BRL Seminar

(2023. 01. 31. Tue)

Self-Supervised Learning with Time Series Data (2)

통합과정 5학기 이승한

Papers

1. Self-supervised Time Series Representation Learning by Inter-Intra Relational Reasoning (2020)
2. Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification (2022)

SELF-SUPERVISED TIME SERIES REPRESENTATION LEARNING BY INTER-INTRA RELATIONAL REASONING

Haoyi Fan, Fengbin Zhang
School of Computer Science and Technology
Harbin University of Science and Technology
Harbin 150080, China
isfanhy@hrbust.edu.cn

Yue Gao
School of Software
Tsinghua University
Beijing 100084, China
kevin.gaoy@gmail.com

ABSTRACT

Self-supervised learning achieves superior performance in many domains by extracting useful representations from the unlabeled data. However, most of traditional self-supervised methods mainly focus on exploring the inter-sample structure while less efforts have been concentrated on the underlying intra-temporal structure, which is important for time series data. In this paper, we present **Self-Time**: a general **Self**-supervised **Time** series representation learning framework, by exploring the inter-sample relation and intra-temporal relation of time series to learn the underlying structure feature on the unlabeled time series. Specifically, we first generate the inter-sample relation by sampling positive and negative sam-

https://arxiv.org/pdf/2011.13548.pdf

Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li and Cunta
Guan *Fellow, IEEE*

Abstract—Learning time-series representations when only unlabeled data or few labeled samples are available can be a challenging task. Recently, contrastive self-supervised learning has shown great improvement in extracting useful representations from unlabeled data via contrasting different augmented views of data. In this work, we propose a novel **T**ime-**S**eries representation learning framework via **T**emporal and **C**ontextual **C**ontrasting (**TS-TCC**) that learns representations from unlabeled data with contrastive learning. Specifically, we propose time-series-specific weak and strong augmentations and use their views to learn robust temporal relations in the proposed temporal contrasting module, besides learning discriminative representations by our proposed contextual contrasting module. Additionally, we conduct a systematic study of time-series data augmentation selection, which is a key part of contrastive learning. We also extend TS-TCC to the semi-supervised learning settings and propose a **C**lass-**A**ware TS-**TCC** (**CA-TCC**) that benefits from the available few labeled data to further improve representations learned by TS-TCC. Specifically, we leverage the robust pseudo labels produced by TS-TCC to realize a class-aware contrastive loss. Extensive experiments show that the linear evaluation of the features learned by our proposed framework performs comparably with the fully supervised training. Additionally, our framework shows high efficiency in few labeled data and transfer learning scenarios. The code is publicly available at https://github.com/emadeldeen24/CA-TCC.

Index Terms—self-supervised learning, semi-supervised learning, time-series classification, temporal contrasting, contextual contrasting, augmentation.

https://arxiv.org/pdf/2208.06616.pdf

Papers

1. <u>Self-supervised Time Series Representation Learning by Inter-Intra Relational Reasoning (2020)</u>
2. Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification (2022)

SELF-SUPERVISED TIME SERIES REPRESENTATION LEARNING BY INTER-INTRA RELATIONAL REASONING

Haoyi Fan, Fengbin Zhang
School of Computer Science and Technology
Harbin University of Science and Technology
Harbin 150080, China
isfanhy@hrbust.edu.cn

Yue Gao
School of Software
Tsinghua University
Beijing 100084, China
kevin.gaoy@gmail.com

ABSTRACT

Self-supervised learning achieves superior performance in many domains by extracting useful representations from the unlabeled data. However, most of traditional self-supervised methods mainly focus on exploring the inter-sample structure while less efforts have been concentrated on the underlying intra-temporal structure, which is important for time series data. In this paper, we present **Self-Time**: a general **Self**-supervised **Time** series representation learning framework, by exploring the inter-sample relation and intra-temporal relation of time series to learn the underlying structure feature on the unlabeled time series. Specifically, we first generate the inter-sample relation by sampling positive and negative sam-

https://arxiv.org/pdf/2011.13548.pdf

Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li and Cuntai Guan *Fellow, IEEE*

Abstract—Learning time-series representations when only unlabeled data or few labeled samples are available can be a challenging task. Recently, contrastive self-supervised learning has shown great improvement in extracting useful representations from unlabeled data via contrasting different augmented views of data. In this work, we propose a novel **T**ime-**S**eries representation learning framework via **T**emporal and **C**ontextual **C**ontrasting (**TS-TCC**) that learns representations from unlabeled data with contrastive learning. Specifically, we propose time-series-specific weak and strong augmentations and use their views to learn robust temporal relations in the proposed temporal contrasting module, besides learning discriminative representations by our proposed contextual contrasting module. Additionally, we conduct a systematic study of time-series data augmentation selection, which is a key part of contrastive learning. We also extend TS-TCC to the semi-supervised learning settings and propose a **C**lass-**A**ware TS-**TCC** (**CA-TCC**) that benefits from the available few labeled data to further improve representations learned by TS-TCC. Specifically, we leverage the robust pseudo labels produced by TS-TCC to realize a class-aware contrastive loss. Extensive experiments show that the linear evaluation of the features learned by our proposed framework performs comparably with the fully supervised training. Additionally, our framework shows high efficiency in few labeled data and transfer learning scenarios. The code is publicly available at https://github.com/emadeldeen24/CA-TCC.

Index Terms—self-supervised learning, semi-supervised learning, time-series classification, temporal contrasting, contextual contrasting, augmentation.

https://arxiv.org/pdf/2208.06616.pdf

Contents

1. Preliminaries

a) Self-supervised Learning (SSL)

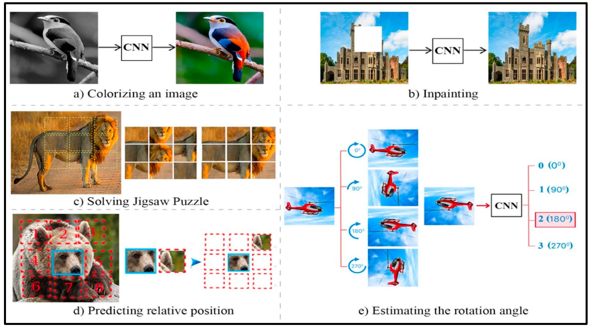
b) Self-supervised Learning (SSL) in Time Series

c) Relational Reasoning

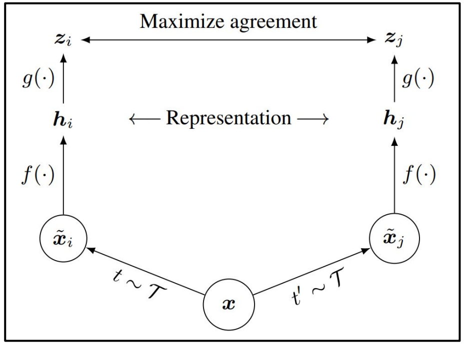
1. Preliminaries

a) Self-supervised Learning (SSL)

- capture the most informative properties from **unlabeled** data, using **self-generated supervisory signal**
- SSL in Computer Vision
  - (1) **pretext tasks** (Jigsaw puzzles (2016), Inpainting (2016), Rotation prediction (2018) ···)
  - (2) **contrastive learning**



(1) pretext tasks https://www.mdpi.com/1099-4300/24/4/551



(2) contrastive learning https://arxiv.org/abs/2002.05709.pdf

1. Preliminaries

b) SSL in Time Series

- triplet loss (2019) , contrastive loss for TS (2020)
- SSL methods that predict handcrafted features (2019, 2020)
- signal transformations (2019, 2020)

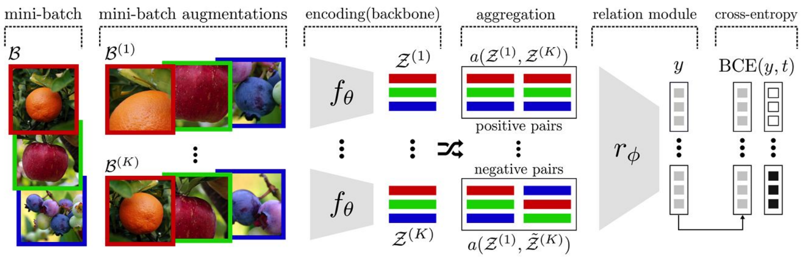
Limitations of previous SSL in Time Series :

few of those works consider the intra-temporal structure of TS

1. Preliminaries

c) Relational Reasoning

- **Reasoning the relations between entities and their properties**
- Used in various domains
  - knowledge base (Socher et al., 2013), question answering (Johnson et al., 2017; Santoro et al., 2017), video action recognition (Zhou et al., 2018), reinforcement learning (Zambaldi et al., 2019), graph representation (Battaglia et al., 2018),
  - most of them attempt to learn a relation reasoning head for a special task
- **inter-sample relation reasoning based on UNLABELED image data** (Patacchiola & Storkey, 2020)



https://arxiv.org/abs/2006.05849

2. Methodology

propose SelfTime (general Self-supervised Time series representation learning framework)

- focus on TS data by exploring both …
  - (1) inter-sample relation
  - (2) intra-temporal relation

  for TS representation in a self-supervised manner

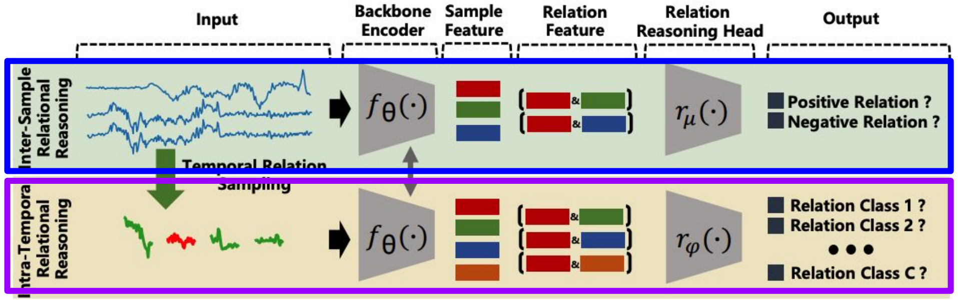


Figure 2: Architecture of SelfTime.

2. Methodology

Notation

Unlabeled TS : $\mathcal{T} = \{\boldsymbol{t}_n\}_{n=1}^{N}$

- each TS : $\boldsymbol{t}_n = (t_{n,1}, \ldots t_{n,T})^{\mathrm{T}}$

Goal : learn a useful representation $z_n = f_\theta(\boldsymbol{t}_n)$

- from the backbone encoder $f_\theta(\cdot)$

2. Methodology

Overview

Input : original TS & sampled time pieces

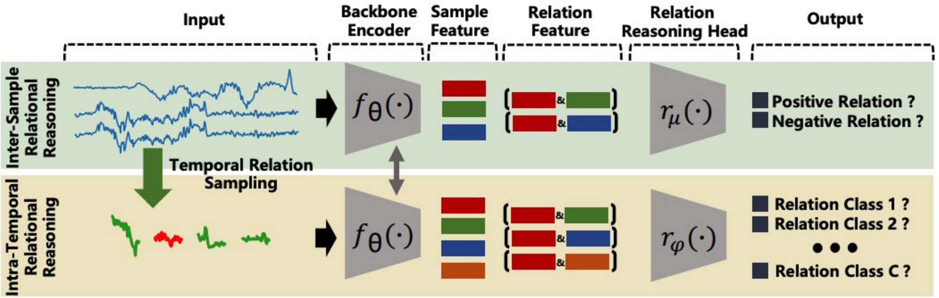


Figure 2: Architecture of SelfTime.

2. Methodology

Overview

Input : original TS & sampled time pieces

Feature Extraction

- Extracts TS feature & Time Piece feature
- to aggregate the inter-sample relation feature and intra-temporal relation feature
- by backbone encoder $f_\theta(\cdot)$

4-layer 1D CNN with ReLU + BN

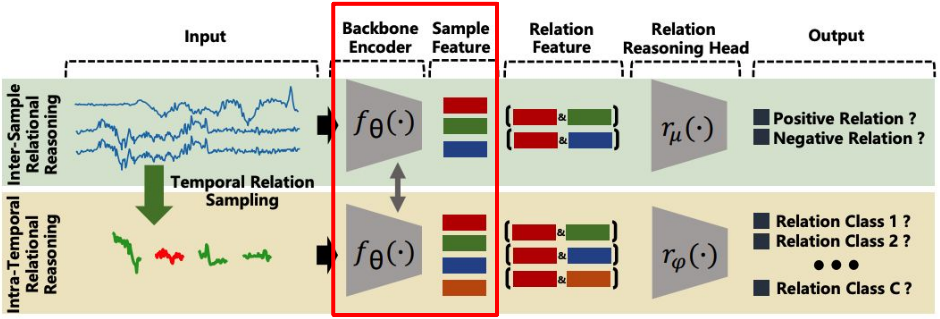


Figure 2: Architecture of SelfTime.

2. Methodology

Overview

Input : original TS & sampled time pieces

Feature Extraction

- Extracts TS feature & Time Piece feature
- to aggregate the inter-sample relation feature and intra-temporal relation feature
- by backbone encoder $f_\theta(\cdot)$

Relation Reasoning

- feed to 2 separate relation reasoning heads $r_\mu(\cdot)$ and $r_\varphi(\cdot)$
- to reason the final relation score of inter-sample & intra-temporal relation.

4-layer 1D CNN with ReLU + BN

2-layer FC layers (dim = 256)

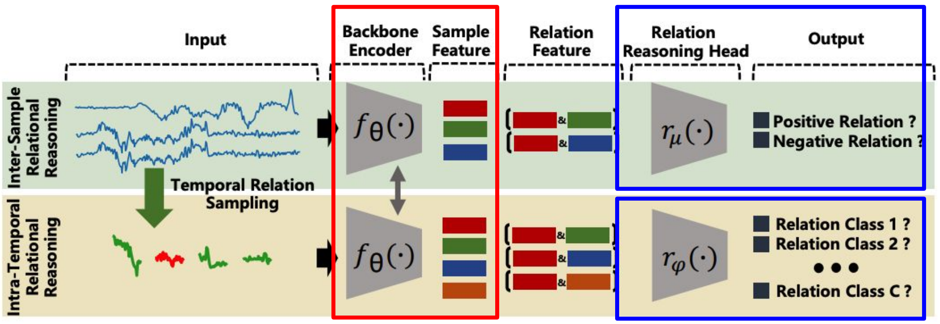
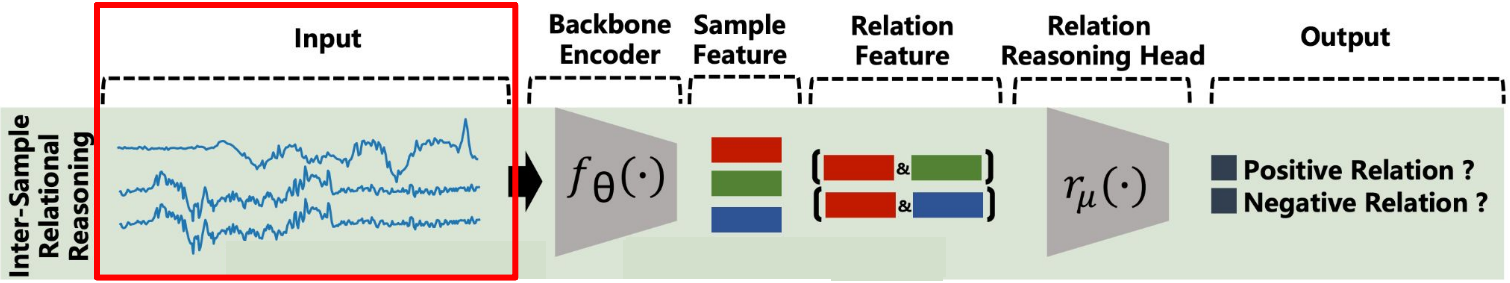


Figure 2: Architecture of SelfTime.

2. Methodology

(1) Inter-Sample Relational Reasoning



Step 1) 2 sets of K augmentations

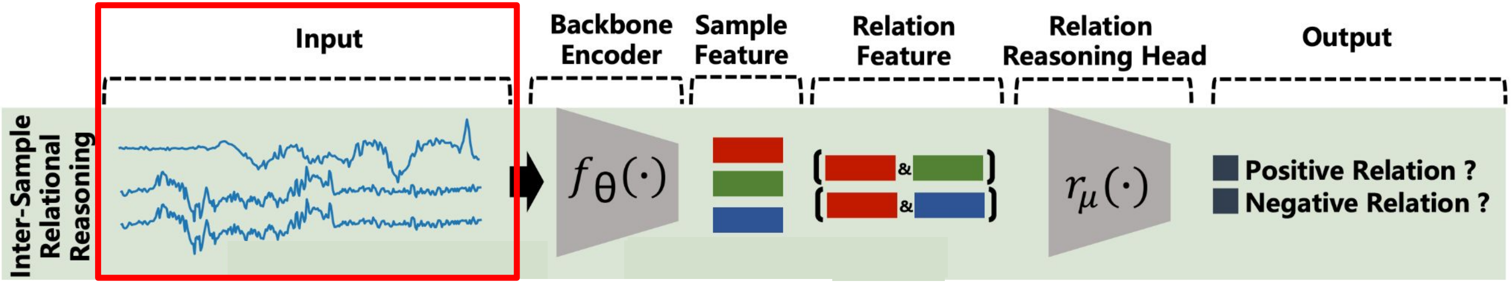
- with 2 different TS samples (t_m & t_n)
- Augmented samples :

$$\mathcal{A}(\boldsymbol{t}_m) = \left\{\boldsymbol{t}_m^{(i)}\right\}_{i=1}^{K}.$$

$$\mathcal{A}(\boldsymbol{t}_n) = \left\{\boldsymbol{t}_n^{(i)}\right\}_{i=1}^{K}.$$

2. Methodology

(1) Inter-Sample Relational Reasoning



Step 1) 2 sets of K augmentations

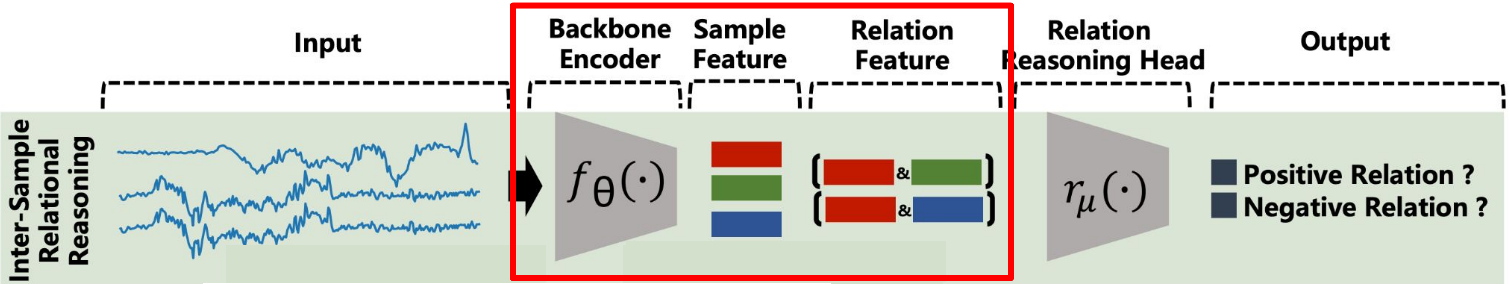
- with 2 different TS samples (t_m & t_n)
- Augmented samples :
  - $\mathcal{A}(\boldsymbol{t}_m) = \left\{\boldsymbol{t}_m^{(i)}\right\}_{i=1}^{K}$.
  - $\mathcal{A}(\boldsymbol{t}_n) = \left\{\boldsymbol{t}_n^{(i)}\right\}_{i=1}^{K}$.

Step 2) construct 2 types of relation pairs (let m : anchor)

- (pair 1) positive relation pairs
  - $\left(\boldsymbol{t}_m^{(i)}, \boldsymbol{t}_m^{(j)}\right)$ sampled from same augmentation set $\mathcal{A}(\boldsymbol{t}_m)$
- (pair 2) negative relation pairs
  - $\left(\boldsymbol{t}_m^{(i)}, \boldsymbol{t}_n^{(j)}\right)$ sampled from different augmentation sets $\mathcal{A}(\boldsymbol{t}_m)$ and $\mathcal{A}(\boldsymbol{t}_n)$

2. Methodology

(1) Inter-Sample Relational Reasoning



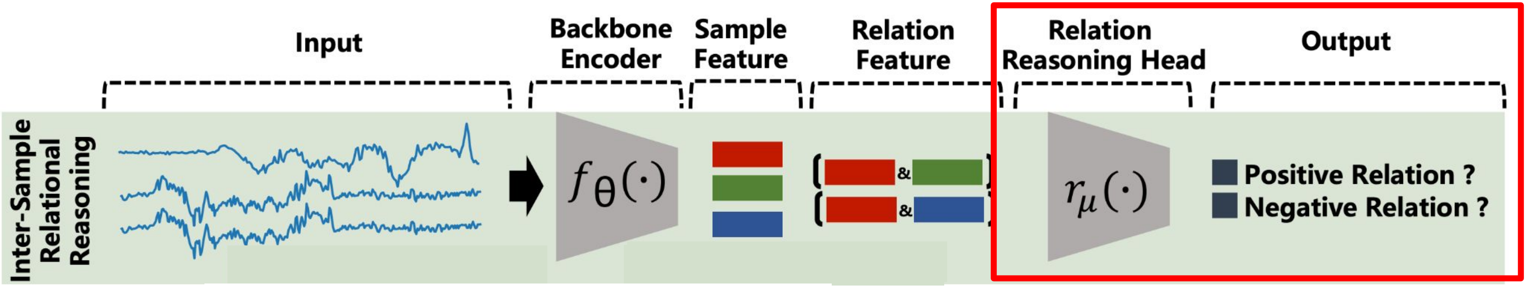
Step 3) Learn the relation representation

- based on the sampled relation pairs
- use the backbone encoder f_θ

- step 3-1) extract sample representations
  - $\boldsymbol{z}_m^{(i)} = f_\theta\left(\boldsymbol{t}_m^{(i)}\right)$ & $\boldsymbol{z}_m^{(j)} = f_\theta\left(\boldsymbol{t}_m^{(j)}\right)$.
  - $\boldsymbol{z}_n^{(j)} = f_\theta\left(\boldsymbol{t}_n^{(j)}\right)$.
- step 3-2) construct pos & neg relation representations
  - [pos] $\left[\boldsymbol{z}_m^{(i)}, \boldsymbol{z}_m^{(j)}\right]$
  - [neg] $\left[\boldsymbol{z}_m^{(i)}, \boldsymbol{z}_n^{(j)}\right]$

2. Methodology

(1) Inter-Sample Relational Reasoning

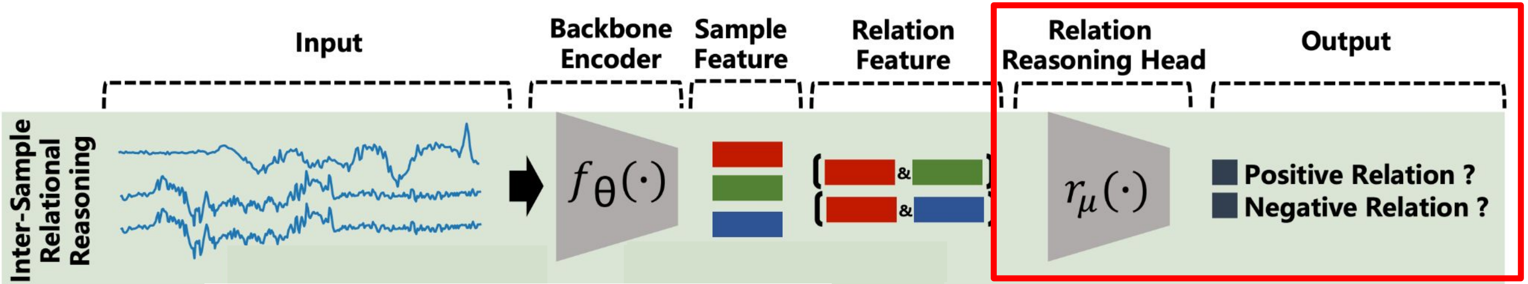


Step 4) Relation reasoning

- input : pos & neg relation representations
- model : inter-sample relation reasoning head $r_\mu(\cdot)$
- output : final relation score
  - for pos : $h_{2m-1}^{(i,j)} = r_\mu\left(\left[\boldsymbol{z}_m^{(i)}, \boldsymbol{z}_m^{(j)}\right]\right)$
  - for neg : $h_{2m}^{(i,j)} = r_\mu\left(\left[z_m^{(i)}, z_n^{(j)}\right]\right)$

2. Methodology

(1) Inter-Sample Relational Reasoning



Step 4) Relation reasoning

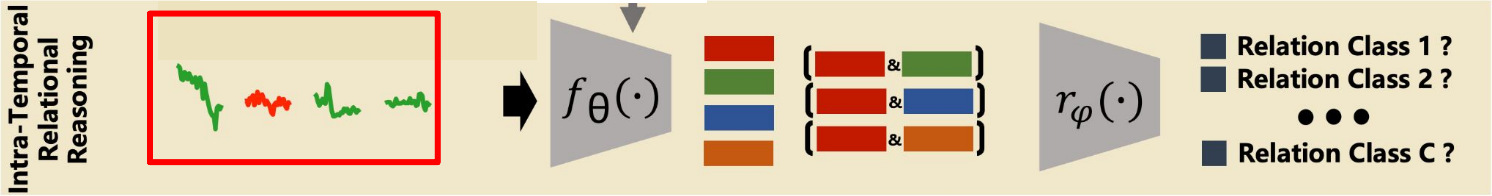
- input : pos & neg relation representations
- model : inter-sample relation reasoning head $r_\mu(\cdot)$
- output : final relation score
  - for pos : $h_{2m-1}^{(i,j)} = r_\mu\left(\left[\boldsymbol{z}_m^{(i)}, \boldsymbol{z}_m^{(j)}\right]\right)$
  - for neg : $h_{2m}^{(i,j)} = r_\mu\left(\left[z_m^{(i)}, z_n^{(j)}\right]\right)$

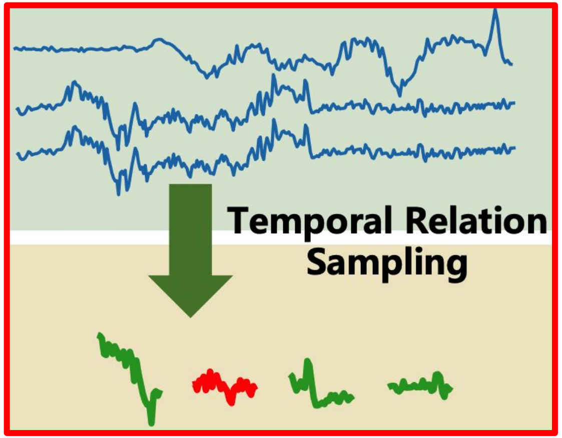
Step 5) inter-sample relation reasoning task

- binary classification task
- loss : BCE loss
  - $\mathcal{L}_{\text{inter}} = -\sum_{n=1}^{2N}\sum_{i=1}^{K}\sum_{j=1}^{K}\left(y_n^{(i,j)} \cdot \log\left(h_n^{(i,j)}\right) + \left(1 - y_n^{(i,j)}\right) \cdot \log\left(1 - h_n^{(i,j)}\right)\right)$
    - $y_n^{(i,j)} = 1$ for pos relation
    - $y_n^{(i,j)} = 0$ for neg relation

2. Methodology

(2) Intra-Sample Relational Reasoning





Notation

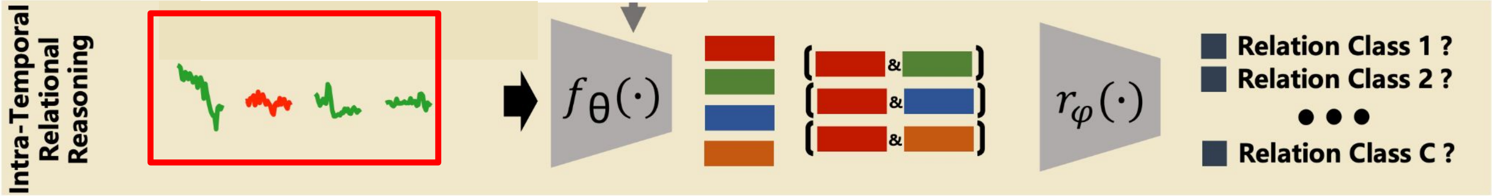
input TS sample : $\boldsymbol{t}_n = (t_{n,1}, \ldots t_{n,T})^{\mathrm{T}}$

L - length time piece of $\boldsymbol{t}_n$: $\boldsymbol{p}_{n,u}$

- $\boldsymbol{p}_{n,u} = (t_{n,u}, t_{n,u+1}, \ldots, t_{n,u+L-1})^{\mathrm{T}}$

2. Methodology

(2) Intra-Sample Relational Reasoning

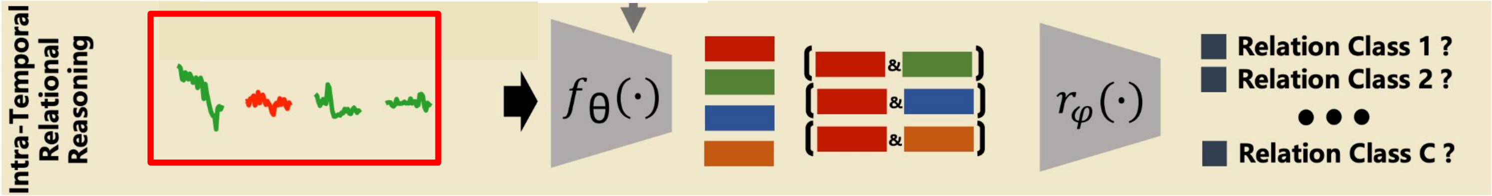


Step 1) sample different types of temporal relation

- step 1-1) sample two $\boxed{\boldsymbol{p}_{n,u} \ \& \ \boldsymbol{p}_{n,v}}$
- step 1-2) assign temporal relation between $\boldsymbol{p}_{n,u}$ and $\boldsymbol{p}_{n,v}$
  - based on $\boxed{\text{temporal distance } d_{u,v}}$ (where $d_{u,v} = | u - v |$)

2. Methodology

(2) Intra-Sample Relational Reasoning



step 2) define C types of temporal relations

- for each pair of pieces based on their temporal distance
- step 2-1) set a distance threshold as $D = \lfloor T/C \rfloor$,
- step 2-2) if the distance $d_{u,v}$ of a piece pair is ....
  - $0 \sim D \rightarrow$ assign relation label as 0
  - $D \sim 2D \rightarrow$ assign relation label as 1
  - ...
  - $CD \sim (C+1)D \rightarrow$ assign relation label as C

2. Methodology

(2) Intra-Sample Relational Reasoning

step 2) define C types of temporal relations

- for each pair of pieces based on their temporal distanc
- step 2-1) set a distance threshold as $D = \lfloor T/C \rfloor$,

pair is ....

)

s 1

n label as C

Algorithm 1: Temporal Relation Sampling.

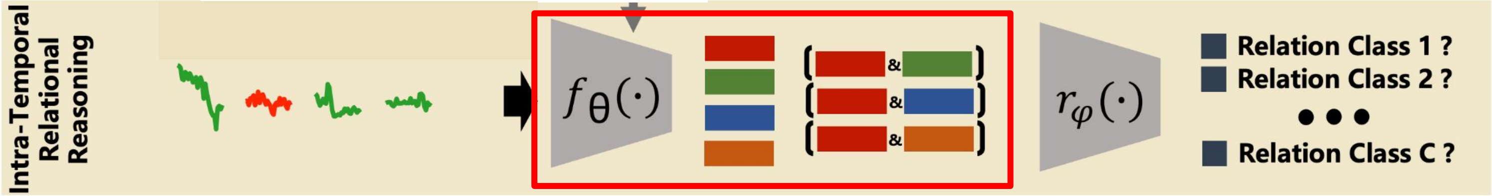
Require:
$\boldsymbol{t}_n$: A T-length time series.
$\boldsymbol{p}_{n,u}$, $\boldsymbol{p}_{n,v}$: two L-length pieces of $\boldsymbol{t}_n$.
C: Number of relation classes.

Ensure:
$y_n^{(u,v)} \in \{1, 2, ..., C\}$: The label of the temporal relation between $\boldsymbol{p}_{n,u}$ and $\boldsymbol{p}_{n,v}$.

$d_{u,v} = |u - v|, D = \lfloor T/C \rfloor$
if $d_{u,v} \leq D$ **then**
    $y_n^{(u,v)} = 0$
else if $d_{u,v} \leq 2 * D$ **then**
    $y_n^{(u,v)} = 1$
    ...
else if $d_{u,v} \leq (C-1) * D$ **then**
    $y_n^{(u,v)} = C - 2$
else
    $y_n^{(u,v)} = C - 1$
end if
return $y_n^{(u,v)}$

2. Methodology

(2) Intra-Sample Relational Reasoning



step 3) extract representations of time pieces

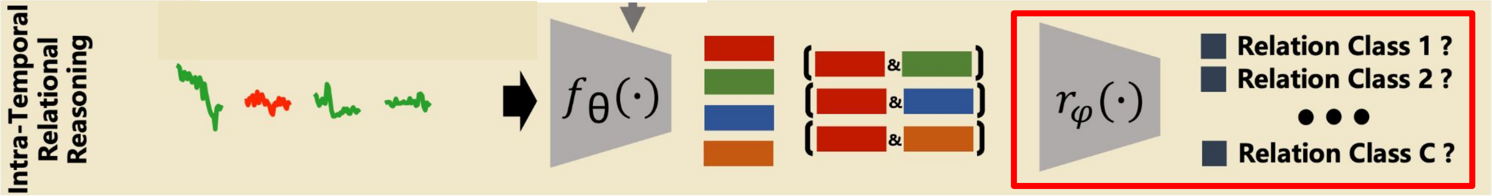
- Based on "sampled time pieces" & "their temporal relations"
- use shared backbone encoder f_θ
- extracted feature : $\boxed{\boldsymbol{z}_{n,u} = f_\theta\left(\boldsymbol{p}_{n,u}\right) \,\&\, \boldsymbol{z}_{n,v} = f_\theta\left(\boldsymbol{p}_{n,v}\right)}$

step 4) construct the temporal relation representation

- as $\boxed{[\boldsymbol{z}_{n,u}, \boldsymbol{z}_{n,v}].}$

2. Methodology

(2) Intra-Sample Relational Reasoning



step 5) relation reasoning

- input : $[\boldsymbol{z}_{n,u}, \boldsymbol{z}_{n,v}]$
- model : relation reasoning head $r_\varphi(\cdot)$
- output : final relation score (= $h_n^{(u,v)} = r_\varphi([\boldsymbol{z}_{n,u}, \boldsymbol{z}_{n,v}])$)
- task : multi-class classification problem
    - loss function : CE Loss ($\mathcal{L}_{\text{intra}} = -\sum_{n=1}^{N} y_n^{(u,v)} \cdot \log \frac{\exp\left(h_n^{(u,v)}\right)}{\sum_{c=1}^{C} \exp\left(h_n^{(u,v)}\right)}$)

2. Methodology

Summary

Jointly optimize 2 loss functions : $\boxed{\mathcal{L} = \mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}}$

$$\mathcal{L}_{\text{inter}} = -\sum_{n=1}^{2N} \sum_{i=1}^{K} \sum_{j=1}^{K} \left(y_n^{(i,j)} \cdot \log\left(h_n^{(i,j)}\right) + \left(1 - y_n^{(i,j)}\right) \cdot \log\left(1 - h_n^{(i,j)}\right) \right)$$

- $y_n^{(i,j)} = 1$ for pos relation
- $y_n^{(i,j)} = 0$ for neg relation

$$\mathcal{L}_{\text{intra}} = -\sum_{n=1}^{N} y_n^{(u,v)} \cdot \log \frac{\exp\left(h_n^{(u,v)}\right)}{\sum_{c=1}^{C} \exp\left(h_n^{(u,v)}\right)}$$

2. Methodology

Summary

Jointly optimize 2 loss functions : $\boxed{\mathcal{L} = \mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}}$

$$\boxed{\mathcal{L}_{\text{inter}} = -\sum_{n=1}^{2N} \sum_{i=1}^{K} \sum_{j=1}^{K} \left(y_n^{(i,j)} \cdot \log\left(h_n^{(i,j)}\right) + \left(1 - y_n^{(i,j)}\right) \cdot \log\left(1 - h_n^{(i,j)}\right)\right)}$$

- $y_n^{(i,j)} = 1$ for pos relation
- $y_n^{(i,j)} = 0$ for neg relation

$$\boxed{\mathcal{L}_{\text{intra}} = -\sum_{n=1}^{N} y_n^{(u,v)} \cdot \log \frac{\exp\left(h_n^{(u,v)}\right)}{\sum_{c=1}^{C} \exp\left(h_n^{(u,v)}\right)}}$$

- efficient algorithm compared with the traditional contrastive learning models such as SimCLR
- complexity
  - SimCLR : $O\left(N^2 K^2\right)$
  - SelfTime : $O\left(NK^2\right) + O(NK)$
    - $O\left(NK^2\right)$: complexity of inter-sample relation reasoning module
    - $O(NK)$: complexity of intra-tempora relation reasoning module

2. Methodology

Summary

Algorithm 2: SelfTime

Require:

Time series set $\mathcal{T} = \{\boldsymbol{t}_n\}_{n=1}^{N}$.

f_θ: Encoder backbone.

r_μ: Inter-sample relation reasoning head.

r_φ: Intra-temporal relation reasoning head.

Ensure:

f_θ: An updated encoder backbone.

```
for t_m, t_n ∈ T do
    Generate two augmentation sets A(t_m) and A(t_n)
    Sample positive relation pair (t_m^(i), t_m^(j)) and negative
        relation pair (t_m^(i), t_n^(j)) from A(t_m) and A(t_n)
    z_m^(i) = f_θ(t_m^(i))                          ▷ Sample representation
    z_m^(j) = f_θ(t_m^(j))                          ▷ Sample representation
    z_n^(j) = f_θ(t_n^(j))                          ▷ Sample representation
    h_{2m-1}^(i,j) = r_μ([z_m^(i), z_m^(j)])        ▷ Reasoning score of positive relation
    h_{2m}^(i,j) = r_μ([z_m^(i), z_n^(j)])          ▷ Reasoning score of negative relation
    Sample time piece relation pair (p_{n,u}, p_{n,v}) by Algorithm 1
    z_{n,u} = f_θ(p_{n,u})                          ▷ Time piece representation
    z_{n,v} = f_θ(p_{n,v})                          ▷ Time piece representation
    h_n^(u,v) = r_φ([z_{n,u}, z_{n,v}])             ▷ Reasoning score of intra-temporal relation
end for
```

$\mathcal{L}_{inter} = -\sum_{n=1}^{2N}\sum_{i=1}^{K}\sum_{j=1}^{K}(y_n^{(i,j)} \cdot \log(h_n^{(i,j)}) + (1-y_n^{(i,j)}) \cdot \log(1-h_n^{(i,j)}))$ ▷ Inter-sample relation reasoning loss

$\mathcal{L}_{intra} = -\sum_{n=1}^{N} y_n^{(u,v)} \cdot \log \frac{\exp(h_n^{(u,v)})}{\sum_{c=1}^{C}\exp(h_n^{(u,v)})}$ ▷ Intra-temporal relation reasoning loss

Update f_θ, r_μ, and r_φ by minimizing $\mathcal{L} = \mathcal{L}_{inter} + \mathcal{L}_{intra}$

return encoder backbone f_θ, throw away r_μ, and r_φ

3. Experiments

Experimental Settings

1. Time Series Augmentation
2. Ablation Study
3. TS Classification

Category	Dataset	Sample	Length	Class
Motion	CricketX	780	300	12
	UWaveGestureLibraryAll	4478	945	8
Sensor	DodgerLoopDay	158	288	7
	InsectWingbeatSound	2200	256	11
Device	MFPT	2574	1024	15
	XJTU	1920	1024	15

Table 1: Statistics of Datasets.

only small-scale datasets

3. Experiments

(1) Time Series Augmentation

- generally based on random transformation in 2 domains : magnitude & time
- magnitude domain :
  - values of TS where the values at each time step are modified but the time steps are constant
  - ex) jittering, scaling, magnitude warping (Um et al., 2017), cutout (DeVries & Taylor, 2017)
- time domain :
  - transformations are performed along the time axis
  - elements of the TS are displaced to different time steps
  - ex) time warping (Um et al., 2017), window slicing, and window warping (Le Guennec et al., 2016)

3. Experiments

(1) Time Series Augmentation

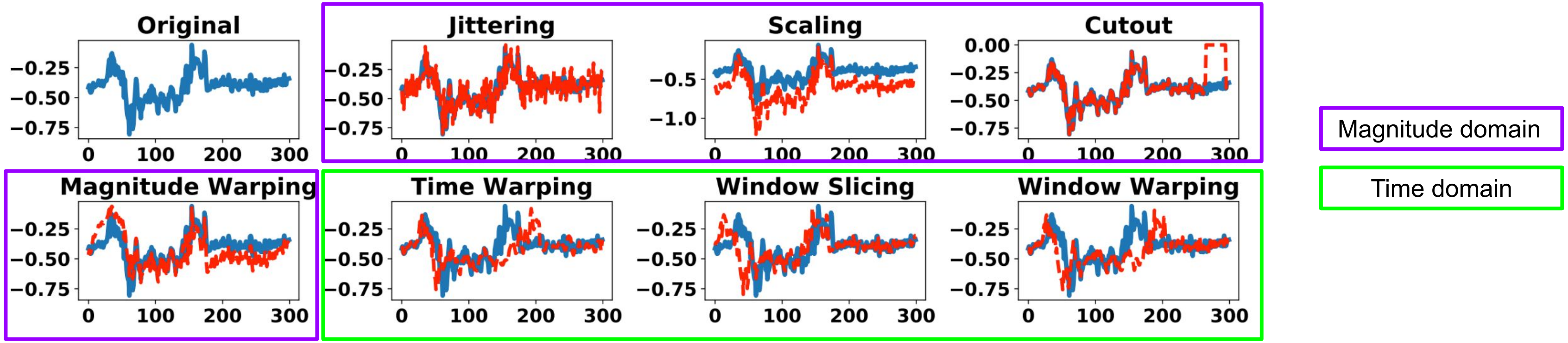


Figure 3: Data augmentations examples from CricketX dataset. The **blue** solid line is the original signal and the **red** dotted lines are the transformations.

3. Experiments

(1) Time Series Augmentation

set K = 16 augmentations for each sample

(although more augmentation results in better performance)

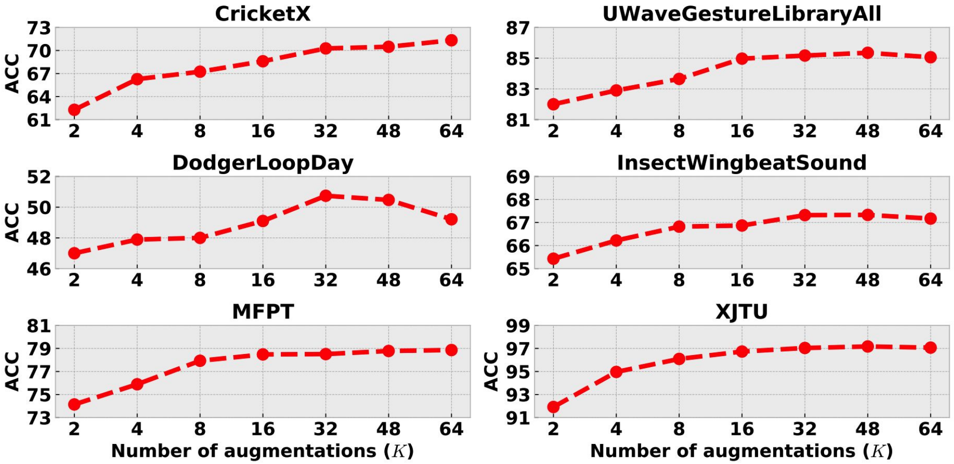


Figure 7: Impact of augmentation number K.

3. Experiments

(2) Ablation Study

- impact of different **temporal relation sampling** settings
- effectiveness of…
  - a) inter-sample relation reasoning
  - b) intra-temporal relation reasoning
  - c) SelfTime (= a) + b))

  under **different TS augmentation**

3. Experiments

(2) Ablation Study

Different temporal relation sampling settings

- Notation
  - C : numbers of temporal relation class
  - L : time piece length
- Goal : investigate the impact of class number (= C)
- Two cases :
  - (1) set L = 0.2∗T & vary C
  - (2) set C = 3 & vary L
- limitations : appropriate C & L differs by datasets !

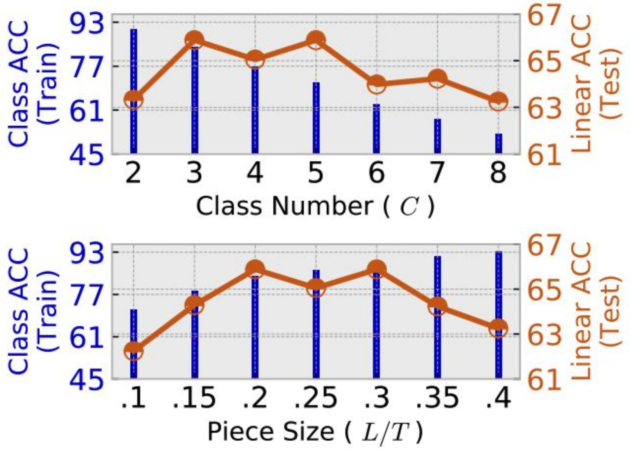


Figure 4: Impact of different temporal relation class numbers and piece sizes on CricketX dataset.

3. Experiments

(2) Ablation Study

Impact of Different Relation Modules and Data Augmentations (DAs)

- investigate the different DAs on the impact of linear evaluation for different modules
- Findings :
  - 1) composition of different DAs is crucial
  - 2) inter-sample relation reasoning is more sensitive to DA
  - 3) combining both (inter-sample & intra-temporal relation reasoning) is better
  - 4) combination of time-based DA & magnitude-based DA is better (?)

3. Experiments

Magnitude domain

Time domain

(2) Ablation Study

Impact of Different Relation Modules and Data Augmentations (DAs)

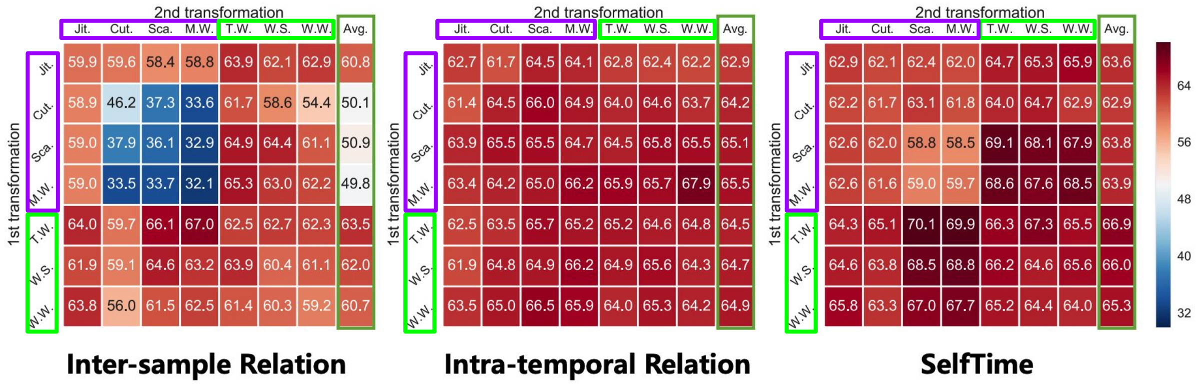


Figure 5: Linear evaluation on CricketX under individual or composition of data augmentations. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

3. Experiments

Magnitude domain

Time domain

(2) Ablation Study

Impact of Different Relation Modules and Data Augmentations (DAs)

Inter-sample Relation **Intra-temporal Relation** **SelfTime**

(a) UWaveGestureLibraryAll

Inter-sample Relation **Intra-temporal Relation** **SelfTime**

(b) DodgerLoopDay

3. Experiments

(3) TS Classification

- a) linear evaluation
- b) transfer learning
- c) qualitative evaluation

3. Experiments

(3) TS Classification

a) linear evaluation

- baselines : considers …
  - global features (Deep InfoMax, Transformation, SimCLR, Relation)
  - local features (Triplet Loss, Deep InfoMax, Forecast)
  - temporal information of TS (Triplet Loss and Forecast)
- ***SelfTime : considers all the above!***

3. Experiments

(3) TS Classification

a) linear evaluation

Method	Dataset					
	CricketX	UGLA	DLD	IWS	MFPT	XJTU
Supervised	62.44±1.53	87.83±0.32	37.05±1.61	66.23±0.45	80.29±0.8	95.9±0.42
Random Weights	36.9±0.92	70.01±1.68	32.95±2.57	52.85±1.36	46.68±2.35	52.58±4.67
Triplet Loss (Franceschi et al., 2019)	40.01±2.64	71.41±1.1	41.37±2.47	53.61±2.82	47.87±2.97	53.31±3.43
Deep InfoMax (Hjelm et al., 2019)	49.16±3.03	73.88±2.37	38.95±2.47	55.99±1.31	58.99±2.72	76.27±1.83
Forecast (Jawed et al., 2020)	44.59±1.09	75.7±0.9	38.74±3.05	54.89±1.99	52.6±1.65	62.28±2.55
Transformation (Sarkar & Etemad, 2020)	52.12±2.02	75.41±0.27	35.47±1.56	59.68±1.2	60.33±3.29	85.08±2.01
SimCLR (Chen et al., 2020)	59.0±3.19	74.9±0.92	37.74±3.8	56.19±0.98	71.81±1.21	88.84±0.63
Relation (Patacchiola & Storkey, 2020)	65.3±0.43	80.87±0.78	42.84±3.23	62.0±1.49	73.53±0.65	95.14±0.72
SelfTime (ours)	**68.6±0.66**	**84.97±0.83**	**49.1±2.93**	**66.87±0.71**	**78.48±0.94**	**96.73±0.76**

Table 2: Linear evaluation of representations learned by different models on different datasets.

3. Experiments

(3) TS Classification

b) transfer learning

training on the **unlabeled source dataset A**

& conduct linear evaluation on the **labeled target dataset B**

Method	Source→Target					
	UGLA→CricketX	CricketX→UGLA	IWS→DLD	DLD→IWS	XJTU→MFPT	MFPT→XJTU
Supervised	31.31±2.76	71.85±1.2	22.9±2.55	44.31±3.25	63.15±2.08	82.58±3.98
Random Weights	36.9±0.92	70.01±1.68	32.95±2.57	52.85±1.36	46.68±2.35	52.58±4.67
Triplet Loss (Franceschi et al., 2019)	30.08±4.66	55.32±2.51	34.67±3.12	45.22±3.09	53.75±2.96	59.24±3.02
Deep InfoMax (Hjelm et al., 2019)	45.92±2.3	64.2±4.19	37.42±1.99	47.75±1.74	56.75±0.77	77.14±3.14
Forecast (Jawed et al., 2020)	32.67±0.86	72.42±1.17	25.47±2.93	55.39±1.36	53.08±2.75	61.74±3.92
Transformation (Sarkar & Etemad, 2020)	39.24±2.25	70.4±1.98	30.0±1.66	**57.71±0.83**	54.71±1.68	69.81±8.03
SimCLR (Chen et al., 2020)	45.48±3.46	65.67±2.91	36.21±1.02	36.2±4.03	63.11±2.0	81.62±3.95
Relation (Patacchiola & Storkey, 2020)	52.55±2.67	75.67±0.54	36.0±1.52	56.29±1.82	70.27±1.14	92.77±1.15
SelfTime (ours)	**55.04±2.58**	**77.77±0.35**	**45.0±1.48**	**57.8±1.33**	**75.06±1.84**	**93.79±2.46**

Table 3: Domain transfer evaluation by training with self-supervision on the unlabeled source data and linear evaluation on the labeled target data (e.g. source→target: UGLA→CricketX).

3. Experiments

(3) TS Classification

c) qualitative evaluation

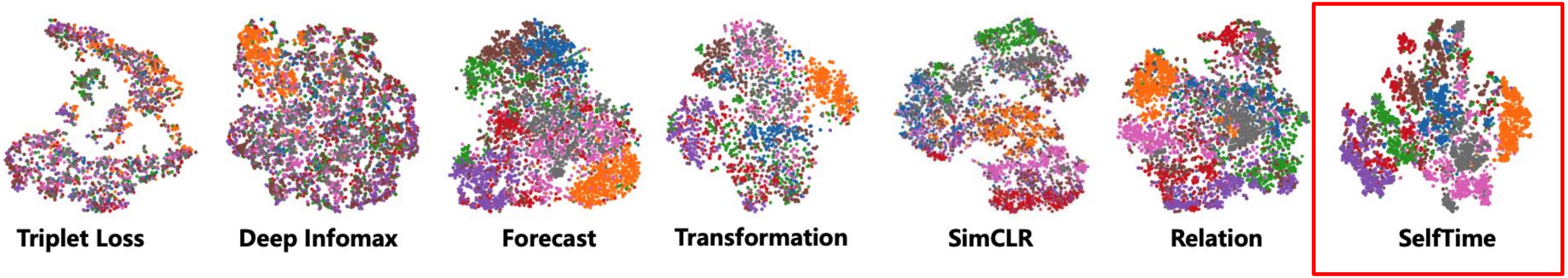


Figure 6: t-SNE visualization of the learned feature on UGLA dataset. Different colors indicate different labels.

Papers

1. Self-supervised Time Series Representation Learning by Inter-Intra Relational Reasoning (2020)
2. **Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification (2022)**

SELF-SUPERVISED TIME SERIES REPRESENTATION LEARNING BY INTER-INTRA RELATIONAL REASONING

Haoyi Fan, Fengbin Zhang
School of Computer Science and Technology
Harbin University of Science and Technology
Harbin 150080, China
isfanhy@hrbust.edu.cn

Yue Gao
School of Software
Tsinghua University
Beijing 100084, China
kevin.gaoy@gmail.com

ABSTRACT

Self-supervised learning achieves superior performance in many domains by extracting useful representations from the unlabeled data. However, most of traditional self-supervised methods mainly focus on exploring the inter-sample structure while less efforts have been concentrated on the underlying intra-temporal structure, which is important for time series data. In this paper, we present **Self-Time**: a general **Self**-supervised **Time** series representation learning framework, by exploring the inter-sample relation and intra-temporal relation of time series to learn the underlying structure feature on the unlabeled time series. Specifically, we first generate the inter-sample relation by sampling positive and negative sam-

https://arxiv.org/pdf/2011.13548.pdf

Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li and Cunta
Guan *Fellow, IEEE*

Abstract—Learning time-series representations when only unlabeled data or few labeled samples are available can be a challenging task. Recently, contrastive self-supervised learning has shown great improvement in extracting useful representations from unlabeled data via contrasting different augmented views of data. In this work, we propose a novel **T**ime-**S**eries representation learning framework via **T**emporal and **C**ontextual **C**ontrasting (**TS-TCC**) that learns representations from unlabeled data with contrastive learning. Specifically, we propose time-series-specific weak and strong augmentations and use their views to learn robust temporal relations in the proposed temporal contrasting module, besides learning discriminative representations by our proposed contextual contrasting module. Additionally, we conduct a systematic study of time-series data augmentation selection, which is a key part of contrastive learning. We also extend TS-TCC to the semi-supervised learning settings and propose a **C**lass-**A**ware TS-**TCC** (**CA-TCC**) that benefits from the available few labeled data to further improve representations learned by TS-TCC. Specifically, we leverage the robust pseudo labels produced by TS-TCC to realize a class-aware contrastive loss. Extensive experiments show that the linear evaluation of the features learned by our proposed framework performs comparably with the fully supervised training. Additionally, our framework shows high efficiency in few labeled data and transfer learning scenarios. The code is publicly available at https://github.com/emadeldeen24/CA-TCC.

Index Terms—self-supervised learning, semi-supervised learning, time-series classification, temporal contrasting, contextual contrasting, augmentation.

https://arxiv.org/pdf/2208.06616.pdf

Contents

1. Methods

TS-TCC (Time-Series representation learning framework via Temporal and Contextual Contrasting)

Notation

- input : x
- strong & weak sample : $x^s \sim \mathcal{T}_s$ and $x^w \sim \mathcal{T}_w$
  - then passed to the encoder
- Encoder : $\mathbf{z} = f_{enc}(\mathbf{x})$
  - 3 block convolutional architecture
- Encoded : $\mathbf{z} = [z_1, z_2, \ldots z_T]$
  - where T is the total timesteps, $z_i \in \mathbb{R}^d$, where d is the feature length
- Encoded (weak & strong) : $\mathbf{z}^s$ & $\mathbf{z}^w$
  - then fed into the **temporal contrasting module**

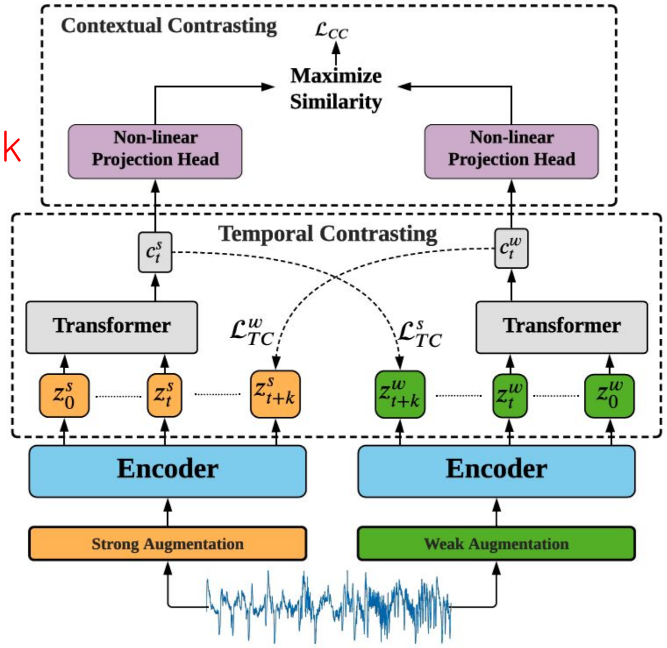


Fig. 2: Overall architecture of the proposed TS-TCC. The Temporal Contrasting module learns robust temporal features by a tough cross-view prediction task. The Contextual Contrasting learns discriminative features by maximizing the similarity between the contexts of the same sample, while minimizing its similarity with the other samples within the mini-batch.

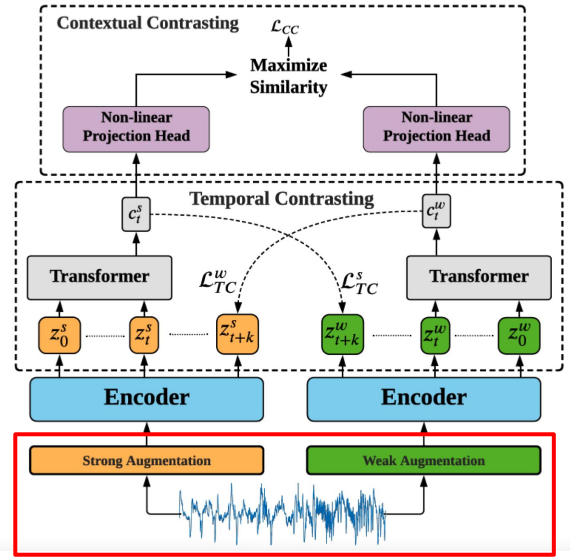
1. Methods

(1) Time-Series Data Augmentation

argue that ***producing views from DIFFERENT augmentations***

can improve the robustness of the learned representations

→ propose 2 separate augmentations : **WEAK** & **STRONG**



STRONG augmentation

- to enable the tough cross-view prediction task in the next module
- helps in learning robust representations

WEAK augmentation

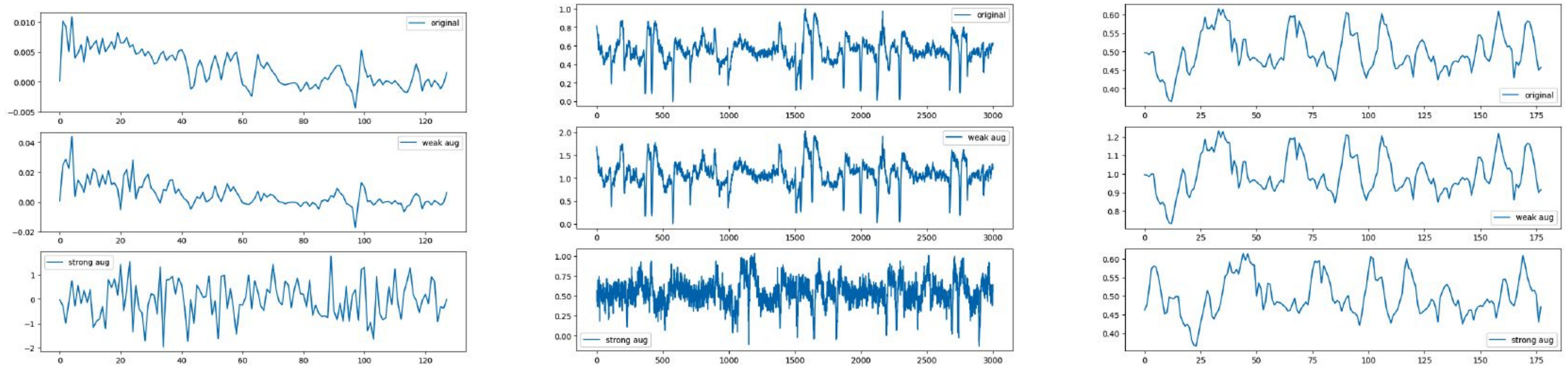
- aims to add some small variations to the signal w.o affecting its characteristics

1. Methods

(1) Time-Series Data Augmentation

TABLE 7: A study of TS-TCC *linear evaluation* performance with 5% of labeled HAR data when using different variations of weak and strong augmentations.

Weak Augmentation	Strong Augmentation	Accuracy	MF1-score
scale	no Aug	46.12	36.67
scale + jitter	no Aug	56.98	50.88
no Aug	permutation	62.07	52.64
no Aug	jitter + permutation	72.35	68.03
time Shift	jitter + permutation	71.57	66.97
time Shift + jitter	jitter + permutation	74.69	69.33
scale	jitter + permutation	72.59	68.90
scale + jitter	jitter + permutation	**77.58**	**76.66**



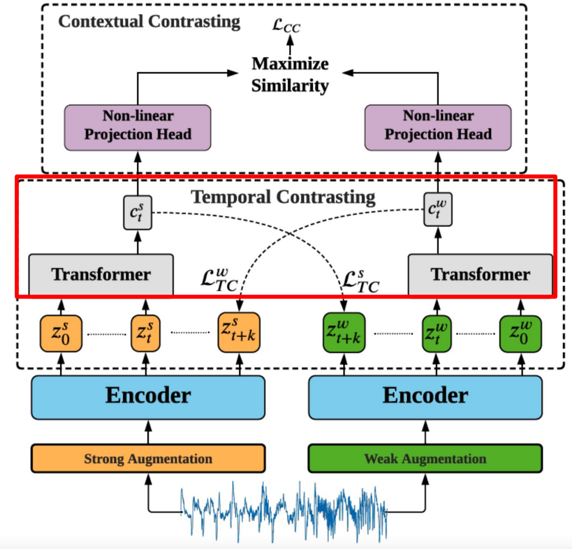
(a) HAR (b) Sleep-EDF (c) Epilepsy

Fig. 9: Sample from each adopted dataset after normalization along with its weak and strong augmented views. The first row is the original samples, the second row shows the weak augmented views, and the third row shows the strong augmented views.

1. Methods

(2) Temporal Contrasting

- deploys a contrastive loss to extract temporal features
- with an AR model (f_{ar})
  - generates context vector : $c_t = f_{ar}\left(\mathbf{z}_{\leq t}\right), c_t \in \mathbb{R}^h$
    - h is the hidden dimension of f_{ar}
  - c_t is then used to predict the timesteps from z_{t+1} until $z_{t+k}(1 < k \leq K)$
    - use log-bilinear model ... $f_k\left(x_{t+k}, c_t\right) = \exp\left(\left(\mathcal{W}_k\left(c_t\right)\right)^T z_{t+k}\right)$



1. Methods

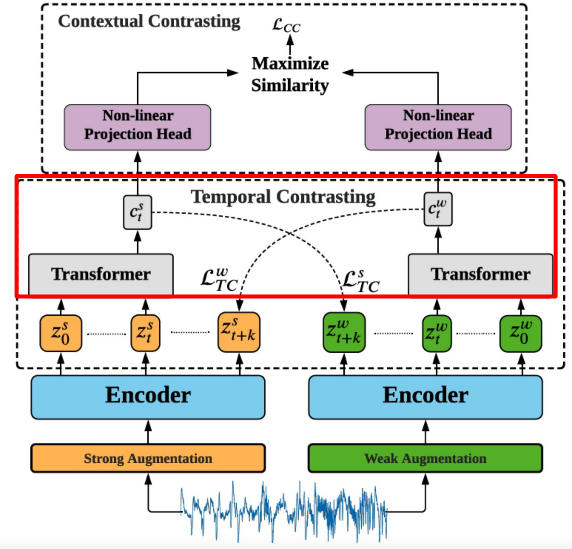
(2) Temporal Contrasting

propose a tough **cross-view prediction** task

- use c_t^s to predict future timesteps of z_{t+k}^w
- use c_t^w to predict future timesteps of z_{t+k}^s

$$\mathcal{L}_{TC}^s = -\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp\left(\left(\mathcal{W}_k\left(c_t^s\right)\right)^T z_{t+k}^w\right)}{\sum_{n\in\mathcal{N}_{t,k}}\exp\left(\left(\mathcal{W}_k\left(c_t^s\right)\right)^T z_n^w\right)}$$

$$\mathcal{L}_{TC}^w = -\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp\left(\left(\mathcal{W}_k\left(c_t^w\right)\right)^T z_{t+k}^s\right)}{\sum_{n\in\mathcal{N}_{t,k}}\exp\left(\left(\mathcal{W}_k\left(c_t^w\right)\right)^T z_n^s\right)}$$

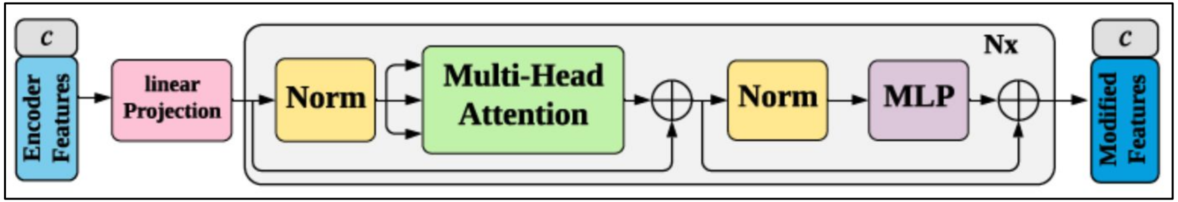


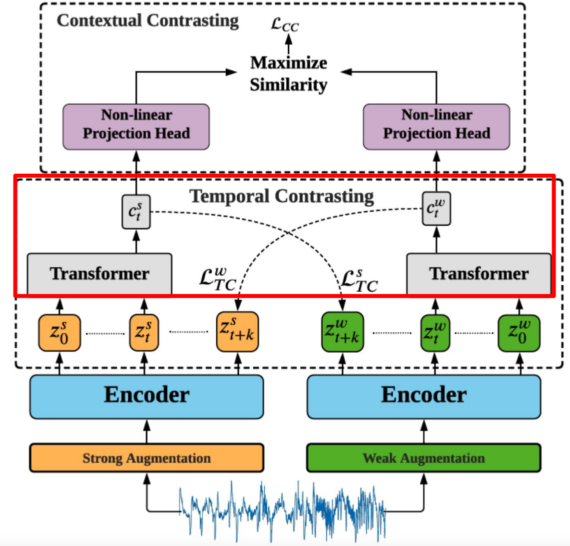
1. Methods

(2) Temporal Contrasting

use **Transformer** as an AR model

- mainly consists of multi-headed attention (MHA) followed by a Multilayer Perceptron (MLP)
  - MLP : 2 FC layer + ReLU + dropout
- use Pre-norm residual connections
- stack L identical layers to generate the final features
- add a token $c \in \mathbb{R}^h$ to the input
  - acts as a representative context vector in the output





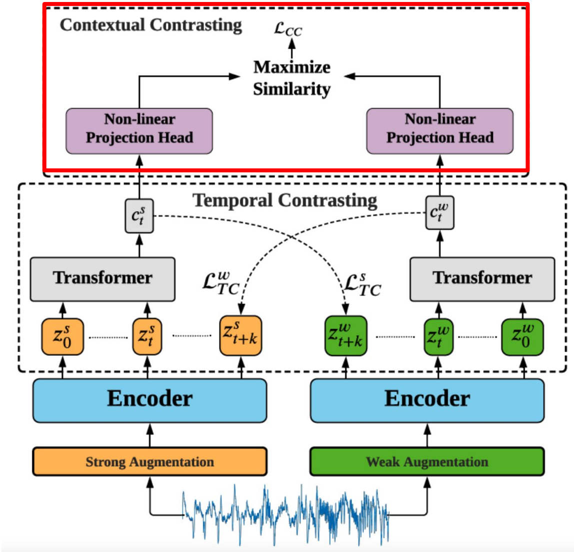
1. Methods

(3) Contextual Contrasting

step 1) apply a non-linear transformation to the contexts

- via projection head
  - maps the contexts into the space where the contextual contrasting is applied
- ex) given a batch of N input samples ... will have 2 contexts for each sample

  $\rightarrow$ Have $2N$ contexts
- Notation) for context c_t^i ....
  - positive sample : $c_t^{i^+}$
  - positive pair : $(c_t^i, c_t^{i^+})$
  - negative samples : remaining $(2N-2)$ contexts



1. Methods

(3) Contextual Contrasting

step 2) Contextual Contrasting Loss

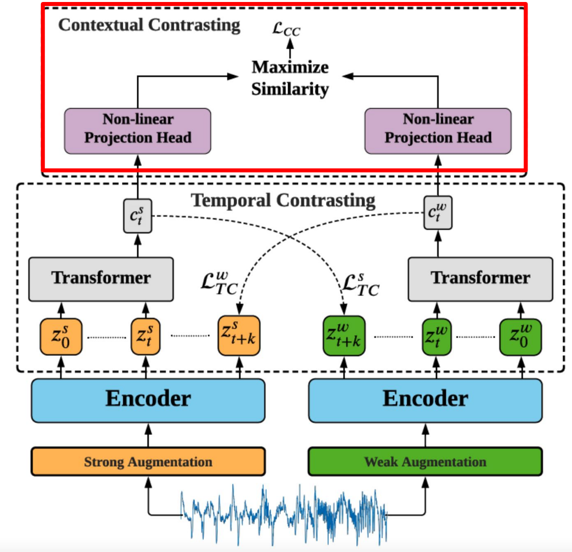
$$\mathcal{L}_{CC} = \frac{1}{2N} \sum_{k=1}^{2N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$$

$$\ell\left(i, i^{+}\right) = -\log \frac{\exp\left(\operatorname{sim}\left(c_t^i, c_t^{i+}\right)/\tau\right)}{\sum_{m=1}^{2N} \mathbb{1}_{[m \neq i]} \exp\left(\operatorname{sim}\left(c_t^i, c_t^m\right)/\tau\right)}$$

step 3) Overall SSL Loss

- (1) temporal contrasting loss
- (2) contextual contrasting loss

$$\rightarrow \mathcal{L}_{\text{unsup}} = \lambda_1 \cdot (\mathcal{L}_{TC}^s + \mathcal{L}_{TC}^w) + \lambda_2 \cdot \mathcal{L}_{CC}$$



1. Methods

(4) Class-Aware TS-TCC

Drawback : ***considering all the samples in mini-batch as negative pairs***

→ solution : **CA TCC (Class-Aware TS-TCC)**

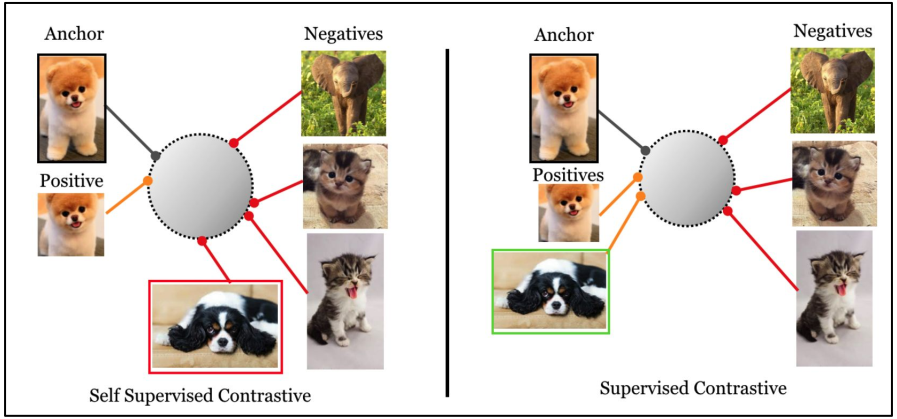
https://arxiv.org/pdf/2004.11362.pdf

Preliminaries : **Supervised Contrastive Learning**

→ use information of (few) labeled dataset!

→ use single (X) multiple (O) POSITIVE

- same class : POSITIVE
- others : NEGATIVE



1. Methods

(4) Class-Aware TS-TCC

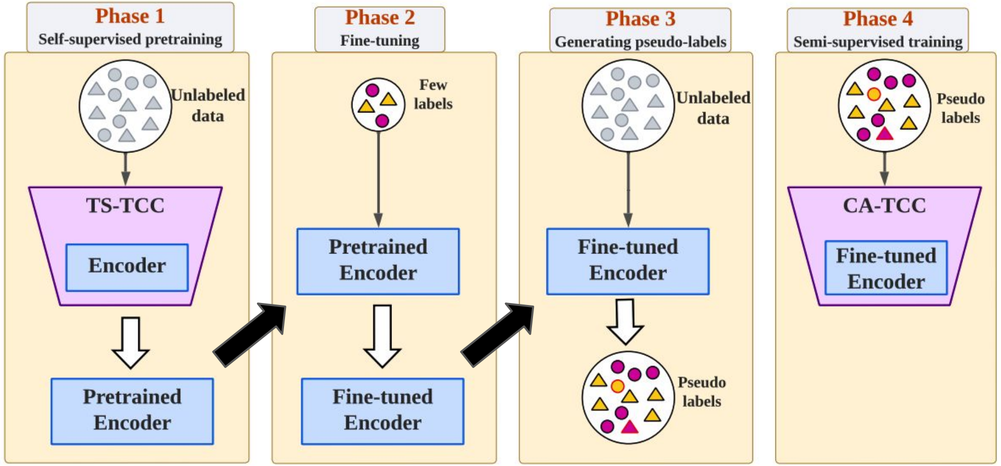


Fig. 3: The four phases for CA-TCC semi-supervised training. In Phase 1, TS-TCC is trained with fully unlabeled data. Next, we use the available few labeled samples to fine-tune the pretrained encoder in Phase 2. Following that, in Phase 3, we generate pseudo labels for the unlabeled data with the fine-tuned TS-TCC encoder. Finally, Phase 4 includes deploying the pseudo-labeled data in the semi-supervised training.

1. Methods

(4) Class-Aware TS-TCC

Notation

- N labeled samples $\{\mathbf{x}_k, y_k\}_{k=1...N}$
- after augmentations) $2N$ samples, $\{\hat{\mathbf{x}}_l, \hat{y}_l\}_{l=1...2N}$
  - such that $\hat{\mathbf{x}}_{2k}$ and $\hat{\mathbf{x}}_{2k-1}$ are the two views of $\mathbf{x}_k$
  - $y_k = \hat{y}_{2k} = \hat{y}_{2k-1}$.
- $A(i) \equiv I \backslash \{i\}$,

Supervised Contextual Contrasting Loss

$$\mathcal{L}_{SCC} = \sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i)} \ell(i, p)$$

- $P(i) = \{p \in A(i) : \hat{y}_p = \hat{y}_i\}$
- $|P(i)|$: cardinality of $P(i)$

Overall Loss

$$\boxed{\mathcal{L}_{\text{semi}} = \lambda_3 \cdot (\mathcal{L}_{TC}^s + \mathcal{L}_{TC}^w) + \lambda_4 \cdot \mathcal{L}_{SCC}}$$

2. Experiments

Experimental Settings

1. Linear Evaluation
2. Supervised Contrastive Loss
3. Semi-supervised Training
4. Transfer Learning

Dataset	# Train	# Test	Length	# Channel	# Class
HAR	7,352	2,947	128	9	6
Sleep-EDF	25,612	8,910	3,000	1	5
Epilepsy	9,200	2,300	178	1	2
FD	8,184	2,728	5,120	1	3
Wafer	1,000	6,174	152	1	2
FordA	1,320	3,601	500	1	2
FordB	810	3,636	500	1	2
POC	1,800	858	80	1	2
PPOC	600	291	80	1	2
StarLightCurves	1,000	8,236	1,024	1	3
ElectricDevices	8,926	7,711	96	1	7

Metrics

$$\text{Accuracy} = \frac{\sum_{i=1}^{\mathcal{K}} TP_i}{N},$$

$$\text{MF1-score} = \frac{1}{\mathcal{K}} \sum_{i=1}^{\mathcal{K}} \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i},$$

2. Experiments

(1) Linear Evaluation

step 1) pretrain (w.o labeled data)

step 2) evaluation (with a portion of the labeled data)

- standard linear evaluation scheme
- train a linear classifier on top of a frozen SSL pretrained encoder model

Implications

- contrastive methods > pretext-based method
  - contrastive methods : CPC, SimCLR and our TS-TCC
  - pretext-based method : SSL-ECG
- CPC > SimCLR
  - temporal features are more important than general features in TS data

2. Experiments

(1) Linear Evaluation

	Random Init		Supervised		SSL-ECG		CPC		SimCLR		TS-TCC	
1% of labeled data												
Datasets	Accuracy	MF1-score	Accuracy	MF1-score	Accuracy	MF1-score	Accuracy	MF1-score	Accuracy	MF1-score	Accuracy	MF1-score
HAR	39.8±3.8	34.7±5.0	44.9±6.7	41.0±6.7	60.0±4.0	54.0±6.0	65.4±1.4	63.8±1.7	65.8±0.7	64.3±0.9	**70.5±0.3**	**69.5±0.5**
Sleep-EDF	20.5±1.0	20.7±1.4	34.1±0.3	30.8±0.8	70.9±0.8	61.6±0.7	74.7±0.2	68.7±0.0	58.0±1.0	56.9±0.6	**75.8±0.4**	**70.0±0.2**
Epilepsy	70.3±2.1	66.2±2.6	76.1±0.7	74.8±0.4	89.3±0.4	86.0±0.3	88.9±1.1	85.8±0.3	88.3±1.5	84.0±1.0	**91.2±0.5**	**89.2±0.2**
Wafer	90.6±1.6	58.1±2.1	91.9±1.3	67.6±9.2	93.4±0.5	76.1±2.4	93.5±0.4	78.4±1.5	**93.8±0.2**	**78.5±1.1**	93.2±0.8	76.7±4.6
FordA	50.6±2.1	36.6±4.2	56.4±1.6	54.4±3.5	67.9±8.8	66.2±9.5	75.8±1.4	75.2±1.8	55.9±3.7	55.7±3.8	**80.6±2.0**	**80.0±2.4**
FordB	52.5±1.8	47.5±6.4	51.9±2.6	48.0±3.6	64.4±6.2	60.5±7.9	66.8±3.1	65.0±3.9	50.9±1.3	49.8±2.2	**72.7±0.9**	**71.9±1.0**
POC	61.4±0.0	38.3±0.0	62.0±0.8	40.0±2.1	62.5±1.8	41.2±4.9	**64.8±1.0**	**48.2±2.9**	61.5±0.1	38.4±0.3	63.8±0.5	48.1±0.9
PPOC	68.4±0.0	40.6±0.0	**64.3±0.7**	**64.2±0.7**	49.8±7.3	37.6±8.4	63.3±0.7	63.0±0.8	37.6±5.1	32.8±7.7	63.4±0.3	63.1±0.4
StarLightCurves	83.9±1.6	65.4±3.6	78.8±0.9	71.4±0.1	78.3±0.9	72.0±0.8	80.8±1.4	74.4±0.6	80.6±0.6	71.6±0.2	**86.0±0.4**	**79.2±0.7**
ElectricDevices	50.8±4.4	41.9±3.6	57.8±1.1	47.5±1.0	60.1±4.1	50.0±4.9	59.3±4.1	48.9±6.7	62.5±1.1	51.2±0.5	**63.6±1.2**	**56.4±0.6**
Average	58.8	45.0	61.8	54.0	69.7	60.5	73.3	67.1	65.5	58.3	**76.1**	**70.4**
5% of labeled data												
HAR	49.6±2.5	45.8±2.0	52.8±1.5	50.9±0.2	63.7±5.3	58.6±7.4	75.4±2.1	74.7±2.5	75.8±1.4	74.9±1.5	**77.6±1.8**	**76.7±1.7**
Sleep-EDF	22.8±2.8	22.8±2.2	60.5±3.9	54.8±5.5	73.4±0.5	63.7±0.1	76.3±0.4	70.5±0.3	64.2±1.0	61.9±0.8	**77.0±0.6**	**70.9±0.5**
Epilepsy	75.5±3.6	70.5±3.3	83.4±0.7	80.4±0.7	92.8±0.2	89.0±0.3	92.8±0.3	90.2±0.5	91.3±0.5	89.2±1.0	**93.1±0.3**	**93.7±0.6**
Wafer	91.2±1.2	65.5±8.2	94.6±0.3	83.9±0.6	**94.9±0.3**	**84.5±0.7**	92.5±0.4	79.4±0.8	94.8±0.2	83.3±0.6	93.2±0.4	81.2±0.7
FordA	54.4±2.4	50.5±7.6	54.5±4.3	44.1±8.0	73.6±1.2	70.7±1.5	86.5±1.9	86.5±1.9	69.6±1.3	68.9±1.7	**89.9±0.1**	**89.9±0.1**
FordB	51.3±3.2	48.2±5.3	60.5±2.8	58.8±3.7	71.7±3.1	69.8±3.8	**86.3±0.8**	**86.2±0.8**	63.0±3.0	60.7±4.2	86.1±1.5	85.9±1.6
POC	61.6±0.3	38.8±1.0	61.4±0.0	38.3±0.0	62.9±0.3	43.3±1.4	**66.9±2.6**	**44.3±8.4**	62.7±1.1	42.4±4.0	62.6±1.1	42.6±3.0
PPOC	64.1±2.8	57.4±9.8	69.1±2.4	62.2±6.4	68.8±0.0	40.7±0.0	71.5±3.4	63.9±1.5	48.0±2.3	42.9±2.1	**72.1±4.4**	**64.2±3.7**
StarLightCurves	74.2±1.4	69.8±4.1	81.8±0.8	71.4±4.1	82.6±1.3	74.5±1.3	**89.1±1.0**	**84.5±0.8**	84.2±1.3	74.0±2.3	89.6±0.2	82.7±0.9
ElectricDevices	57.4±1.2	52.3±1.0	59.7±0.7	55.6±0.8	63.7±0.8	56.1±2.2	62.4±0.6	58.1±0.6	63.9±1.2	58.6±0.6	**65.1±0.3**	**59.2±0.4**
Average	60.2	52.2	67.8	60.0	74.8	65.1	80.0	73.8	71.8	65.7	**80.6**	**74.7**

2. Experiments

(2) Supervised Contrastive Loss

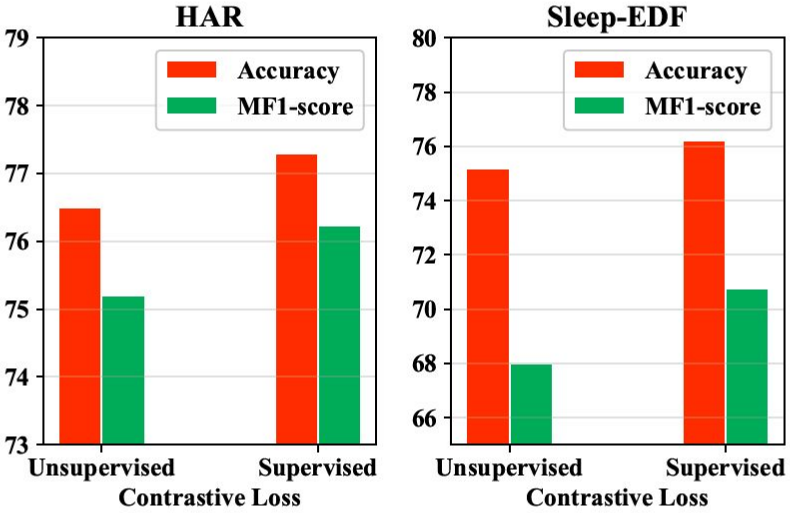


Fig. 7: Performance comparison with and without supervised contrastive loss in the semi-supervised training (Phase 4).

2. Experiments

(3) Semi-supervised Training

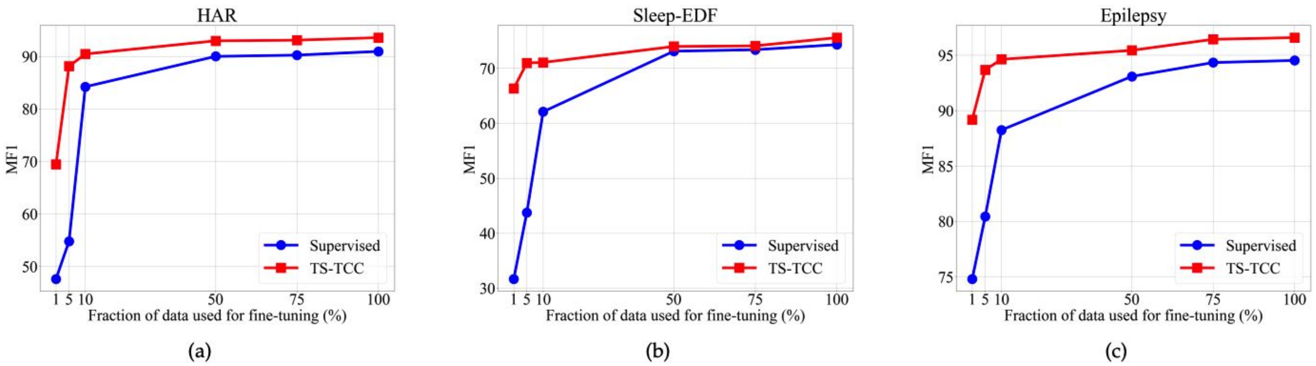


Fig. 5: Comparison between supervised training vs. TS-TCC *fine-tuning* with different few labeled data scenarios in terms of MF1.

investigate TS-TCC, under different semi-supervised settings

- fine-tune the pretrained model using 1%, 5%, 10%, 50%, 75% data
- metric : MF1 (∵ imbalance of Sleep-EDF)

2. Experiments

Dataset : Fault Diagnosis (FD)

- has 4 working conditions (= 4 domains, A/B/C/D)

(4) Transfer Learning

Process

- step 1) train the model on the data from one condition (i.e., source domain)
- step 2) test it on another condition (i.e., target domain)

3 training schemes on the source domain

- (1) Supervised training
- (2) TS-TCC fine-tuning
- (3) CATCC fine-tuning

TABLE 3: Cross-domain transfer learning experiments performed on Fault Diagnosis dataset in terms of accuracy. For CA-TCC, we used 1% of labeled data for pretraining. Best results are in bold and second bests are underlined.

Method	A→B	A→C	A→D	B→A	B→C	B→D	C→A	C→B	C→D	D→A	D→B	D→C	AVG
Supervised	34.38	44.94	34.57	**52.93**	63.67	99.82	**52.93**	84.02	83.54	**53.15**	99.56	62.43	63.83
TS-TCC	43.15	51.50	42.74	47.98	70.38	99.30	38.89	98.31	99.38	51.91	99.96	70.31	67.82
CA-TCC	**44.75**	**52.09**	**45.63**	46.26	**71.33**	**100.0**	52.71	**99.85**	**99.84**	46.48	**100.0**	**77.01**	**69.66**

Thank You!