Representation Learning for Time Series

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
Yein Park

Outline

- Abstract
- Background
- Problem Formulation
- Introduction
- Proposed Method
- Experimental Results
- Conclusion

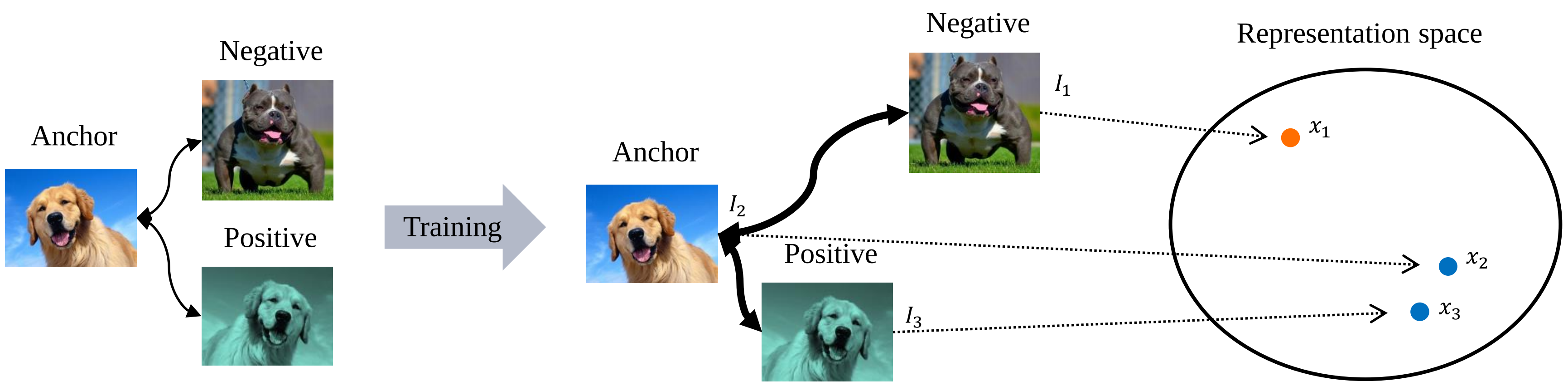
"SimTS: Rethinking Contrastive Representation Learning for Time Series Forecasting." *arXiv*, 2023.

Abstract

- Contrastive learning
  - Meaningful representations for image or time series classification
  - Less effectiveness for time series forecasting
    - Task of predicting future state from history context
    - Instance discrimination optimization that cannot be applied directly
  - Generalization limits for different types of time series data
    - High dependence on specific time series characteristics of the construction of positive and negative pairs
- SimTS, a simple representation learning approach for improving time series forecasting
  - Learning to predict the future from the past in the latent space
  - No reliance on negative pairs or specific assumptions

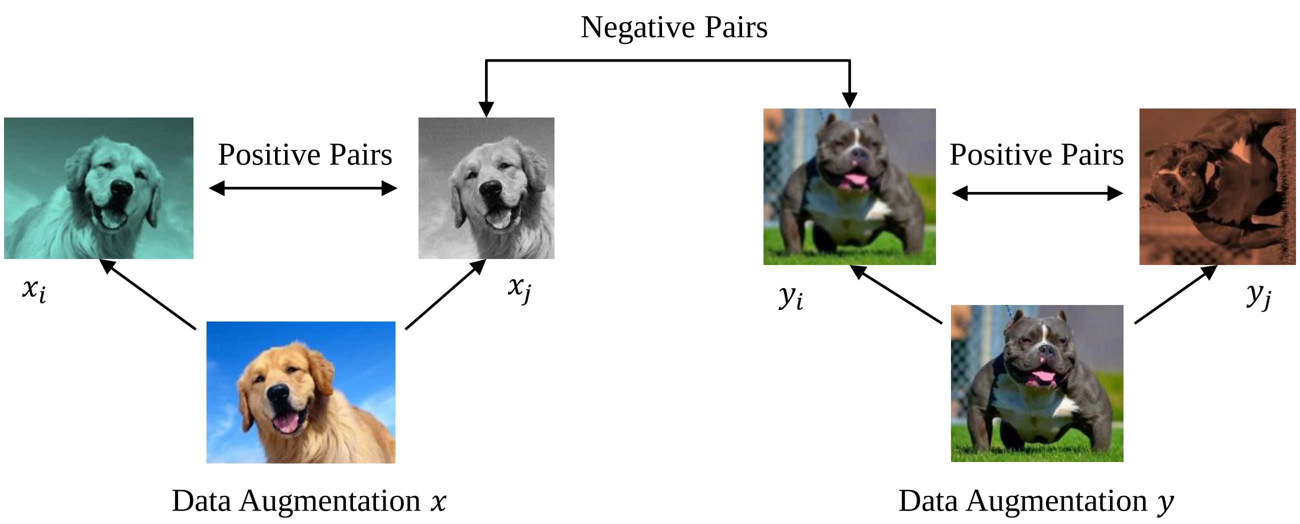
Background

- Contrastive learning
  - A method of representation learning
  - Representation learning through comparison between input samples
    - Learning the representation space so that "similar" data is close and "different" data is far
    - For multiple input pairs, the similarity is learned by label



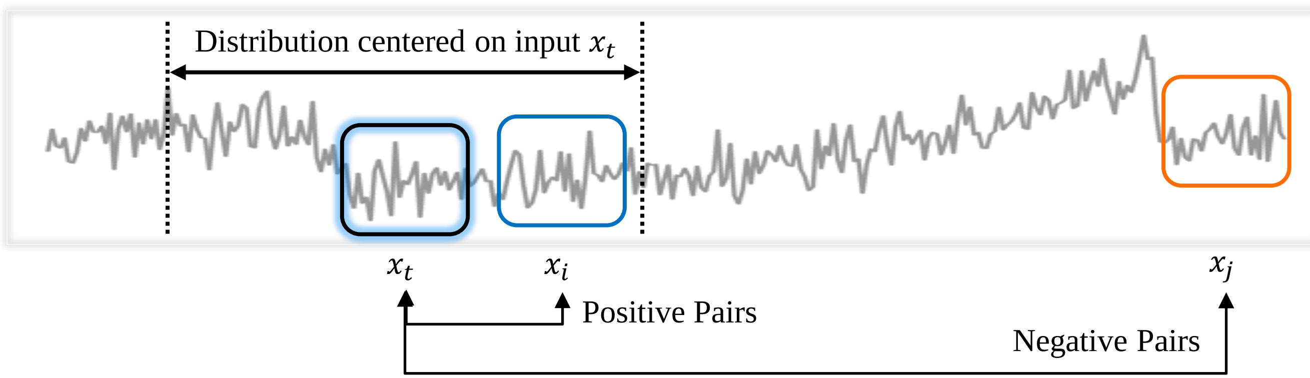
Background

- Contrastive learning
  - Generation of input pairs through data augmentation
    - Data augmentation in image domain
      - Positive pair → data augmented on the same image
      - Negative pair → data augmented on the different images
      - Random crop, rotation, resizing, shifting, noising, blur, color distortion, perspective distortion



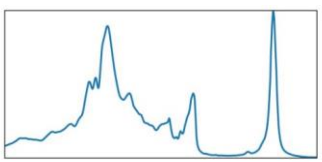
Background

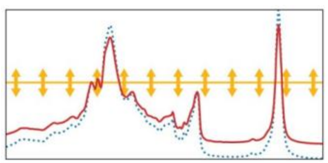
- Contrastive learning
  - Generation of input pairs through data augmentation
    - Data augmentation in <u>time series domain</u>
      - Input-centered distribution
        - ✓Positive pair → belonging to input-centered distribution, neighborhood
        - ✓Negative pair → not belonging to input-centered distribution, non-neighborhood

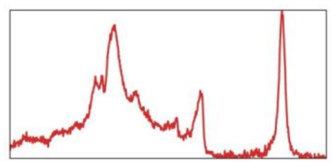


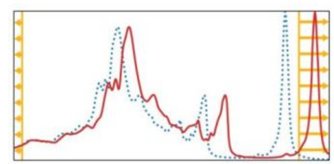
Background

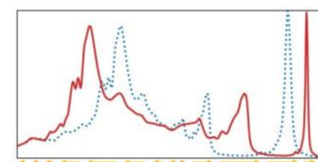
- Contrastive learning
  - Generation of input pairs through data augmentation
    - Data augmentation in <u>time series domain</u>
      - Scaling, jittering, window slicing, time warping
      - Random masking
      - Random warping
      - Random reordering


Original

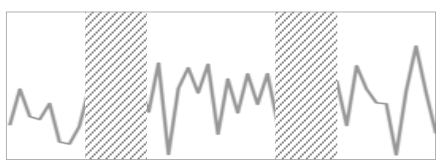

Scaling

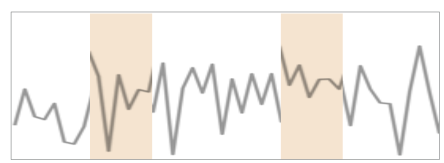

Jittering

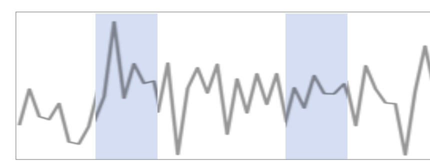

Window slicing


Time warping


Original

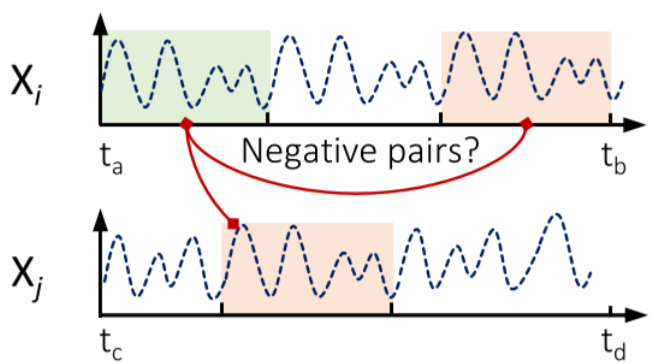

Random masking


Random warping


Random reordering

Problem Formulation

- Contrastive learning
  - Learning representations close to positive samples and far from negative samples
    - Well suited for classification tasks
      - The resulting representation contains information to better distinguish different instances of the time series
    - Ineffective for forecasting tasks
      - Forecasting the future based on past data
  - Negative sample methods of existing methods that are difficult to trust
    - Time stamp difference, different time stamp



< Existing method of selecting negative pairs >

Introduction

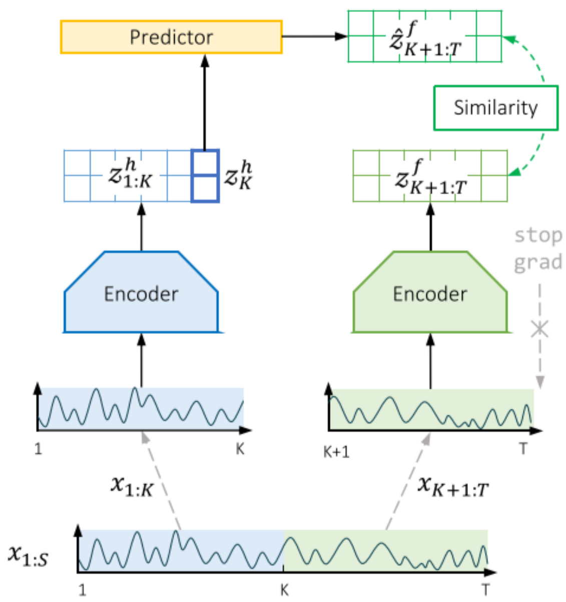
- Key questions
  - What is important for time series forecasting with contrastive learning?
  - How can we adapt contrastive ideas more effectively to time series forecasting tasks?
  - Are the existing assumptions and techniques for constructing positive and negative pairs reasonable?

Proposed Method

- Motivation
  - What is important for time series forecasting with contrastive learning?
    - Existing methods
      - They ignore the possibility that a repeating patterns exist within a time series
      - They disregard the possibility that distinct time series contain similar patterns
  - In forecasting task, a good representation should effectively capture the temporal dependencies between past segments and future predictions.
  - We emphasize that the temporal dependency has greater significance than the similarity between positive and negative pairs.
  - Negative pairs inducing the issue of false repulsion
    - Patterns that repeat across different samples
  - → We train an encoder to learn time series representations by predicting its future from historical segments in the latent space.

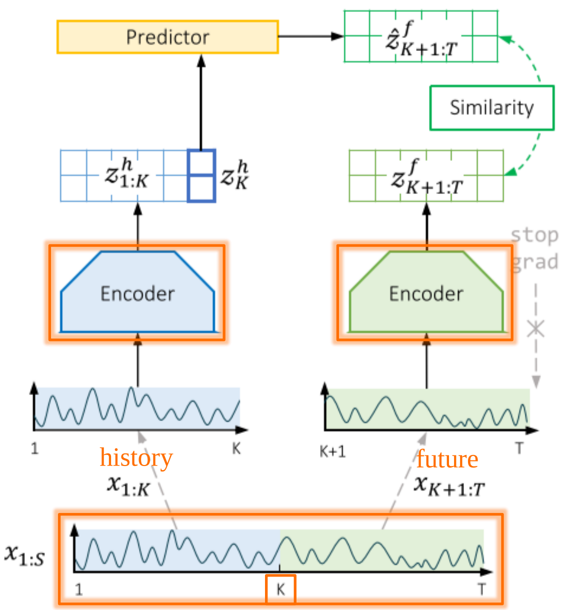
Proposed Method

- SimTS: Simple Representation Learning for Time Series Forecasting
  - Siamese neural network architecture
    - Two identical networks sharing parameters
    - *History* encoding path, *future* encoding path
  - Multi-scale encoder network F_θ
    - Projecting raw features into a high dimensional space
    - Multiple CNN blocks with different kernel sizes
  - Prediction network $G_\emptyset$
    - Input: last column of encoded history view
    - Predicting future in latent space
  - Cosine similarity loss
    - Considering only positive samples



Proposed Method

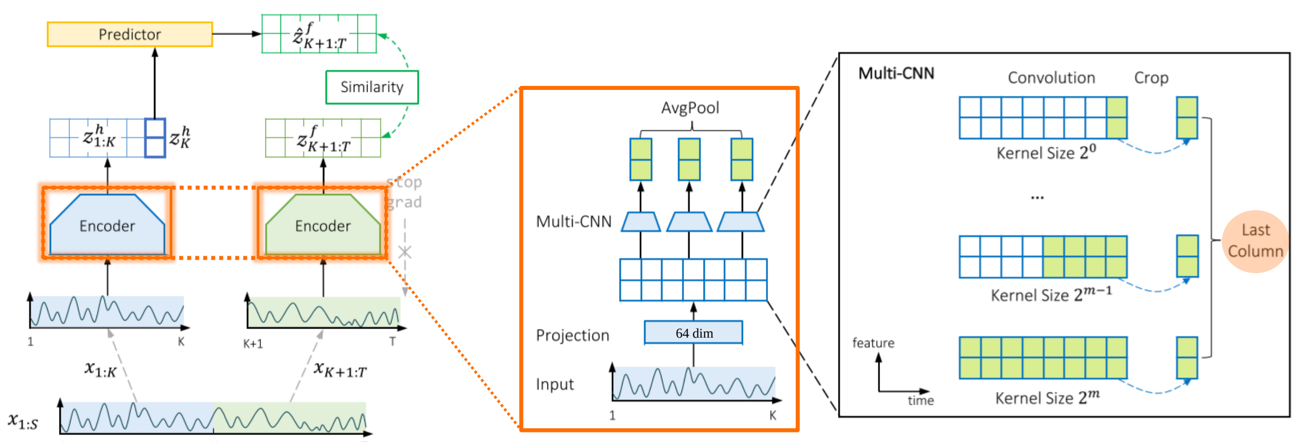
- SimTS
  - Objective
    - Learning latent representation of *history* segment
      $X^h = [x_1, x_2, \dots, x_K], \qquad 0 < K(201) < T(402)$
    - Predicting *future* segment $X^f = [x_{K+1}, x_{K+2}, \dots, x_T]$
  - Multi-scale Encoder network F_θ
    - Inputs: *history* segment X^h, *future* segment X^f
      - Learning to map them to their latent representations Z^h, Z^f
    - *History* encoding path
      - $Z^h = F_\theta(X^h) = [z_1^h, z_2^h, \dots, z_K^h] \in R^{C' \times K}, \ (C':320)$
    - *Future* encoding path
      - $Z^f = F_\theta(X^f) = [z_{K+1}^f, z_{K+2}^f, \dots, z_T^f] \in R^{C' \times (T-K)}$



Proposed Method

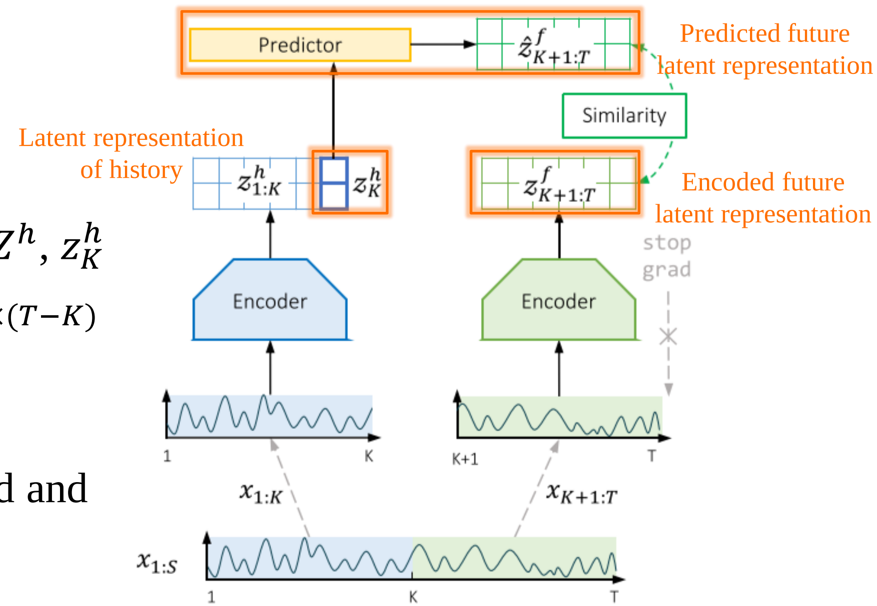
- SimTS
  - Multi-scale Encoder network F_θ
    - Convolutional network with multiple filters that have various kernel sizes
    - Extracting both local/global patterns
    - For a time series X with length K, $\underline{m = [log_2 K] + 1}$ <u>parallel convolution layers,</u> kernel size 2^{i-1} of ith convolution

      ∴ $K = 8, 16, 32, 64, \ldots,$ → $m = 4, 5, 6, 7, \ldots,$
      → kernel size = (1, 2, 4, 8), (1, 2, 4, 8, 16), …,
    - Average pooling of the last column of each CNN layer



Proposed Method

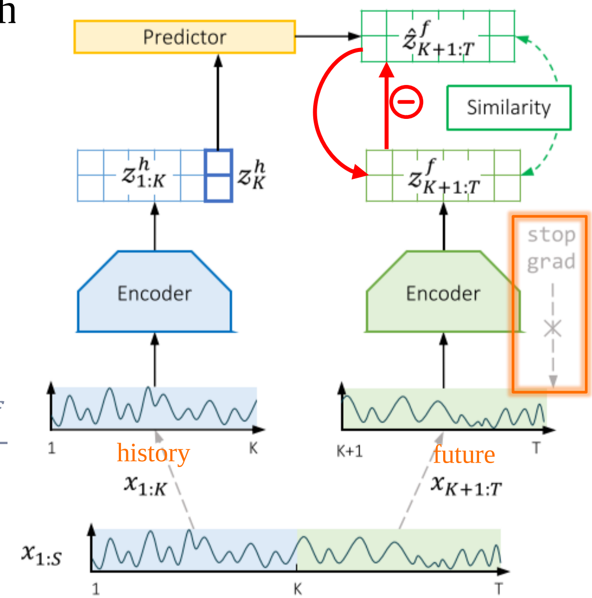
- SimTS
  - Prediction network $G_{\emptyset}$
    - Predicting *future* latent representations
    - Input: last column of encoded *history* view Z^h, z_K^h
    - $\hat{Z}^f = G_\phi(z_K^h) = [\hat{z}_{K+1}^f, \hat{z}_{K+2}^f, \ldots, \hat{z}_T^f] \in R^{C' \times (T-K)}$
  - Similarity
    - <u>Maximizing the similarity</u> between predicted and encoded latent features
      - Forcing them to get closer
      - Learning historical representation that is informative for the future
      - <u>Regarding the predicted $\hat{Z}^f$ and the encoded Z^f as the **positive pair**</u>
      - Cosine similarity



$$\checkmark Sim(\hat{Z}^f, Z^f) = -\frac{1}{T-K}\sum_{i=K+1}^{T} \frac{\hat{z}_i^f}{\|\hat{z}_i^f\|_2} \cdot \frac{z_i^f}{\|z_i^f\|_2}, \quad \|\cdot\|_2: l_2\text{-norm}$$

Proposed Method

- SimTS
  - Stop-gradient Operation
    - Using same encoder for both *history* and *future* encoding path
      - Problem of optimizing the encoder by pushing encoded *future* Z^f towards predicted *future* $\hat{Z}^f$
    - Applying it to *future* encoding path
      - $\hat{Z}^f$ can only move towards Z^f in the latent space
    - Encoder network
      - Unable to receive updates from future representations Z^f
      - Constrained to only optimize the history representation Z^h and its prediction $\hat{Z}^f$
    - Loss function
      - $L_{\theta,\phi}(X^h, X^f) = Sim\left(G_\theta\left(F_\theta(X^h)\right), F_{sg(\theta)}(X^f)\right) = Sim(\hat{Z}^f, sg(Z^f))$



Experimental Results

- Experimental setup
  - Input size
    - Length T = 402
    - First 201 corresponding to history view, subsequent 201 corresponding to future view
    - Projected to a 64-dim latent space in projection
    - Projected to a 320-dim latent space in multi-scale encoder
  - Datasets
    - Electricity Transformer Temperature
      - ETTh1, ETTh2, ETTm1, ETTm2
    - Exchange-Rate
    - Weather
  - Training, validation, test sets in the ratio of 6:2:2
  - 500 epoch, learning rate 0.001, batch size 8

Experimental Results

: using negative pairs

Random masking → TS2Vec

Scaling, shifting, jittering → CoST

Methods		Unsupervised Representation Learning								End-to-end Forecasting			
		Ours		TS2Vec		TNC		CoST		Informer		TCN	
	L	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	**0.377**	**0.422**	0.590	0.531	0.708	0.592	0.386	0.429	0.577	0.549	0.583	0.547
	48	**0.427**	**0.454**	0.624	0.555	0.749	0.619	0.437	0.464	0.685	0.625	0.670	0.606
	168	**0.638**	**0.577**	0.762	0.639	0.884	0.699	0.643	0.582	0.931	0.752	0.811	0.680
	336	0.815	**0.685**	0.931	0.728	1.020	0.768	**0.812**	0.679	1.128	0.873	1.132	0.815
	720	**0.956**	**0.771**	1.063	0.799	1.157	0.830	0.970	0.771	1.215	1.869	1.165	0.813
ETTh2	24	**0.336**	**0.434**	0.424	0.489	0.612	0.595	0.447	0.502	0.720	0.665	0.935	0.754
	48	**0.564**	**0.571**	0.619	0.605	0.840	0.716	0.699	0.637	1.457	1.001	1.300	0.911
	168	**1.407**	**0.926**	1.845	1.074	2.359	1.213	1.549	0.982	3.489	1.515	4.017	1.579
	336	**1.640**	**0.996**	2.194	1.197	2.782	1.349	1.749	1.042	2.723	1.340	3.460	1.456
	720	**1.878**	**1.065**	2.636	1.370	2.753	1.394	1.971	1.092	3.467	1.473	3.106	1.381
ETTm1	24	**0.232**	**0.314**	0.453	0.444	0.522	0.472	0.246	0.329	0.323	0.369	0.522	0.472
	48	**0.311**	**0.368**	0.592	0.521	0.695	0.567	0.381	0.386	0.494	0.503	0.542	0.508
	96	**0.360**	**0.402**	0.635	0.554	0.731	0.595	0.378	0.419	0.678	0.614	0.666	0.578
	288	**0.450**	**0.467**	0.693	0.597	0.818	0.649	0.472	0.486	1.056	0.786	0.991	0.735
	672	**0.612**	**0.563**	0.782	0.653	0.932	0.712	0.620	0.574	1.192	0.926	1.032	0.756
ETTm2	24	**0.108**	**0.223**	0.180	0.293	0.185	0.297	0.122	0.244	0.173	0.301	0.180	0.324
	48	**0.164**	**0.285**	0.244	0.350	0.264	0.360	0.183	0.305	0.303	0.409	0.204	0.327
	96	**0.271**	**0.376**	0.360	0.427	0.389	0.458	0.294	0.394	0.365	0.453	3.041	1.330
	288	**0.716**	**0.646**	0.723	0.639	0.920	0.788	0.723	0.652	1.047	0.804	3.162	1.337
	672	**1.600**	**0.979**	1.753	1.007	2.164	1.135	1.899	1.073	3.126	1.302	3.624	1.484
Exchange	24	**0.059**	**0.172**	0.108	0.252	0.105	0.236	0.136	0.291	0.611	0.626	2.483	1.327
	48	**0.135**	**0.265**	0.200	0.341	0.162	0.270	0.250	0.387	0.680	0.644	2.328	1.256
	168	0.713	0.635	0.412	0.492	**0.397**	**0.480**	0.924	0.762	1.097	0.825	2.372	1.279
	336	1.409	0.938	1.339	0.901	**1.008**	**0.866**	1.774	1.063	1.672	1.036	3.113	1.459
	720	**1.628**	**1.056**	2.114	1.125	1.989	1.063	2.160	1.209	2.478	1.310	3.150	1.458
Weather	24	**0.298**	**0.359**	0.308	0.364	0.320	0.373	**0.298**	0.360	0.335	0.381	0.321	0.367
	48	**0.359**	**0.410**	0.375	0.417	0.380	0.421	**0.359**	0.411	0.395	0.459	0.386	0.423
	168	**0.426**	**0.461**	0.496	0.506	0.479	0.495	0.464	0.491	0.608	0.567	0.491	0.501
	336	0.504	0.520	0.532	0.533	0.505	0.514	**0.497**	**0.517**	0.702	0.620	0.502	0.507
	720	0.535	0.542	0.567	0.558	0.543	0.547	0.533	0.542	0.831	0.731	**0.498**	**0.508**
Avg.		**0.664**	**0.562**	0.818	0.632	0.909	0.669	0.746	0.603	1.151	0.812	1.560	0.884

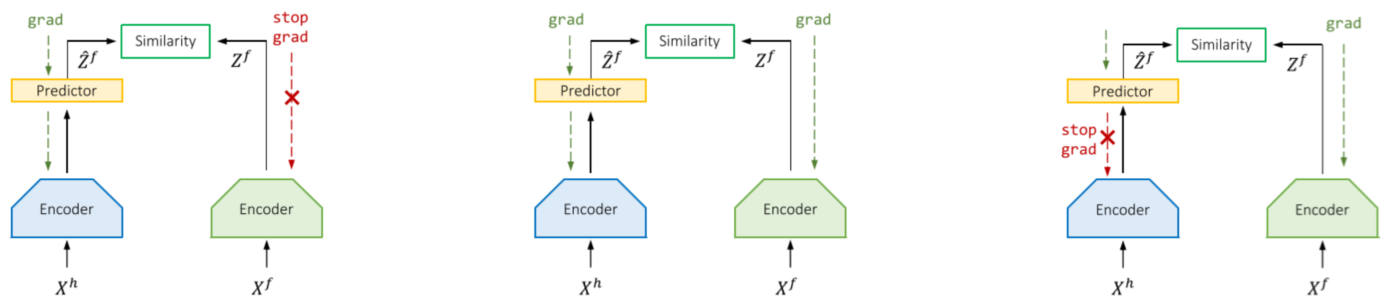
Experimental Results

- Negative samples
  - Required careful construction
  - TS2Vec and CoST using negative pairs for contrastive learning with poor performance
    - Selection of negative pairs may be inaccurate

Datasets	ETTh1		ETTh2		ETTm1		ETTm2		Exchange		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SimTS	0.642	0.582	1.165	0.798	0.393	0.423	0.572	0.502	0.789	0.613	0.424	0.458
SimTS w/ neg‡	0.685	0.632	1.544	0.938	0.392	0.441	0.747	0.572	1.405	0.769	0.434	0.468

Experimental Results

- Stop-Gradient Operation
  - SimTS: applying to *future* encoding path
  - SimTS w/o SG: not applying to any path
  - RevSimTS: applying to *history* encoding path



(a) SimTS (b) SimTS w/o SG (c) RevSimTS

Model	SimTS		SimTS w/o SG†		RevSimTS	
	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.642**	**0.582**	0.783	0.663	0.762	0.634
ETTh2	**1.165**	**0.798**	2.940	1.490	3.128	1.449
ETTm1	**0.393**	**0.432**	0.681	0.609	0.551	0.525
ETTm2	**0.572**	**0.502**	1.315	0.863	1.186	0.796
Exchange	**0.789**	**0.613**	1.808	1.062	1.398	0.900
Weather	**0.424**	**0.458**	0.605	0.592	0.485	0.512

Conclusion

- SimTS for time series forecasting
  - Using a simple encoder to learn representations in latent space without negative pairs
- The effectiveness and generalizability of SimTS

- Our goal to challenge the assumptions and components that are widely used
- Current representation learning methods that cannot be universally applicable to different types of time series data

Thanks!