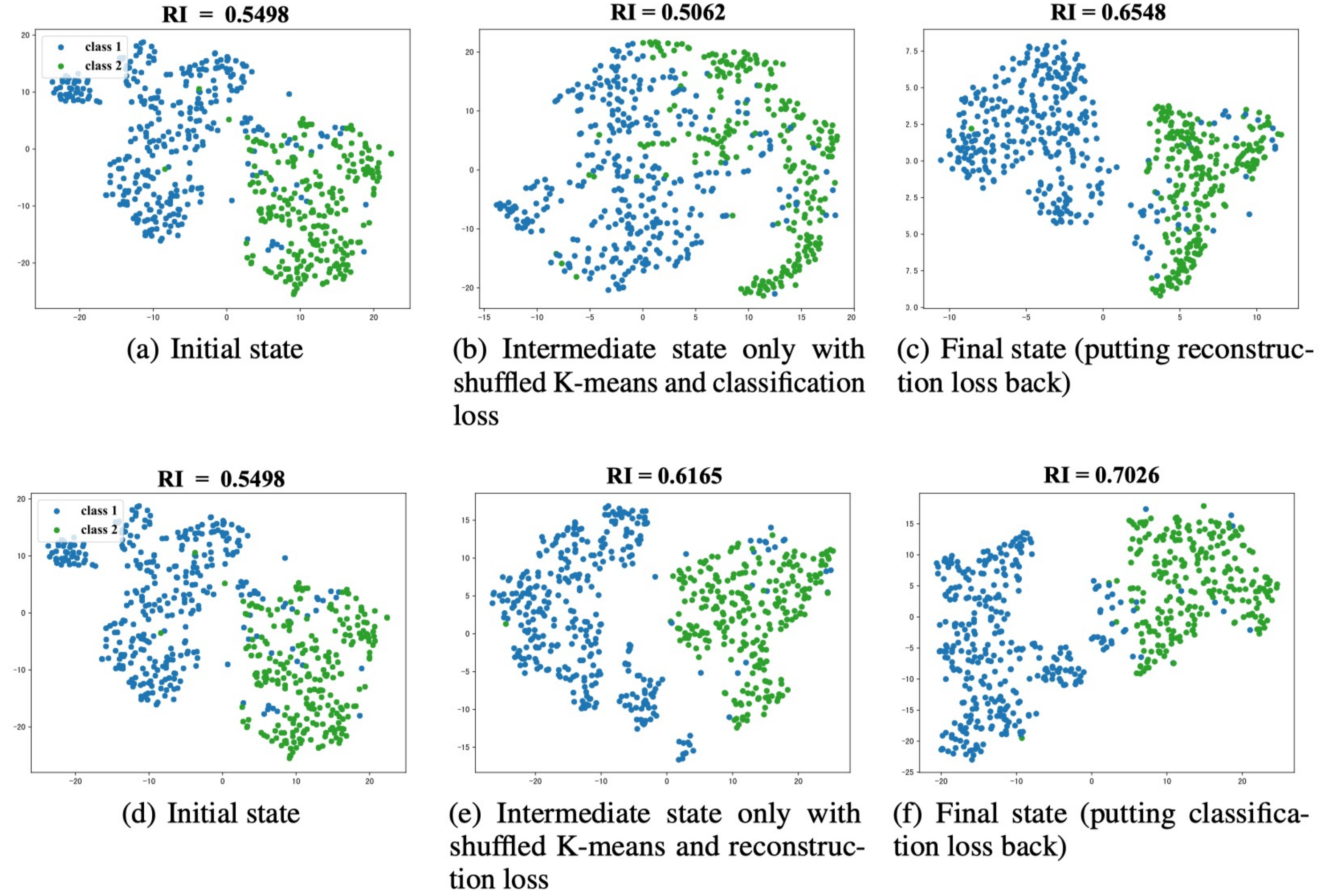


Figure 4: Robustness Analysis of DTCR on *SonyAIBORobotSurface*. Note that the (d) is the same as (a), replicated here for better illustration; hence the first and second rows start with the same state.

Conclusion

- Propose a novel model called Deep Temporal Clustering Representation (DTCR)
- Integrate the temporal reconstruction and K-means objective into the seq2seq model
- A fake-sample generation strategy for time series and auxiliary classification task are proposed to enhance the ability of the encoder.
- Experiment shows the superiority in clustering task , the contribution of each component and the robustness when K-means failed