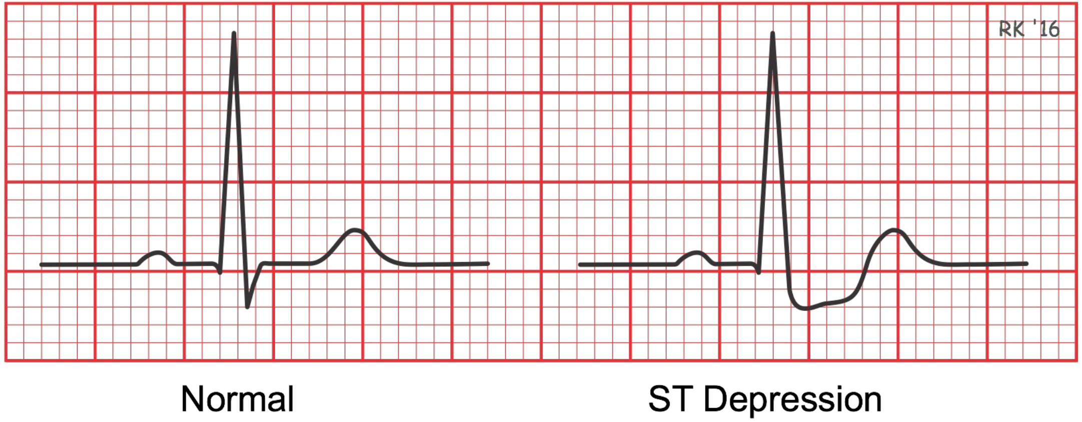
Towards Similarity-Aware Time-Series Classification

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, Xia Hu

Rice University

Time-Series Classification (TSC) Problem

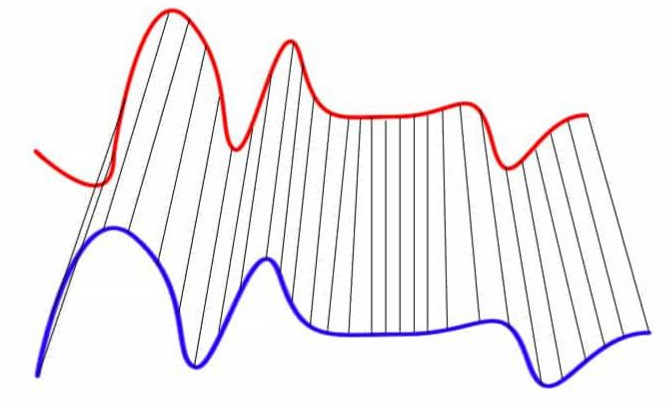
- **Problem setting**
  - Given a collection of time-series with the attached labels, TSC aims to train a classifier to classify unseen time-series.
- **Main challenge**
  - How to model and incorporate the temporal information in the classification?



Source: https://www.cvphysiology.com/uploads/images/CAD012%20ECG%20ST%20depression.png

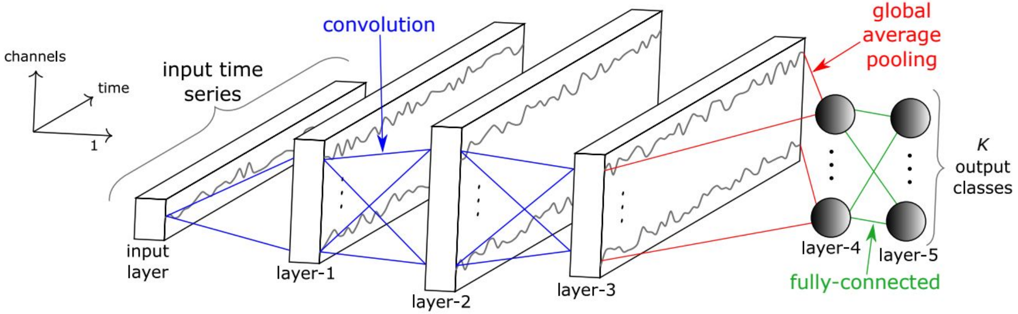
Existing Solutions

- **Existing studies approach TSC in two major directions**
  - **Similarity-based:** Combine a k-NN classifier with a similarity measure for classification.
  - **Deep learning:** Perform end-to-end training on the raw time-series and learn the representations to do classification.



Dynamic Time Warping Matching

Source: https://commons.wikimedia.org/wiki/File:Euclidean_vs_DTW.jpg

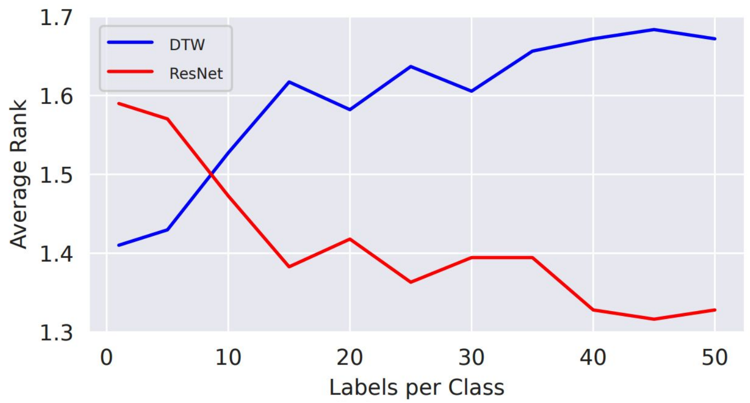


ResNet with 1-D convolution

Source: https://arxiv.org/pdf/1809.04356.pdf

Motivation

- **Preliminary experiments**
  - We compare **DTW** (a representative similarity-based method) and **ResNet** (a representative deep learning approach) on **the full 128 UCR datasets**. We report the average ranks. The lower the better.



Average ranks of ResNet and DTW on the full 128 UCR datasets, where different numbers of labels per class is given.

Daochen Zha (daochen.zha@rice.edu)

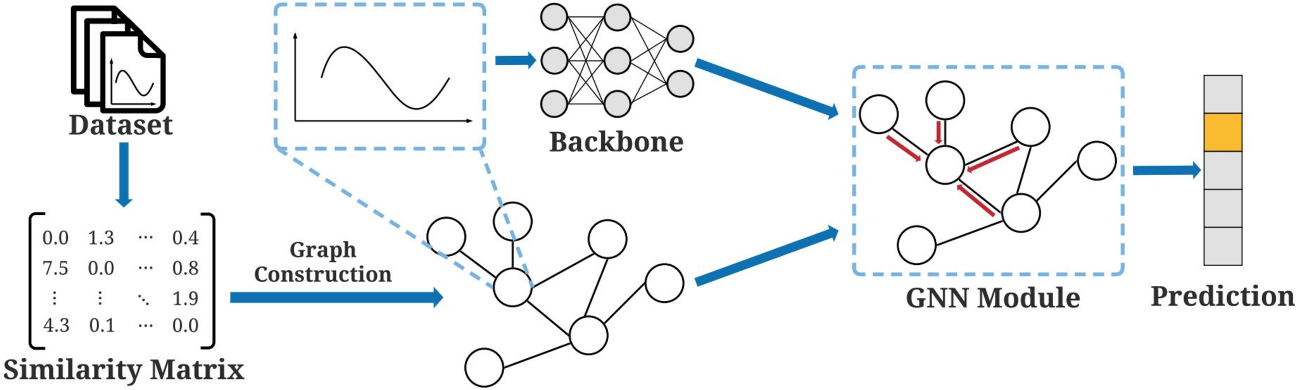
Research Question and Challenges

- **Our research question**
  - Can we connect the two research lines in such a way as to jointly model time-series similarities and learn the representations?

- **Challenges**
  - How can we incorporate similarity information into representation learning?
  - Even though we can enable similarity in deep learning models, how can we balance similarity information and the original representation learning?

SimTSC Framework

- **Our simple yet effective solution**
  - We propose Similarity-Aware Time-Series Classification (SimTSC) framework based on Graph Neural Networks.
  - Time-series -> node
  - Similarity of time-series -> edge
  - TSC -> node classification



Each time-series is first processed by a backbone, and enhanced by GNN with aggregation.

Our Instantiation

- **Backbone**
  - We use **ResNet** as the backbone since it has strong performance.
- **Similarity Measure**
  - We use **DTW** as the the similarity measure because it is the most popular one.
- **Graph Neural Networks**
  - We use **Graph Convolutional Networks (GCN)** because it is the most basic one.
  - We only use 1-layer GCN. We find that it delivers the best performance.
- **Other Tricks**
  - We use **negative sampling** to sample a half batch of labeled time-series and a half batch of unlabeled time-series.

Results on Univariate Time-Series

- **Experimental Setting**
  - We compare SimTSC with the existing **similarity-based** and **deep learning** methods on the **full 128 UCR datasets**. We report average rank and Wilcoxon signed rank test ($p < 0.05$) for the significance test.

	Algorithm \ Labels	1	5	10	15	20	25	30	35	40	45	50
Similarity-based	DTW	3.776	4.163	4.465▲▽	4.738▲▽	4.824†▲▽	5.048†▲▽	4.965†▲▽	5.160†▲▽	5.309†▲▽	5.199†▲▽	5.211†▲▽
Deep Learning	MLP	5.504†▲▽	5.496†▲▽	5.438†▲▽	5.309†▲▽	5.316†▲▽	5.256†▲▽	5.367†▲▽	5.477†▲▽	5.195†▲▽	5.402†▲▽	5.348†▲▽
	FCN	4.630▲▽	4.310	4.383▲▽	4.508▲▽	4.723†▲▽	4.803†▲▽	4.699†▲▽	4.910†▲▽	4.773†▲▽	4.883†▲▽	4.852†▲▽
	ResNet	4.846†▲▽	4.857†▲▽	4.617†▲▽	4.047	4.449▲▽	4.039	4.102	4.090	4.086	3.840	3.895
	InceptionTime	5.484†▲▽	5.302†▲▽	5.438†▲▽	5.434†▲▽	5.215†▲▽	5.145†▲▽	5.168†▲▽	4.914†▲▽	4.941†▲▽	5.066†▲▽	5.039†▲▽
Ours	SimTSC-S	4.224▲	4.278▲▽	4.074	4.277▲▽	4.141▲▽	4.044▽	4.148▲	3.988	3.887	3.918	4.047
	SimTSC-I	**3.724**	3.817	3.793	**3.836**	3.746	4.031▲▽	**3.762**	3.734	**3.852**	3.867	**3.797**
	SimTSC-T	3.811	**3.778**	**3.781**	3.852	**3.586**	**3.632**	3.789	**3.727**	3.957	**3.824**	3.812

Results on Multivariate Time-Series

- **Experimental Setting**
  - We conduct experiments on 4 multivariate time-series classification tasks, including **Character Trajectories**, **ECG**, **KickVsPunch**, and **NetFlow**.

Similarity-based

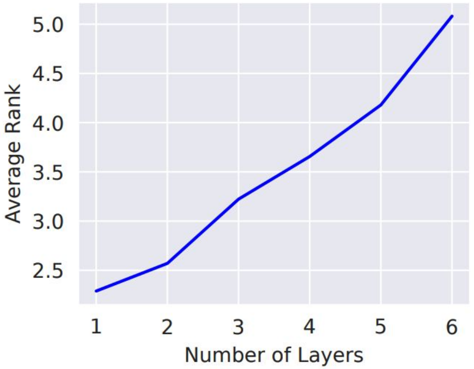
Deep Learning

Ours

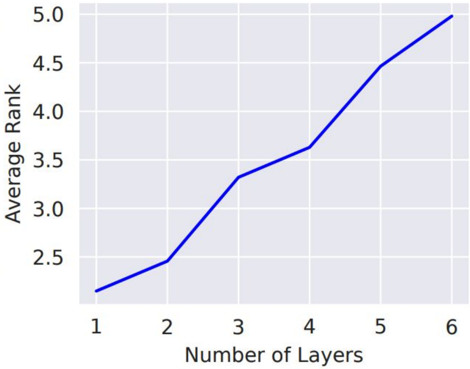
Dataset	Labels / Algorithm	5	10	15	20	25	30	35	40	45	50
Character Trajectories	DTW	.847±.014	.881±.005	.895±.009	.900±.014	.908±.014	.907±.013	.906±.010	.906±.007	.909±.010	.913±.008
	ResNet	.834±.024	.898±.017	.920±.010	.937±.010	.939±.008	.941±.009	.949±.008	.956±.011	.958±.007	.958±.007
	InceptionTime	.883±.010	.939±.007	.947±.006	**.968±.006**	.964±.003	.974±.005	.979±.003	.978±.005	.979±.001	**.986±.003**
	TapNet	-	-	-	-	-	-	-	-	-	-
	SimTSC-S	.894±.020	.939±.009	.949±.007	.947±.017	.964±.011	.975±.003	.977±.011	.975±.004	.981±.007	.982±.005
	SimTSC-I	**.914±.012**	.944±.009	.951±.015	.953±.012	**.969±.011**	**.978±.006**	**.981±.007**	.979±.005	.977±.008	.980±.003
	SimTSC-T	.903±.014	**.946±.005**	**.957±.011**	.964±.009	.967±.012	.973±.009	.976±.009	**.981±.006**	**.983±.008**	**.986±.004**
ECG	DTW	.605±.124	.670±.086	.740±.112	.755±.103	.805±.043	.825±.050	.805±.053	.800±.057	.805±.053	.800±.057
	ResNet	.745±.048	.795±.037	.805±.058	.800±.079	**.860±.030**	**.855±.048**	.850±.052	**.855±.029**	.830±.037	**.870±.029**
	InceptionTime	.750±.045	.805±.033	.785±.020	.800±.037	.820±.037	.830±.043	.825±.016	.850±.027	**.855±.0010**	.850±.016
	TapNet	.770±.043	.780±.012	.755±.025	.795±.048	.810±.037	.795±.029	.785±.025	.815±.037	.830±.019	.845±.024
	SimTSC-S	.795±.043	.810±.020	**.855±.040**	**.840±.051**	.830±.056	.840±.020	**.860±.041**	.825±.047	.830±.071	.860±.025
	SimTSC-I	.790±.062	.765±.072	.830±.070	.730±.159	.740±.087	.800±.091	.830±.048	.750±.052	.790±.108	.735±.108
	SimTSC-T	**.810±.041**	**.815±.046**	.770±.108	.815±.115	.730±.118	.745±.075	.745±.099	.780±.051	.775±.071	.710±.101
KickvsPunch	DTW	.433±.082	.433±.082	.433±.082	-	-	-	-	-	-	-
	ResNet	.667±.183	**.833±.149**	.833±.183	-	-	-	-	-	-	-
	InceptionTime	.667±.000	.533±.125	.567±.226	-	-	-	-	-	-	-
	TapNet	.700±.125	.767±.082	.733±.013	-	-	-	-	-	-	-
	SimTSC-S	**.733±.200**	.767±.133	**.867±.125**	-	-	-	-	-	-	-
	SimTSC-I	.700±.125	**.833±.105**	.800±.125	-	-	-	-	-	-	-
	SimTSC-T	.600±.133	.767±.133	.767±.082	-	-	-	-	-	-	-
NetFlow	DTW	.611±.016	.559±.128	.607±.132	.595±.118	.546±.103	.568±.125	.523±.154	.481±.203	.503±.217	.504±.214
	ResNet	.613±.074	.714±.063	.749±.022	.763±.038	.739±.058	.767±.050	.769±.054	.767±.049	.787±.026	.797±.039
	InceptionTime	.418±.052	.456±.046	.484±.058	.618±.049	.642±.036	.657±.024	.678±.014	.675±.036	.681±.018	.681±.015
	TapNet	-	-	-	-	-	-	-	-	-	-
	SimTSC-S	.519±.108	.720±.071	.705±.055	.709±.089	.738±.082	.786±.036	.765±.091	.790±.045	.784±.063	.799±.047
	SimTSC-I	.766±.043	.788±.036	.689±.139	**.776±.042**	.731±.084	.755±.104	**.834±.037**	.798±.066	.810±.065	.839±.035
	SimTSC-T	**.769±.052**	**.805±.035**	**.785±.101**	.766±.095	**.745±.092**	**.825±.029**	.801±.065	**.827±.059**	**.847±.023**	**.852±.028**

Impact of the Number of GCN Layers

- **Experimental Setting**
  - We vary the number of GCN layers when we have 10 or 20 labels



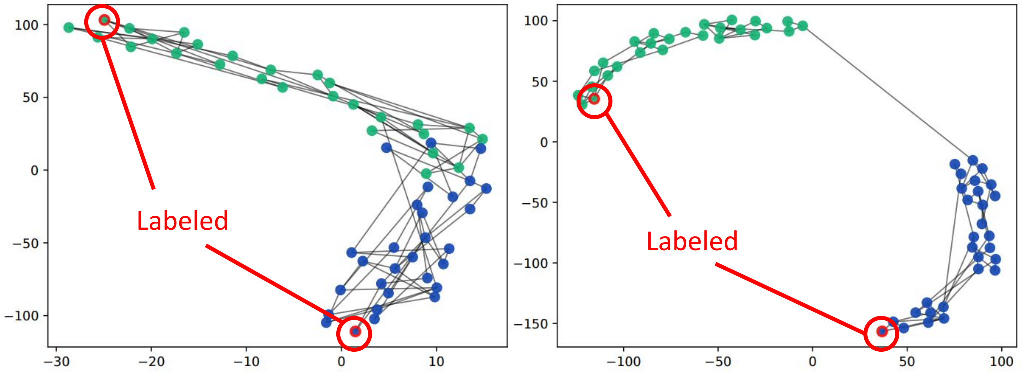
10 Labels



20 Labels

- **Observation**
  - One GCN layer achieves the best performance.
  - Possibly because the graph is dense and more layers lead to over-smoothing.

Visualization



(a) ResNet (83% accuracy) (b) SimTSC (100% accuracy)

Learned representations of ResNet and SimTSC on Coffee with 56 time-series. The two classes are marked in blue and green. Only one time-series is labeled.

Summary and Q & A

- **Takeaways**
  - SimTSC is a conceptually simple yet effective framework to join the research efforts of similarity-based and deep learning methods for time-series classification.
  - We demonstrated the effectiveness of graph neural networks in time-series classification.
- **Future Work**
  - Larger dataset, sparse graph, other tasks in time-series.
  - Differentiable similarity learning.
- **Acknowledgement**

  - This work is supported by National Science Foundation (NSF).

Paper   Code